Scheduling Algorithms for Throughput Maximization in Data Networks

by

Andrew Brzezinski

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2007
(June 2007)

Author ....
Department of Electrical Engineering and Computer Science
May 14, 2007

Certified by ....
Eytan Modiano
Associate Professor
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Scheduling Algorithms for Throughput Maximization in Data Networks

by

Andrew Brzezinski

Submitted to the Department of Electrical Engineering and Computer Science
on May 14, 2007, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

This thesis considers the performance implications of throughput optimal scheduling in physically and computationally constrained data networks. We study optical networks, packet switches, and wireless networks, each of which has an assortment of features and constraints that challenge the design decisions of network architects. In this work, each of these network settings are subsumed under a canonical model and scheduling framework. Tools of queueing analysis are used to evaluate network throughput properties, and demonstrate throughput optimality of scheduling and routing algorithms under stochastic traffic. Techniques of graph theory are used to study network topologies having desirable throughput properties. Combinatorial algorithms are proposed for efficient resource allocation.

In the optical network setting, the key enabling technology is wavelength division multiplexing (WDM), which allows each optical fiber link to simultaneously carry a large number of independent data streams at high rate. To take advantage of this high data processing potential, engineers and physicists have developed numerous technologies, including wavelength converters, optical switches, and tunable transceivers. While the functionality provided by these devices is of great importance in capitalizing upon the WDM resources, a major challenge exists in determining how to configure these devices to operate efficiently under time-varying data traffic. In the WDM setting, we make two main contributions. First, we develop throughput optimal joint WDM reconfiguration and electronic-layer routing algorithms, based on *maxweight* scheduling. To mitigate the service disruption associated with WDM reconfiguration, our algorithms make decisions at frame intervals. Second, we develop analytic tools to quantify the maximum throughput achievable in general network settings. Our approach is to characterize several geometric features of the maximum region of arrival rates that can be supported in the network.

In the packet switch setting, we observe through numerical simulation the attractive throughput properties of a simple *maximal* weight scheduler. Subsequently, we consider small switches, and analytically demonstrate the attractive throughput properties achievable using maximal weight scheduling. We demonstrate that such throughput properties may not be sustained in larger switches.

In the wireless network setting, mesh networking is a promising technology for achieving connectivity in local and metropolitan area networks. Wireless access points and base stations adhering to the IEEE 802.11 wireless networking standard can be bought off the shelf at little cost, and can be configured to access the Internet in minutes. With ubiquitous low-cost Internet access perceived to be of tremendous societal value, such technology is naturally garnering strong interest. Enabling such wireless technology is thus of great importance. An important challenge in enabling mesh networks, and many other wireless network applications, results from the fact that wireless transmission is achieved by broadcasting signals through the air, which has the potential for interfering with other parts of the network. Furthermore, the scarcity of wireless transmission resources implies that link activation and packet routing should be effected using simple distributed algorithms. We make three main contributions in the wireless setting. First, we determine graph classes under which simple, distributed, *maximal* weight schedulers achieve throughput optimality. Second, we use this acquired knowledge of graph classes to develop combinatorial algorithms, based on matroids, for allocating channels to wireless links, such that each channel can achieve maximum throughput using simple distributed schedulers. Third, we determine new conditions under which distributed algorithms for joint link activation and routing achieve throughput optimality.

Thesis Supervisor: Eytan Modiano
Title: Associate Professor

Acknowledgments

My years in grad school have been intensely enjoyable and rewarding, partly as an intellectually stimulating experience, and largely because of the excellent people with whom I've had the pleasure to work, live, and play.

My deepest bow is to Eytan Modiano, my Ph.D. advisor, whose vision inspired this research, whose meticulous oversight kept it focused, whose enthusiasm has always been contagious, whose amiable nature carried the advisor-advisee relationship into friendship.

Thanks to my other Ph.D. committee members, Professors Hari Balakrishnan and Pablo Parrilo, for reading this thesis, providing constructive critiques, and suggesting improvements.

I am grateful for having the opportunity to interact with the students and professors at the Laboratory for Information and Decision Systems, and in the Communications and Networking Research Group. I especially must acknowledge the following people, each of whom has markedly shaped my experience at MIT: Emmanuel Abbe, Mukul Agarwal, Shashi Borade, Guner Celik, Li-Wei Chen, Lillian Dai, Anand Ganti, Shan-Yuan Ho, Doris Inslee, Krishna Jagannathan, Sid Jaggi, Amir Khandani, Michael Lewy, Baris Nakiboglu, Mike Neely, Marylin Pierce, Asu Ozdaglar, Rosangela dos Santos, Anand Srinivas, Jun Sun, Greg Tomezak, Moe Win, Murtaza Zafer, and Gil Zussman. I spent an excellent summer at the Bell Labs Math Sciences Research Center in 2004, and am grateful to have worked with the people there, especially Iraj Saniee, Indra Widjaja, and Carl Nuzman.

Thanks to my friends Michael Baer, Alton Lam, Krishna Pandit, and Alessandro Tarello.

I am grateful to my parents for setting me on the path that led me here, and to my brothers who I absolutely haven't seen enough of in recent years. Thanks to Roya, the kindest mother-in-law in the world, and to Vahid, who is the older brother I never had. *I love all of you guys.*

I dedicate this thesis to my wife Ashley. If we hadn't met, I would never have come to MIT. Seeing how great these years have been, and how promising the future appears, I'm glad to be on this journey with her. *I love you, Ashley!*

Support

This thesis was supported in part by DARPA under grant MDA972-02-1-0021, by NSF under grants ANI-0073730 and ANI-0335217, by ONR under grant N000140610064, by DARPA/AFOSR through the University of Illinois grant F49620-02-1-0325, and by NASA under grant NAG3-2835.

Preliminaries

Notation

We use $\mathbb{R}$ to denote the set of real numbers. We use $\mathbb{R}_+$ to denote $[0, \infty)$, the set of non-negative reals. We use $\mathbb{Z}$ to denote the set of integers, and $\mathbb{Z}_+$ to denote the non-negative integers. For a set $\mathcal{S}$, and an integer $n \geq 1$, the set $\mathcal{S}^n$ denotes the n-fold Cartesian product of $\mathcal{S}$. Thus, $\mathbb{R}^n$ is the n-dimensional real coordinate space, and $\mathbb{R}^n_+$ is the positive orthant in n dimensions. Scalar quantities are italicized, e.g., x. Bold symbols are associated with vectors and matrices, e.g., $\mathbf{x} = (x_1, \ldots, x_n)$. For a vector $\mathbf{x} = (x_1, \ldots, x_n)$ and an index set $I \subseteq \{1, \ldots, n\}$, we denote the subvector $\mathbf{x}_I = (x_i, i \in I)$. The cardinality operator is $|\cdot|$, with $|\mathcal{S}|$ representing the cardinality of the set $\mathcal{S}$. The convex hull operator is $\mathrm{conv}(\cdot)$, where for the set $\mathcal{A} \subseteq \mathbb{R}^n$,

$$\mathrm{conv}(\mathcal{A}) = \{\alpha \mathbf{a}_1 + \beta \mathbf{a}_2 : \mathbf{a}_1, \mathbf{a}_2 \in \mathcal{A}, \alpha \geq 0, \beta \geq 0, \alpha + \beta = 1\}.$$

An accumulation point of $\mathcal{R} \subseteq \mathbb{R}^n$ is such that there exist other points of $\mathcal{R}$ arbitrarily close by. The closure of $\mathcal{R}$ is then given by the union of $\mathcal{R}$ and all its accumulation points [93]. The closure operator is $\mathrm{cl}(\cdot)$. We use w.p.1 to represent the statement 'with probability 1'. The indicator function is represented by $\mathbb{1}_{\{\cdot\}}$, returning unity when its argument is true, and zero when its argument is false.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the continuing growth in demand for data traffic, the existing network infrastructure will be strained in terms of both transport and processing requirements. Advances in technology and hardware capability drive much of the progress in meeting increasing demands, often at the expense of additional costs. It is natural and economically sound that network designers and administrators will always seek cost-effective strategies for addressing the needs of their customers. Consequently, a great deal of interest inevitably arises when low cost, high-performance communication solutions are proposed, particularly when these solutions are compatible with or designed to be implemented using *existing technology*. Recent examples of this phenomenon include the drive towards low cost WIFI-based wireless mesh networks [1,7,8,77,122], the intense interest in software and cognitive radio [18,20,30,50,57,70,100,101,159], and the incorporation of electronic aggregation (grooming) and optical aggregation (wavebanding) techniques for efficient utilization of bandwidth in optical networks [19,33,102,103,116,135,143,168].

The essential driver of data networking has been the steadily decreasing cost and miniaturization of computing technology. As processing power improves, our communication solutions can admit increasingly intelligent and sophisticated network control algorithms. For example, small, inexpensive, mobile wireless devices can be made robust to channel variations, interference, and mobility, leading to effective solutions for cellular communication, wireless ad-hoc networks, and mesh networks. In the optical network setting, configurable components in combination with intelligent network control algorithms enable networks that are adaptable and responsive to traffic variations.

In this thesis, we develop dynamic algorithms for routing and scheduling traffic in data networks. In Chapter 2, we introduce the general network setting of interest. The remainder of the thesis is dedicated to studying the algorithmic and performance implications of networking in various communication settings. Our network model encompasses wireless and optical networks, as well as high-speed electronic packet switches. We study each of these networking scenarios, focusing on enabling efficient network control algorithms given their respective engineering constraints. For example, wireless networks are subject to co-channel interference and cannot effectively admit centralized control policies [90]. WDM-based optical networks have wavelength and port constraints, and incur non-negligible delays associated with propagation of light, component configuration, and link synchronization [121].

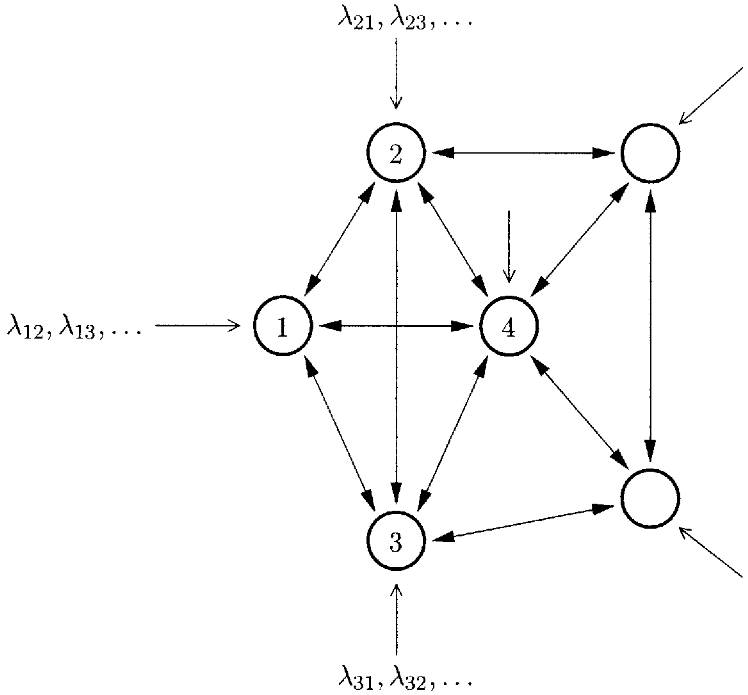

$\lambda_{21}, \lambda_{23}, \ldots$
2
$\lambda_{12}, \lambda_{13}, \ldots$
1
4
3
$\lambda_{31}, \lambda_{32}, \ldots$

Figure 1-1: Generic representation of the networking scenario. Arrivals at each node can be destined to any other node in the network. The corresponding arrival rate for source destination pair (i, j) is indicated with λ_{ij}. Note that available communication links are depicted as bidirectional, though this need not be the case in general.

We address WDM-based optical networks in Chapters 3–5, where we introduce throughput-optimal routing and scheduling algorithms and quantify achievable performance. We study efficient switch scheduling in Chapter 6. Finally, we consider wireless networks in Chapters 7–9, and determine network topologies for which simple distributed algorithms maximize throughput.

1.1 Research overview

Our study of network layer throughput properties in wireless and optical networks has a common underlying network queueing model. The details of this model are presented in Chapter 2. In this section, we introduce the key elements of this model.

Figure 1-1 depicts a generic representation of the networking scenario of interest. The network consists of a set of nodes, representing users in the wireless setting, input and output ports of a crossbar switch, or switching/router equipment in the optical setting. Network links provide a means for establishing communication between nodes. In wireless networks, a link exists between two users when their wireless communication channel is sufficiently strong for reception. In crossbar switches, a link exists between every input/output port pair. In optical networks, a link exists between any two nodes that can communicate all-optically (without intermediate electronic processing) through the network.

Each network node is subjected to random exogenous arrivals of packets. These packets can be destined to any other network node, and often require routing between multiple nodes of the network to reach their destinations.

Service of packets through the network is effected through activation of communication links over the network. In the wireless setting, a communication link is always established between nodes in direct communication with one another. This is not the case in optical networks however, where switching at the optical layer allows communication links to *optically bypass* intermediate nodes. Multiple communication links can be simultaneously active, subject to the physical communication constraints of the network. For example, wireless communication is constrained by co-channel interference, while optical networks are constrained by bandwidth (wavelength) and processing (transceiver and wavelength conversion) capability.

Naturally, each network setting that we consider will have an order of precedence among the various engineering considerations required to enable communication. For example, distributed network control is an essential feature in many wireless networking applications, while being unimportant in electronic switches, where the primary concern is to enable efficient resource utilization using simple schedulers. In this thesis, we focus our attention on the following general questions.

> ***What network control algorithms maximize achievable performance?***
>
> ***Can we determine precise measures of performance?***
>
> ***How do network topology and architectural features affect performance?***
>
> ***How do we effectively utilize resources in a distributed fashion?***
>
> ***Does distributed network control lead to a performance penalty?***

Since we consider optical, wireless, and electronic switching applications, our emphasis on these questions varies throughout the thesis. We determine algorithms for achieving the maximum throughput in optical, wireless, and switch settings. We exactly quantify throughput properties achievable in optical networks. We explore the stability properties of low-complexity switch scheduling algorithms. We develop channel allocation algorithms to enable efficient decentralized network control in wireless networks. We study network topologies that are amenàble to achieving the maximum throughput using simple distributed schedulers.

1.2 Related work

The seminal work of Tassiulas and Ephremides underlies much of the existing literature in the area of stability of data networks [150]. In that work, a backpressure-based algorithm for scheduling multi-commodity packets in a general network setting was proposed and proved to be throughput optimal. The general network setting that we consider in this thesis fits into the framework of Tassiulas and Ephremides. In Chapter 2, we describe our general network model as well as the algorithm of [150].

In the design of traffic-adaptive networking algorithms, the maximum weight scheduling discipline has received considerable attention [3–6,9,10,12,49,54–56,62–65,76,86,88,95–97, 111–115,125,126,138–141,146,148–151,163], owing largely to its application in algorithm design and analysis in crossbar switch scheduling [3,4,6,49,63–65,73,76,88,95–97,138–140]. The first appearance of maximum weight scheduling for general networks was in [150]. In the context of switch scheduling, the maximum weight matching algorithm was first enlisted and shown to be throughput optimal in [95–97]. Subsequent works on switch scheduling focused on developing implementable algorithms by considering speedup [35,44, 49,80,88,118], randomization [64,76,138], approximate maximum weight scheduling [73,139], variable-length packets [4], parallel techniques [65], multicast [3,78], multiclass traffic [6], and modified crossbar architecture [63,167]. Additionally, delay performance of switch scheduling algorithms was considered in [76,78,79,87,140]. Outside of switch scheduling, maximum weight scheduling has been applied in wireless [55,56,111,113,114,148–150], satellite [115], and optical [26,27,74,154,155] network settings.

Fluid models are a standard tool for studying the stability of queueing networks [24], where stability properties are investigated by studying the corresponding *fluid limits* [23, 24,37,38,46–49,51,52,98,107,130,140,141,145,146]. The papers [37,46,48,130,145] are largely responsible for introducing and establishing fluid limits as effective means of studying stability. Rybko and Stolyar [130] studied a simple network, which was generalized by Dai [46] (this is regarded as the main reference on fluid limits in the literature) and by Stolyar [145]. The papers [5,24,47,98] considered the implications of instability in fluid limit models upon corresponding queueing network stability.

Dai and Prabhakar [49] are responsible for the first treatment of fluid models and fluid limits of input-queued switches, where they demonstrated the rate stability of the maximum weight matching service discipline, as well as of maximal size matching under a speedup of two. This analysis was extended to general switched networks in [9,146]. In [146], the maximum weight scheduling discipline was proved to be throughput optimal, and to achieve optimal delay performance when exactly one port is saturated. For the $N \times N$ input-queued switch, the delay optimality of maximum weight scheduling was studied in [140] under general saturated port loadings. The works of [12,140,146] restrict to non-negative service at each queue, which precludes routing in the respective networks considered.

An alternative tool for studying the stability of queueing networks is the Lyapunov drift technique [13], applied directly upon the discrete-time queueing model. This technique was employed in the stability analysis of Tassiulas and Ephremides [150], and has been a standard approach in subsequent works in the scheduling literature [3,4,54–56,63,64,76, 81,86–88,95–97,111–115,148,149,151].

The algorithm of Tassiulas and Ephremides [150] employs centralized maximum weight scheduling, which is often considered too complex to implement on a slot-by-slot basis in high-speed data networks [29,35,36,39,44,49,80,88–90,104,118,136,160,161]. In the wireless setting, the design of distributed scheduling algorithms has attracted a great deal of attention. Lin and Shroff [90] studied the impact of imperfect scheduling on cross-layer rate control. Under primary interference constraints[1], they showed that using a distributed

[1] Primary interference constraints imply that each pair of simultaneously active links must be separated

maximal matching algorithm along with a rate control algorithm is only guaranteed to achieve 50% throughput. Similar results for different settings were also obtained in [36,39, 89,136,160,161]. Chaporkar et al. [36,136] characterize the stability region of a maximal scheduling algorithm under arbitrary topologies and interference models. They show that under secondary interference constraints, the stability region may be reduced to $\Lambda^*/8$, where Λ^* is the stability region under a perfect (centralized) scheduler. A novel distributed *randomized* approach that can achieve 100% throughput has been presented in [104].

An important work that we study at length in this thesis is Dimakis and Walrand [51]. They consider the performance of the Longest Queue First (LQF) scheduling algorithm (a greedy maximal weight scheduling algorithm) in a graph of interfering queues, and present sufficient conditions (called *Local Pooling*) for a maximal weight algorithm to provide 100% throughput.

1.3 Contributions

Chapter 2 of this thesis introduces the queueing model that encompasses the network settings considered in this thesis. Many of the important notations and queueing variables are presented, as well as the formal definition of stability. The algorithm of Tassiulas and Ephremides [150] is presented for the queueing model, and we provide a proof of its stability based on the *fluid limits* technique.

Chapter 3 primarily introduces the optical networking framework, and two throughput-optimal scheduling algorithms. These algorithms make *maximum weight (maxweight)* scheduling and routing decisions. Important engineering aspects of the networking problem are addressed, including link propagation delay, transceiver tuning latency, and link synchronization delay. The delay associated with transmission of packets through the network is studied for several algorithms and networking scenarios.

Chapter 4 focuses on quantifying the maximum throughput properties of WDM-based optical networks, under a single-wavelength constraint. The chapter culminates in several theoretical results that exactly quantify two geometric properties of the network stability region in terms of the Routing and Wavelength Assignment problem. This enables us to determine closed-form expressions for the network performance under many common network topologies of interest.

Chapter 5 seeks to answer questions that arise naturally following the single-wavelength analysis of Chapter 4. We study the computational complexity associated with determining the geometric properties studied in Chapter 4. Additionally, we extend the results of Chapter 4 to the multi-wavelength setting.

Chapter 6 is motivated by the fact that implementing *maximum weight* scheduling may be computationally cumbersome. This chapter looks at a lower complexity scheduling algorithm, which makes link activation decisions based on *maximal weight* scheduling. We consider the simple case of bipartite network graphs, which are commonly studied in the context of input-queued switches. We present numerical results attesting to the attractive throughput properties of maximal scheduling, and proceed to develop a theoretical result

by at least one hop (i.e. the set of active links at any point of time constitutes a matching) [36,68,90,104,164].

for the case of a 2×2 input-queued switch. For the 3×3 switch, although *maximal weight* scheduling can have suboptimal throughput performance, we demonstrate that the network only loses throughput on a set of arrival rates having measure zero. Finally, we study larger switches, and find that maximal weight scheduling may result in throughput loss over a non-negligible portion of the switch capacity region.

Decentralized scheduling in wireless networks is the focus of Chapters 7 through 9. Here again, *maximal weight* scheduling plays an important role. In Chapter 7, we study certain conditions known as Local Pooling conditions, under which maximal weight scheduling can be shown to achieve maximum throughput. Under limited routing and interference models, we determine network topologies for which decentralized scheduling achieves maximum throughput. We also develop network partitioning algorithms, based on *matroids*, to separate the network links into channels, each of which achieves maximum throughput. In Chapter 8, we seek to loosen the restrictions imposed on the network in Chapter 7. In particular we greatly expand our knowledge of graphs that satisfy Local Pooling, and we study the implications of multi-hop interference constraints upon network stability. In Chapter 9, we extend the Local Pooling conditions to networks employing electronic routing.

Chapter 2

Network model, stability, and throughput maximization

In this chapter, we present the fundamental queueing network model used throughout this thesis. We define the capacity region of the network and formalize the concept of a throughput maximizing algorithm. We introduce an algorithm that maximizes throughput in the network.

2.1 Network queueing model for scheduling and routing

We consider a general network structure, which we denote by N. N represents all physical aspects of the network, including network topology, node architecture, and all communication mechanisms. The network consists of n nodes, physically interconnected in a graph structure $G_N = (V, E_N)$, where the vertex set V corresponds to the set of network nodes, and the directed edge set E_N corresponds to the communication links available in the network. Clearly $|V| = n$, and we denote $|E_N| = m$. For a directed edge e, let $\sigma(e)$ denote the source (initial) vertex, and $\tau(e)$ denote the terminal (destination) vertex.

The network consists in general of multiple sources and sinks of data. Hence, it is well described as a *multicommodity* data network. Throughout the work, we will treat data destined for a particular terminal node $v \in V$ as *commodity* v *data.*

For simplicity, we assume that time is slotted and that packets are of equal size, each packet requiring one time slot of service across any network link. Each node i is equipped with n queues, one for each possible destination of data traffic originating or passing through node i. The queue corresponding to packets at node i destined to node j is denoted by Q_{ij}, with $Q_{ij}(t)$ equal to the number of enqueued packets at the beginning of time slot t. The *differential backlog (backpressure)* of commodity j packets across edge $e \in E_N$ at time t is $Z_{ej}(t) = Q_{\sigma(e)j}(t) - Q_{\tau(e)j}(t)$. For link $e \in E_N$, the maximum backpressure at time $t \geq 0$ is given by $Z_e^*(t) = \max_{j \in V} Z_{ej}(t)$. For $t \geq 0$, denote

$$\mathbf{Q}(t) = (Q_{ij}(t), i, j \in V), \quad \mathbf{Z}(t) = (Z_{ej}(t), e \in E_N, j \in V), \quad \mathbf{Z}^*(t) = (Z_e^*(t), e \in E_N)$$

as the matrices of queue backlogs, link backpressures, and maximum backpressures, respectively. In some instances, we will refer to $\mathbf{Q}(t)$ as a vector instead of a matrix, where we adopt the convention $\mathbf{Q}(t) = (Q_{\sigma(e)\tau(e)}(t), e \in E_N)$.

Data traffic arrives for service through the network according to a stochastic process, $(A_{ij}(t), t \geq 0)$, where $A_{ij}(t)$ represents the cumulative number of exogenous arrivals up to the end of time slot t of packets to node i that are destined to node j. The arrival processes are assumed to be general, in the sense that they can be temporally and mutually correlated, with λ_{ij} equal to the long term rate of arrivals for each source/destination pair i, j, where

$$\lambda_{ij} = \lim_{t \to \infty} \frac{A_{ij}(t)}{t} \quad \text{w.p.1}$$

We make the assumption that there is no self-traffic in the network, which is represented symbolically with $A_{ii}(t) = 0$ for all times $t \geq 0$ and all nodes $i \in V$. Denote the *arrival rate matrix* $\boldsymbol{\lambda} = (\lambda_{ij}, i, j \in V)$. Service of packets is effected through network link activation and routing decisions for active links. Let Π_N denote the set of available link activations in the network graph G_N: the vector $\boldsymbol{\pi} = (\pi_e, e \in E_N) \in \Pi_N$ is a nonnegative integer vector, where π_e equals the total number of active communication links from node $\sigma(e)$ to node $\tau(e)$. Each allowable link activation is subject to the physical communication constraints of the network: a set of network links can be simultaneously activated depending on both network topology and network node functionality.

Service is applied to the system at each time slot by activating a set of edges, and routing a packet of a single commodity across each active link. We denote the corresponding *service activation matrix* by $\mathbf{S} = (S_{ej}, e \in E_N, j \in V)$. Here, S_{ej} equals the number of communication links from node $\sigma(e)$ to node $\tau(e)$ used to service commodity j packets under the activation $\mathbf{S}$. Every feasible matrix $\mathbf{S}$ is an integer matrix. Note that an admissible service activation matrix must have a valid underlying link activation belonging to Π_N. This property characterizes the set of admissible service activation matrices, $\mathcal{S}$:

$$\mathcal{S} = \left\{ \mathbf{S} \in \mathbb{Z}_+^{m \times n} : \sum_{j \in V} \mathbf{S}_{\cdot j} \in \Pi_N \right\}. \tag{2.1}$$

The set $\mathcal{S}$ places no restriction on which commodity is allowed to cross an active link. This means that service activations belonging to $\mathcal{S}$ can correspond to *multi-hop routing*, where packets are re-enqueued after transmission across a link. Thus, we will occasionally refer to $\mathcal{S}$ as $\mathcal{S}^{\text{mh}}$, to emphasize this multi-hop capability. In this thesis, we will also deal with networks in which single-hop routing is exclusively employed. In such a situation, the set of admissible service activations is denoted $\mathcal{S}^{\text{sh}}$, where $\mathbf{S} \in \mathcal{S}^{\text{sh}}$ must satisfy

$$S_{ej} > 0 \text{ implies } j = \tau(e). \tag{2.2}$$

In words, the above statement means that a link can only be activated to service traffic directly to its destination node.

The matrix $\mathbf{S} \in \mathcal{S}$ leads to packet transitions through the network. To model the queue evolution implied by invoking $\mathbf{S}$, we introduce for each commodity $j \in V$ the $n \times m$ *routing* *matrix* $\mathbf{R}^j = (R^j_{ie},\, i \in V,\, e \in E_N)$, where:

$$R^j_{ie} = \begin{cases} 1, & \text{if } \sigma(e) = i \\ -1, & \text{if } \tau(e) = i \text{ and } i \neq j \\ 0, & \text{else} \end{cases}$$

Denote by $d_{ij}(\mathbf{S})$ the service to queue Q_{ij} under activation matrix $\mathbf{S}$. Using the above routing matrix we can express $d_{ij}(\mathbf{S}) = \sum_k \mathbf{R}^j_{ik}\mathbf{S}_{kj}$. Denote matrix $\mathbf{d}(\mathbf{S}) = (d_{ij}(\mathbf{S}), i, j \in V)$.

Denote by $D_{ij}(t)$ the total service applied to commodity j packets at node i up to the end of time slot t. Finally for each $\mathbf{S} \in \mathcal{S}$, denote by $F_{\mathbf{S}}(t)$ the number of time slots up to the end of time slot t in which service activation matrix $\mathbf{S} \in \mathcal{S}$ was active. The following are the dynamics of the queueing system for $t \geq 0$:

$$Q_{ij}(t) = Q_{ij}(0) + A_{ij}(t) - D_{ij}(t), \quad \forall (i,j) \tag{2.3}$$

$$D_{ij}(t) = \sum_{\mathbf{S} \in \mathcal{S}} d_{ij}(\mathbf{S}) F_{\mathbf{S}}(t), \quad \forall (i,j) \tag{2.4}$$

$$\sum_{\mathbf{S} \in \mathcal{S}} F_{\mathbf{S}}(t) = t, \quad \forall t \tag{2.5}$$

$$F_{\mathbf{S}} \text{ is non-decreasing}, \quad \forall \mathbf{S} \in \mathcal{S} \tag{2.6}$$

In this thesis, we will only consider the case $Q_{ij}(0) = 0$ for all i, j.

2.2 Throughput optimality

We are now prepared to define the stability region of the network.

Definition 2.2.1 (Admissible Rate Vector) *An arrival rate matrix* $\boldsymbol{\lambda} = (\lambda_{ij},\, i, j \in V)$ *is admissible if it is non-negative and there exists a collection of service activation matrices* $\mathbf{S}^l \in \mathcal{S}$, $1 \leq l \leq L$ *such that for all* $i, j \in V$,

$$\lambda_{ij} \leq \sum_{l=1}^{L} \alpha_l d_{\sigma(e)\tau(e)}(\mathbf{S}^l), \quad \textit{where } \alpha_l \geq 0\, \forall l \textit{ and } \sum_{l=1}^{L} \alpha_l \leq 1. \tag{2.7}$$

The set of all admissible rate matrices is called the network capacity region and is denoted by $\boldsymbol{\Lambda}^*$.

A scheduling algorithm at each time slot makes a link activation and routing decision that is constrained to the set of available service activations $\mathcal{S}$. Under an algorithm for link activation and routing decisions, the queue backlogs evolve according to the process $(\mathbf{Q}(t), t \geq 0)$. We next define the network capacity region, based on the notion of stability usually referred to as *rate stability* [6, 36, 49].

Definition 2.2.2 (Stability Region) *The network stability region under algorithm* A, $\boldsymbol{\Lambda}_{\mathrm{A}}$, *consists of the set of rate vectors* $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*$ *such that any arrival process having rate* *matrix* $\boldsymbol{\lambda}$ *induces a process* $(\mathbf{Q}(t), t \geq 0)$ *satisfying*

$$\lim_{t\to\infty} \frac{Q_{ij}(t)}{t} = 0 \quad w.p.1 \quad \forall i,j \in V. \tag{2.8}$$

When the queue backlog process satisfies (2.8), we say that the algorithm is *stable* for arrival rates $\boldsymbol{\lambda}$. A *throughput optimal algorithm* or an algorithm that *achieves 100% throughput*, is defined as follows.

Definition 2.2.3 (Throughput Optimal Algorithm) *A scheduling algorithm* A *is throughput optimal if* $\boldsymbol{\Lambda}_{\mathrm{A}} = \boldsymbol{\Lambda}^*$.

Note that the above definitions can all be understood for the case of single-hop routing as well. Thus, given the set of single-hop service activations, $\mathcal{S}^{\mathrm{sh}}$, a single-hop admissible rate region, $\boldsymbol{\Lambda}^*_{\mathrm{sh}}$, can be defined:

Definition 2.2.4 *An arrival rate matrix* $\boldsymbol{\lambda} = (\lambda_{ij},\ i,j \in V)$ *is single-hop admissible if it is non-negative and there exists a collection of service activation matrices* $\mathbf{S}^l \in \mathcal{S}^{\mathrm{sh}}$, $1 \leq l \leq L$ *such that for all* $i,j \in V$,

$$\lambda_{ij} \leq \sum_{l=1}^{L} \alpha_l d_{\sigma(e)\tau(e)}(\mathbf{S}^l), \quad \textit{where } \alpha_l \geq 0 \, \forall l \textit{ and } \sum_{l=1}^{L} \alpha_l \leq 1.$$

is satisfied. The set of all single-hop admissible rate matrices is called the single-hop capacity region and is denoted by $\boldsymbol{\Lambda}^*_{\mathrm{sh}}$.

It is simple to demonstrate (see Appendix 2.A) that

$$\boldsymbol{\Lambda}^*_{\mathrm{sh}} = \mathrm{conv}(\Pi_N). \tag{2.9}$$

To clearly distinguish the single-hop and general multi-hop capacity regions, we will refer occasionally to the multi-hop capacity region $\boldsymbol{\Lambda}^*$ as $\boldsymbol{\Lambda}^*_{\mathrm{mh}}$.

2.3 Centralized throughput optimal scheduling and routing

Tassiulas and Ephremides developed a stable scheduling and routing algorithm that applies in this setting [150]. The algorithm is presented as Algorithm 1 below.

In step 3, Algorithm 1 assigns a weight to each edge $e \in E_N$, equal to the maximum backpressure across that edge. In step 4, the algorithm obtains a maximum weight link activation based on the backpressure link weights. In step 5, the algorithm makes routing decisions to service commodities achieving maximum backpressure. Note that the combination of steps 4 and 5, where link activation and routing decisions are respectively made, implies the selection of a service activation matrix $\mathbf{S} \in \mathcal{S}$.

Algorithm 1 is often referred to as a *maximum weight (maxweight) scheduling algorithm.* In [150], it was proved that Algorithm 1 is stable over the network capacity region, up to a set of measure zero. The result requires the following restrictions on the arrival processes:

Algorithm 1 Backpressure-based maximum weight scheduling algorithm

```
for time $t \geq 0$ do
    For each directed edge $e \in E_N$ assign
        $Z_{ej}(t) \leftarrow (Q_{\sigma(e)j}(t) - Q_{\tau(e)j}(t))$
    Assign $Z_e^*(t) \leftarrow \max_j Z_{ej}(t)$
    Obtain a maximum weight link activation $\boldsymbol{\pi}^* = (\pi_e^*,\ e \in E_N)$, where
        $\boldsymbol{\pi}^* \in \arg\max_{\boldsymbol{\pi} \in \Pi_N} \boldsymbol{\pi}^T \mathbf{Z}^*(t)$        (2.10)
    For each $e \in E_N$ such that $\pi_e^* \geq 1$, choose a commodity $j^* \in \arg\max_j Z_{ej}(t)$. Route
    $\min\{\pi_e^*, Q_{\sigma(e)j^*}(t)\}$ packets of commodity $j^*$ across $e$
end for
```

each traffic stream is i.i.d. with finite second moments, and the traffic streams are mutually independent. The proof uses a Lyapunov drift argument, which implies that the state space of queue backlogs can be partitioned into a set of states that are positive recurrent and a set of transient states that the system departs from in finite time with probability 1.

The following theorem demonstrates that Algorithm 1 achieves 100% throughput in the queueing network model of interest. The theorem is proved using the *fluid limits* technique, as opposed to the Lyapunov approach of [150]. This proof is a valuable exercise, as we will use the fluid limits approach to demonstrate stability of scheduling algorithms throughout this thesis. A secondary motivation for providing a proof of this theorem is that although it demonstrates a weaker notion of stability than that in [150], it covers a wider class of arrival processes: none of the above-mentioned restrictions need to be explicitly assumed at the outset. Finally, our weaker notion of stability allows us to conclude that the stability region under Algorithm 1 equals the *closed* network capacity region $\mathbf{\Lambda}^*$, instead of asserting equality *up to a set of measure zero.*

Theorem 2.3.1 *Algorithm 1 achieves 100% throughput.*

Proof: See Appendix 2.B. ■

In the following chapters, we will specialize our treatment to particular networking settings, which in each instance will require a characterization of the network G_N and the allowed service activation set $\mathcal{S}$. We will see that network topology and physical communication constraints play an important role in determining these quantities.

2.4 Model extensions

One might suggest the assumption of error-free transmission as a limitation of the networking model considered in this thesis. This model obviously abstracts away physical layer communication impairments that can lead to errors. However, augmenting the model with a finite probability of transmission failure is indeed trivial, and can be found in [150].

Appendix

2.A Proof that $\mathbf{\Lambda}^*_{\text{sh}} = \text{conv}(\Pi_N)$

We first show that $\text{conv}(\Pi_N) \subseteq \mathbf{\Lambda}^*_{\text{sh}}$. Consider $\boldsymbol{\lambda} \in \text{conv}(\Pi_N)$. Then there exist $\boldsymbol{\pi}_l$, $1 \leq l \leq L$ such that for all i, j,

$$\lambda_{ij} = \sum_{i=1}^{L} \alpha_l \pi^l_{ij}.$$

Above, $\alpha_l \geq 0$ for all l, and $\sum_l \alpha_l = 1$. We construct corresponding single-hop service matrices $\mathbf{S}^l$, $1 \leq l \leq L$ according to Algorithm 2, as follows. It is clear from the above algorithm that for all $e \in E_N$, $\sum_j S^l_{ej} = \pi^l_{\sigma(e)\tau(e)}$, and further that $S^l_{ej} > 0$ implies that $j = \tau(e)$. Thus, $\mathbf{S}^l \in \mathcal{S}^{\text{sh}}$ for $1 \leq l \leq L$. For each $e \in E_N$, this single-hop property allows us to express

Algorithm 2 Constructing a single-hop service matrix $\mathbf{S}^l = (S_{ej})$ from matrix $\boldsymbol{\pi}^l$

```
Assign $S^l_{ej} \leftarrow 0$ for all $e, j$
for all $e \in E_N$ do
    Assign $S^l_{e\tau(e)} \leftarrow \pi_{\sigma(e)\tau(e)}$
end for
```

$$\lambda_{\sigma(e)\tau(e)} = \sum_{l=1}^{L} \alpha_l S_{e\tau(e)} = \sum_{l=1}^{L} \alpha_l d_{\sigma(e)\tau(e)}(\mathbf{S}^l).$$

Thus, (2.7) is satisfied, which implies $\boldsymbol{\lambda} \in \mathbf{\Lambda}^*_{\text{sh}}$, as desired.

Next we show that $\mathbf{\Lambda}^*_{\text{sh}} \subseteq \text{conv}(\Pi_N)$. Consider $\boldsymbol{\lambda} \in \mathbf{\Lambda}^*_{\text{sh}}$, which implies that there exist $\mathbf{S}^l \in \mathcal{S}^{\text{sh}}$, $1 \leq l \leq L$ such that for all i, j (2.7) is satisfied. From the single-hop property of each matrix $\mathbf{S}^l$, we have for each $e \in E_N$

$$\lambda_{\sigma(e)\tau(e)} \leq \sum_{l=1}^{L} \alpha_l S^l_{e\tau(e)} = \sum_{l=1}^{L} \alpha_l \sum_j S^l_{ej} = \sum_{l=1}^{L} \alpha_l \pi^l_e,$$

where we assign $\pi^l_e \triangleq \sum_j S^l_{ej}$. Clearly $\boldsymbol{\pi}^l = (\pi_e)$ is by definition a valid link activation vector. If we assign $\tilde{\lambda}_{ij} = \lambda_{ij}$ for all i, j, then Algorithm 3 (below) obtains a decomposition of $\boldsymbol{\lambda}$ as a convex combination of link activation matrices.

At termination, we can express for all i, j $\lambda_{i,j} = \sum_{l=1}^{n^*} \alpha_l \pi^l_{ij}$, where $\alpha_l \geq 0$ for all l, and without loss of generality $\sum_l \alpha_l = 1$ (since we can always associate additional weight with the zero matrix, which is a valid logical configuration where no link is active). Thus, $\boldsymbol{\lambda} \in \text{conv}(\Pi_N)$, as desired.

We conclude that $\mathbf{\Lambda}^*_{\text{sh}} = \text{conv}(\Pi_N)$.

Algorithm 3 Translating the decomposition of $\tilde{\boldsymbol{\lambda}}$ to a decomposition of $\boldsymbol{\lambda}$

```
$n^* \leftarrow L$
for all $e \in E_N$ do
  if $\tilde{\lambda}_{\sigma(e)\tau(e)} = \lambda_{\sigma(e)\tau(e)}$ then
    continue
  end if
  $\omega \leftarrow \tilde{\lambda}_{\sigma(e)\tau(e)} - \lambda_{\sigma(e)\tau(e)}$
  $n^+ \leftarrow 1$
  for all $l \in \{1, \dots, n^*\}$ do
    if $0 < \alpha_l \pi^l_{\sigma(e)\tau(e)} \leq \omega$ then
      $\pi^l_{\sigma(e)\tau(e)} \leftarrow 0$
      $\omega \leftarrow \omega - \alpha_l \pi^l_{\sigma(e)\tau(e)}$
      if $\omega = 0$ then
        break
      end if
    else if $\alpha_l \pi^l_{\sigma(e)\tau(e)} > \omega$ then
      $\boldsymbol{\pi}^{n^*+n^+} \leftarrow \boldsymbol{\pi}^l$
      $\pi^{n^*+n^+}_{\sigma(e)\tau(e)} \leftarrow 0$
      $\alpha_{n^*+n^+} \leftarrow \alpha_l - \omega/\pi^l_{\sigma(e)\tau(e)}$
      $\alpha_l \leftarrow \omega/\pi^l_{\sigma(e)\tau(e)}$
      $n^+ \leftarrow n^+ + 1$
      break
    end if
  end for
  $n^* \leftarrow n^* + n^+ - 1$
end for
```

2.B Proof of Theorem 2.3.1

Recall from our definition that $Q_{ij}(t)$ is the number of commodity j packets enqueued at node i at the beginning of time slot $t \geq 0$. For the purposes of our analysis, we extend the queueing variables to the reals. For functions $A_{ij}, Q_{ij}, Z_{ej}, Z^*_e$, we use the floor function, where $A_{ij}(t)$ is to be interpreted as $A_{ij}(\lfloor t \rfloor)$, and similarly for the other functions. For functions D_{ij}, F_{S}, linear interpolation is employed, where $D_{ij}(t) = D_{ij}(\lfloor t \rfloor) + (t - \lfloor t \rfloor)(D_{ij}(\lceil t \rceil) - D_{ij}(\lfloor t \rfloor))$, and similarly for $F_{\mathrm{S}}(t)$. The linear interpolation is needed for its continuity properties.

For each of the above functions, we define for any $r > 0$ the scaled functions

$$\begin{aligned} A^r_{ij} &= \frac{A_{ij}(rt)}{r}, & D^r_{ij} &= \frac{D_{ij}(rt)}{r}, & Q^r_{ij} &= \frac{Q_{ij}(rt)}{r}, \\ Z^r_{ej} &= \frac{Z_{ej}(rt)}{r}, & Z^{*r}_e &= \frac{Z^*_e(rt)}{r}, & F^r_{\mathrm{S}} &= \frac{F_{\mathrm{S}}(rt)}{r}. \end{aligned}$$

The following lemma demonstrates convergence properties of sequences of the scaled functions, indexed by r. A sequence of functions $\{f^r\}$ where for each r, $f^r : \mathbb{R} \to \mathbb{R}$ converges uniformly on compact sets (u.o.c) if for each sequence $\{r_k\}$ there exists a subsequence $\{r_{k_l}\}$ and a function $\bar{f} : \mathbb{R} \to \mathbb{R}$, such that for $t \geq 0$,

$$\lim_{l \to \infty} \sup_{0 \leq t' \leq t} |f^{r_{k_l}}(t') - \bar{f}(t')| = 0.$$

In order to demonstrate uniform convergence on compact sets, we will enlist the Arzela-Ascoli theorem.

Theorem 2.B.1 (Arzela-Ascoli Theorem) *Consider a sequence of functions $\{f^{r_k}\}$ defined on closed interval $[t_1, t_2]$. If the sequence is uniformly bounded*[1] *and uniformly equicontinuous*[2]*, then there exists a subsequence $\{r_{k_l}\}$ that converges uniformly.*

We now proceed with a lemma concerning the convergence properties of the scaled functions.

Lemma 2.B.1 *The following statements hold with probability* 1*. For any sequence $\{r_k\}$, there exists a subsequence $\{r_{k_l}\}$ such that*

$$(A_{ij}^{r_{k_l}}(t), t \geq 0) \to (\bar{A}_{ij}(t), t \geq 0) \quad u.o.c., \quad \forall i,j \in V, \tag{2.11}$$

$$(D_{ij}^{r_{k_l}}(t), t \geq 0) \to (\bar{D}_{ij}(t), t \geq 0) \quad u.o.c., \quad \forall i,j \in V, \tag{2.12}$$

$$(Q_{ij}^{r_{k_l}}(t), t \geq 0) \to (\bar{Q}_{ij}(t), t \geq 0) \quad u.o.c., \quad \forall i,j \in V, \tag{2.13}$$

$$(F_{\mathbf{S}}^{r_{k_l}}(t), t \geq 0) \to (\bar{F}_{\mathbf{S}}(t), t \geq 0) \quad u.o.c., \quad \forall \mathbf{S} \in \mathcal{S}, \tag{2.14}$$

$$(Z_{ej}^{r_{k_l}}(t), t \geq 0) \to (\bar{Z}_{ej}(t), t \geq 0) \quad u.o.c., \quad \forall e \in E_N, j \in V, \tag{2.15}$$

$$(Z_e^{*r_{k_l}}(t), t \geq 0) \to (\bar{Z}_e^*(t), t \geq 0) \quad u.o.c., \quad \forall e \in E_N, \tag{2.16}$$

where the functions $\bar{A}_{ij}, \bar{D}_{ij}, \bar{F}_{\mathbf{S}}$ are Lipschitz-continuous[3] *in $[0, \infty)$, and functions $\bar{Q}_{ij}, \bar{Z}_{ej}, \bar{Z}_e^*$*

[1] A sequence of functions $\{f^{r_k}\}$, where $f^{r_k} : [t_1, t_2] \to \mathbb{R}$ is uniformly bounded if there exists $M \geq 0$ such that $|f^r(t)| \leq M$ $\forall t \in [t_1, t_2]$ and $\forall k$. [129]

[2] A sequence of functions $\{f^{r_k}\}$, where $f^{r_k} : [t_1, t_2] \to \mathbb{R}$ is uniformly equicontinuous if for every $\varepsilon > 0$ there exists $\delta > 0$ such that for all k and all $t_3, t_4 \in [t_1, t_2]$ with $|t_3 - t_4| < \delta$ we have $|f^{r_k}(t_3) - f^{r_k}(t_4)| < \varepsilon$. [129]

[3] A function $f : \mathbb{R} \to \mathbb{R}$ is Lipschitz-continuous if there exists $K \geq 0$ such that for all $t_1, t_2 \in \mathbb{R}$, $|f(t_1) - f(t_2)| \leq K|t_1 - t_2|$. [93]

are continuous in $[0, \infty)$*. Additionally, the following properties hold:*

$$\bar{A}_{ij}(t) = \lambda_{ij} t, \quad \forall i, j \in V, \, t \geq 0 \tag{2.17}$$

$$\bar{D}_{ij}(0) = 0, \quad \forall i, j \in V \tag{2.18}$$

$$\bar{F}_{\mathbf{S}}(0) = 0, \quad \forall \mathbf{S} \in \mathcal{S} \tag{2.19}$$

$$\bar{Q}_{ij}(t) = \bar{A}_{ij}(t) - \bar{D}_{ij}(t), \quad \forall i, j \in V, \, t \geq 0 \tag{2.20}$$

$$\bar{D}_{ij}(t_2) - \bar{D}_{ij}(t_1) = \sum_{\mathbf{S} \in \mathcal{S}} d_{ij}(\mathbf{S})(\bar{F}_{\mathbf{S}}(t_2) - \bar{F}_{\mathbf{S}}(t_1)), \quad \forall i, j \in V \tag{2.21}$$

$$\bar{Z}_{ej}(t) = \bar{Q}_{\sigma(e)j}(t) - \bar{Q}_{\tau(e)j}(t), \quad \forall e \in E_N, j \in V \tag{2.22}$$

$$\bar{Z}^*_e(t) = \max_{j \in V} \bar{Z}_{ej}(t), \quad \forall e \in E_N \tag{2.23}$$

$$\bar{F}_{\mathbf{S}}(t) \text{ is non-decreasing } \forall \mathbf{S} \in \mathcal{S}, \text{ and } \sum_{\mathbf{S} \in \mathcal{S}} \bar{F}_{\mathbf{S}}(t) = t, \quad t \geq 0 \tag{2.24}$$

Proof: From the strong law of large numbers, we have

$$(A^{r_k}_{ij}(t), t \geq 0) \to (\lambda_{ij} t, t \geq 0) \quad \text{u.o.c.,} \quad \text{w.p.1} \quad \forall i, j.$$

Equations (2.11) and (2.17), and the Lipschitz continuity of $\bar{A}_{ij}$ for all i, j follow.

Note that for any $0 \leq t_1 \leq t_2$, we have

$$(\min\nolimits_{\mathbf{S} \in \mathcal{S}} d_{ij}(\mathbf{S}))\,(t_2 - t_1) \leq D^{r_k}_{ij}(t_2) - D^{r_k}_{ij}(t_1) \leq (\max\nolimits_{\mathbf{S} \in \mathcal{S}} d_{ij}(\mathbf{S}))\,(t_2 - t_1)$$
$$0 \leq F^{r_k}_{\mathbf{S}}(t_2) - F^{r_k}_{\mathbf{S}}(t_1) \leq (t_2 - t_1)$$

Thus the sequence of functions $\{D^{r_k}_{ij}\}$ is uniformly equicontinuous, and since $D^{r_k}_{ij}(0) = 0$, the sequence is also uniformly bounded. Similarly, the sequence $\{F^{r_k}_{ij}\}$ is uniformly bounded and uniformly equicontinuous. Consequently there must exist a subsequence of $\{r_k\}$ for which (2.12) and (2.14) hold. Note also that the above equations imply the Lipschitz-continuity of $\bar{D}_{ij}, \bar{F}_{\mathbf{S}}$.

Applying (2.4), for any fixed $0 \leq t_1 \leq t_2$, and any i, j we have

$$D^{r_k}_{ij}(t_2) - D^{r_k}_{ij}(t_1) = \sum_{\mathbf{S} \in \mathcal{S}} d_{ij}(\mathbf{S})(F^{r_k}_{ij}(t_2) - F^{r_k}_{ij}(t_1)).$$

Thus, there must exist a further subsequence of $\{r_k\}$ under which (2.21) holds. Since $D^{r_k}_{ij}(0) = F^{r_k}_{\mathbf{S}}(0) = 0$, we must have (2.19). Further, we have that F_{r_k} is non-decreasing, with our linear interpolation providing $\sum_{\mathbf{S} \in \mathcal{S}} F^{r_k}_{\mathbf{S}}(t) = t$, from which we conclude that (2.24) holds. Since $Q^{r_k}_{ij}(t) = A^{r_k}_{ij}(t) - D^{r_k}_{ij}(t)$, we have in the limit (2.13) and (2.20). Finally, since $Z^{r_k}_{ej}(t) = Q^{r_k}_{\sigma(e)j}(t) - Q^{r_k}_{\tau(e)j}(t)$ and $Z^{*r_k}_e(t) = \max_{j \in V} Z^{r_k}_{ej}(t)$, we have (2.15) and (2.16). Since $\bar{A}_{ij}, \bar{D}_{ij}, \bar{F}_{\mathbf{S}}$ are Lipschitz continuous, the Lipschitz continuity of $\bar{Q}_{ij}, \bar{Z}_{ej}, \bar{Z}^*_e$ also follows. ■

Note that under Algorithm 1 the following additional properties of the fluid limit functions can be inferred:

$$\text{If } \exists j, j' \in V \text{ with } \bar{Z}_{ej}(t) < \bar{Z}_{ej'}(t) \text{ then } \dot{\bar{F}}_{\mathbf{S}}(t) = 0 \, \forall \mathbf{S} \in \mathcal{S} \text{ such that } S_{ej} > 0; \tag{2.25}$$

$$\text{If } \exists \boldsymbol{\pi}, \tilde{\boldsymbol{\pi}} \in \Pi_N \text{ with } \boldsymbol{\pi}^T \bar{\mathbf{Z}}^*(t) < \tilde{\boldsymbol{\pi}}^T \bar{\mathbf{Z}}^*(t) \text{ then } \dot{\bar{F}}_{\mathbf{S}}(t) = 0 \, \forall \mathbf{S} \in \mathcal{S} \text{ such that } \sum_j \mathbf{S}_{\cdot j} = \boldsymbol{\pi}. \tag{2.26}$$

We are now prepared to present the proof of Theorem 2.3.1. Let $h(t) = (1/2) \sum_{i,j} \bar{Q}_{ij}^2(t)$. Consider a regular time[4] $t \geq 0$ for which $h(t) > 0$. Consider a fluid model solution satisfying (2.17)-(2.26). Denote by $\mathcal{S}'$ the subset of $\mathcal{S}$, where $\mathbf{S} \in \mathcal{S}'$ satisfies

$$S_{ej} > 0 \text{ implies } \bar{Z}_{ej}(t) = \bar{Z}_e^*(t),$$
$$\left(\sum_j \mathbf{S}_{\cdot j} \right)^T \bar{\mathbf{Z}}^*(t) = \max_{\boldsymbol{\pi} \in \Pi_N} \boldsymbol{\pi}^T \bar{\mathbf{Z}}^*(t).$$

By properties (2.25)-(2.26) we must have

$$\sum_{\mathbf{S} \in \mathcal{S}'} \dot{\bar{F}}_{\mathbf{S}}(t) = 1. \tag{2.27}$$

Then,

$$\begin{aligned} \dot{h}(t) &= \sum_{i,j} \bar{Q}_{ij}(t) \dot{\bar{Q}}_{ij}(t) \\ &= \sum_{i,j} \bar{Q}_{ij}(t) \left(\lambda_{ij} - \dot{\bar{D}}_{ij}(t) \right), \\ &= \sum_{i,j} \bar{Q}_{ij}(t) \left(\lambda_{ij} - \sum_{\mathbf{S} \in \mathcal{S}} d_{ij}(\mathbf{S}) \dot{\bar{F}}_{\mathbf{S}}(t) \right). \end{aligned}$$

Recall the definition of an admissible arrival rate matrix from (2.7). Suppose that $\boldsymbol{\lambda} = (\lambda_{ij},\, i, j \in V)$ is admissible. Then, for some nonnegative vector $(\alpha_{\mathbf{S}}, \mathbf{S} \in \mathcal{S})$, where

[4] A regular time is a point at which the system is differentiable. By the Lipschitz continuity of the fluid limit, almost every time in $[0, \infty)$ is regular.

$\sum_{\mathbf{S}\in\mathcal{S}} \alpha_{\mathbf{S}} = 1$, we have

$$\begin{aligned}
\dot{h}(t) \leq & \sum_{i,j} \bar{Q}_{ij}(t) \sum_{\mathbf{S}\in\mathcal{S}} d_{ij}(\mathbf{S}) \left(\alpha_{\mathbf{S}} - \dot{\bar{F}}_{\mathbf{S}}(t)\right), \\
= & \sum_{\mathbf{S}\in\mathcal{S}} \sum_{i,j} \bar{Q}_{ij}(t) \mathbf{R}^{j}_{i\cdot} \mathbf{S}_{\cdot j} \left(\alpha_{\mathbf{S}} - \dot{\bar{F}}_{\mathbf{S}}(t)\right), \\
= & \sum_{\mathbf{S}\in\mathcal{S}} \sum_{j} \left(\bar{\mathbf{Q}}_{\cdot j}(t)\right)^{T} \mathbf{R}^{j} \mathbf{S}_{\cdot j} \left(\alpha_{\mathbf{S}} - \dot{\bar{F}}_{\mathbf{S}}(t)\right), \\
= & \sum_{\mathbf{S}\in\mathcal{S}} \sum_{j} \left(\bar{\mathbf{Z}}_{\cdot j}(t)\right)^{T} \mathbf{S}_{\cdot j} \left(\alpha_{\mathbf{S}} - \dot{\bar{F}}_{\mathbf{S}}(t)\right), \\
\leq & \left(\bar{\mathbf{Z}}^{*}(t)\right)^{T} \sum_{\mathbf{S}\in\mathcal{S}} \sum_{j} \alpha_{\mathbf{S}} \mathbf{S}_{\cdot j} - \sum_{\mathbf{S}\in\mathcal{S}} \sum_{j} \dot{\bar{F}}_{\mathbf{S}}(t) \left(\bar{\mathbf{Z}}_{\cdot j}(t)\right)^{T} \mathbf{S}_{\cdot j}.
\end{aligned}$$

From (2.24) and (2.27), we obtain

$$\begin{aligned}
\dot{h}(t) \leq & \left(\bar{\mathbf{Z}}^{*}(t)\right)^{T} \sum_{\mathbf{S}\in\mathcal{S}} \sum_{j} \alpha_{\mathbf{S}} \mathbf{S}_{\cdot j} - \sum_{\mathbf{S}\in\mathcal{S}'} \sum_{j} \dot{\bar{F}}_{\mathbf{S}}(t) \left(\bar{\mathbf{Z}}_{\cdot j}(t)\right)^{T} \mathbf{S}_{\cdot j}, \\
= & \left(\bar{\mathbf{Z}}^{*}(t)\right)^{T} \sum_{\mathbf{S}\in\mathcal{S}} \sum_{j} \alpha_{\mathbf{S}} \mathbf{S}_{\cdot j} - \left(\bar{\mathbf{Z}}^{*}(t)\right)^{T} \sum_{\mathbf{S}\in\mathcal{S}'} \sum_{j} \dot{\bar{F}}_{\mathbf{S}}(t) \mathbf{S}_{\cdot j}, \\
= & \sum_{\mathbf{S}\in\mathcal{S}} \alpha_{\mathbf{S}} \left(\bar{\mathbf{Z}}^{*}(t)\right)^{T} \left(\sum_{j} \mathbf{S}_{\cdot j}\right) - \sum_{\mathbf{S}\in\mathcal{S}'} \dot{\bar{F}}_{\mathbf{S}}(t) \left(\bar{\mathbf{Z}}^{*}(t)\right)^{T} \left(\sum_{j} \mathbf{S}_{\cdot j}\right), \\
= & \sum_{\mathbf{S}\in\mathcal{S}} \alpha_{\mathbf{S}} \left(\bar{\mathbf{Z}}^{*}(t)\right)^{T} \left(\sum_{j} \mathbf{S}_{\cdot j}\right) - \max_{\mathbf{S}\in\mathcal{S}} \left(\bar{\mathbf{Z}}^{*}(t)\right)^{T} \left(\sum_{j} \mathbf{S}_{\cdot j}\right), \\
\leq & \ 0.
\end{aligned}$$

Using the terminology of [49], we call the above fluid model with $\bar{\mathbf{Q}}(0) = 0$ *weakly stable* if $\bar{\mathbf{Q}}(t) = 0$ for $t \geq 0$. Clearly, since $h(0) = 0$ and $\dot{h}(t) \leq 0$ for every regular t at which $h(t) > 0$, we have that $h(t) = 0$ almost everywhere. Then, we must have that $\bar{\mathbf{Q}}(t) = 0$ almost everywhere and the fluid model is weakly stable. (Similar conclusions are made in [49, Lem. 1] and [146, Lem. 6(i), Thm. 2(i)-(ii)].) We draw the following result from [49] to complete the proof.

Theorem 2.B.2 (Dai and Prabhakar [49, Thm. 3]) *A network operating under a joint routing and scheduling algorithm is rate stable if the corresponding fluid model is weakly stable.*

By the weak stability of the fluid model, and using Theorem 2.B.2, we conclude that the network is rate stable.

Chapter 3

Dynamic reconfiguration and routing algorithms for WDM-based optical networks

In this chapter, we consider scheduling and routing in optical networks employing wavelength division multiplexing (WDM). In particular, we consider the interaction of the electronic layer and the optical WDM layer. Since the electronic layer commonly employs the Internet Protocol (IP) for packet routing, joint consideration of electronic and optical layers is often called IP-over-WDM [61]. We establish a queueing model for the optical networking architecture. We develop throughput-optimal algorithms, based on the backpressure-based algorithm of Tassiulas and Ephremides [150], taking into account delay overheads. Finally, we conduct numerical simulations to evaluate the delay performance of the algorithms.

3.1 Network architecture

We consider an optical networking architecture consisting of nodes having an electronic router overlaying an optical interface, with the nodes interconnected by an optical transport layer. Depicted at the top in Figure 3-1 is an example of our architecture with electronic edge nodes interconnected by an optical transport network using optical fiber links. This constitutes the *physical topology* of the network. Optical transceivers, multiplexers/demultiplexers, wavelength converters, and optical switches allow individual wavelength signals to be either *dropped* to the electronic routers at each node or to pass through the node optically. The *logical topology* consists of the set of all-optical interconnections between the electronic routers and is determined by the configuration of the optical interface at each node [42].[1] Future optical networks will make use of optical bypass, tunable transceivers, optical switches, and wavelength converters in order to harness the full capacity of the optical transport network. The interaction of these optical components with the electronic interface is depicted at the bottom in Figure 3-1.

[1] Logical links are sometimes called virtual links, lightpaths, or MPλS tunnels. Essentially, these are all-optical connections established for a sustained period of time.

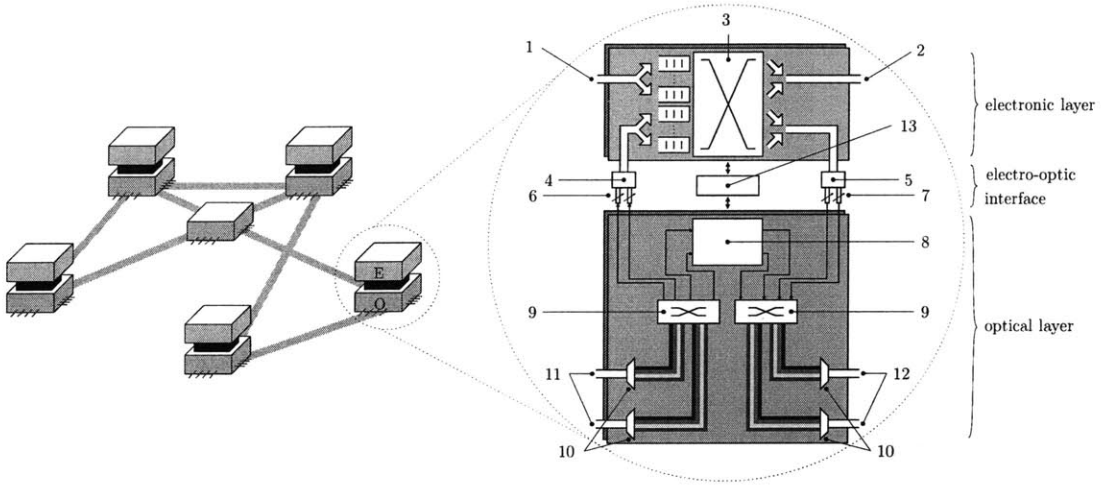


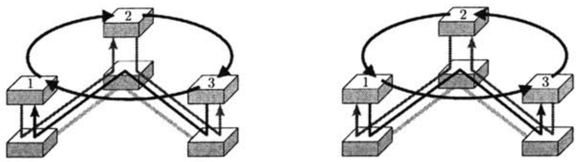
Figure 3-1: Network architecture, with each edge node having the following features: 1–electronic inflows; 2–electronic outflows; 3–electronic packet switch; 4–optical to electronic converter; 5–electronic to optical converter; 6–tunable optical receivers; 7–tunable optical transmitters; 8–wavelength converter; 9–optical switch; 10–optical multiplexer/demultiplexer; 11–incoming fiber; 12–outgoing fiber; 13–controller. The network also includes all-optical nodes providing switching/conversion services to incoming fibers.

Tunable optical components introduce flexibility to optical networks by enabling logical topology *reconfiguration.* As network traffic changes with time, the optimal logical topology varies as well. Consequently, dynamic reconfiguration algorithms can be employed in order to improve the throughput and delay properties of the network, as well as recover from network failures. In essence, a trade-off emerges between lightpath reconfiguration at the WDM layer and routing at the electronic layer. We explore this trade-off in the following example.

3.1.1 Performance trade-off example

Consider a 3-node line network, with a single transceiver per node. The single transceiver constraint implies that each node can source at most one lightpath at any given time and can simultaneously terminate at most one lightpath at any given time. In this example, we assume that transmission of each packet across a lightpath requires one time slot. There are two possible logical configurations that are rings, as depicted in Figure 3-2. This figure shows the lightpath interconnections over the WDM layer (on the bottom) and the resulting ring configuration at the electronic layer (on the top). Note in Figure 3-2(a) that logical link $3 \rightarrow 1$ is established by *optically bypassing* node 2 at the WDM layer.

In an earlier study [109], Narula-Tam and Modiano considered the gains associated with dynamic topology reconfiguration under changing traffic, and designed algorithms for incremental logical topology reconfiguration to balance link loads. If the traffic matrix T



(a) $\mathcal{C}_1$: Ring 1 → 2 → 3. (b) $\mathcal{C}_2$: Ring 1 → 3 → 2.

Figure 3-2: Lightpath interconnections for 3-node rings on a line physical topology.

(corresponding to transmission requests) is given by[2]

$$T = \begin{bmatrix} \cdot & 0 & 1 \\ 1 & \cdot & 0 \\ 0 & 1 & \cdot \end{bmatrix},$$

then by routing the traffic along the clockwise ring, $\mathcal{C}_1$, each logical link experiences a load of 2, while for the counterclockwise ring, $\mathcal{C}_2$, each logical link load is 1. Clearly, the gain achievable by selecting $\mathcal{C}_2$ is a link load reduction by a factor of 2.

In the stochastic setting, where traffic variations are random processes, and the system is subject to reconfiguration delay, packet service delays are affected by the joint algorithm for WDM topology reconfiguration and IP layer packet routing. In this setting, the traffic configuration is characterized by an arrival rate matrix λ, where the entry on the i-th row and j-th column represents the long-term rate of exogenous arrivals of packets to node i destined for node j, in packets per time slot.

To demonstrate the important delay trade-off between incurring reconfiguration overhead and additional load from IP layer routing, consider arrival rate matrix λ_1 under the 3-node network of Figure 3-2,

$$\lambda_1 = \begin{bmatrix} \cdot & 0.2 & 0.5 \\ 0.5 & \cdot & 0.2 \\ 0.2 & 0.5 & \cdot \end{bmatrix}.$$

Under λ_1, if we fix the topology to be $\mathcal{C}_1$, each logical link has long term arrival rate 1.2, which exceeds the maximum service rate[3] of 1.0 for each link. Thus under $\mathcal{C}_1$, the system becomes overloaded with unserviced traffic as time progresses. If $\mathcal{C}_2$ is employed, each logical link experiences a long-term rate of arrivals of 0.9, which is sufficient to guarantee the stability of the network.

It is not always possible to exclusively make use of a single logical topology configuration.

[2] We adopt the convention in this thesis of discarding all diagonal entries in traffic or service matrices. This follows from our assumption of no self-traffic at any node in the network.

[3] The maximum service rate arises because of the single transceiver constraint. Each logical link can service at most one packet per time slot, and no node can source or terminate more than a single logical link. Thus, the maximum service rate is 1.0 packets per time slot.

Consider the following arrival rate matrix, λ_2:

$$\lambda_2 = \begin{bmatrix} \cdot & 0.4 & 0.5 \\ 0.5 & \cdot & 0.4 \\ 0.4 & 0.5 & \cdot \end{bmatrix}.$$

If we service traffic exclusively on $\mathcal{C}_1$, all links experience a long-term arrival rate of 1.4, while if $\mathcal{C}_2$ is exclusively chosen the link arrival rates are each 1.3. In either case, the system becomes overloaded with unserviced traffic as time progresses. However, a TDM schedule using only single-hop routes allocating at least 40% of its time to $\mathcal{C}_1$ and at least 50% of its time to $\mathcal{C}_2$ is sufficient to guarantee that the network is stable, so long as the contiguous service time allocated to each logical ring is adequately long to make the reconfiguration overhead negligible. Because the TDM schedule employs only single-hop routes, this ensures a long term service rate of at least 0.4 packets per time slot to buffers associated with $\mathcal{C}_1$ (buffers for source-destination pairs $(1,2),(2,3),(3,1)$) and a long term service rate of at least 0.5 packets per time slot to buffers associated with $\mathcal{C}_2$ (buffers for source-destination pairs $(1,3),(2,1),(3,2)$).

It is clear that in order to ensure stability and provide excellent delay properties under a broad class of traffic processes, it is essential to balance the idleness associated with reconfiguration against the additional load incurred from multi-hopping along the IP layer.

3.1.2 Related work

The reconfigurable network architecture has been approached in the literature from several angles. Many studies aim to achieve, in some sense, a *balanced* set of link loads [16,83, 84,109]. The work of [83] considers a reconfigurable multi-hop WDM network subject to deterministic non-uniform traffic. The goal of this study is to determine an algorithm for joint reconfiguration and routing with desirable throughput properties. The authors suggest that minimizing the maximum link load (a *minimax* formulation) is an effective means of achieving strong throughput properties. A mixed integer program is provided for the joint optimization, and a heuristic separating the reconfiguration and routing problems and iterating between them is provided. In [84,109], *branch-exchange* algorithms are introduced to *incrementally* adjust the logical topology towards a desired configuration. Here, [84] approaches the problem essentially in a deterministic setting, by considering an initial WDM configuration as well as a fixed target configuration, and seeking a suitable sequence of *two-branch exchanges*[4] to transition between the two configurations with little overall disruption to the network. In [109], the problem is approached under dynamic traffic. This work recognizes that two-branch exchanges may leave the logical topology disconnected, which is undesirable under dynamic traffic, opting instead for *three-branch exchanges*, which are guaranteed to maintain connectivity. The work of [16] associates for each time a cost for reconfiguring the logical topology and a reward that depends on the degree of load balancing for the current logical topology. An average reward *dynamic program* is then formulated

[4] A two-branch exchange tears down two existing logical links $s_1 \rightarrow d_1, s_2 \rightarrow d_2$ and establishes the new logical links $s_1 \rightarrow d_2, s_2 \rightarrow d_1$.

with the total reward at any time equal to a weighted sum of the cost and reward for that particular time.

To the best of our knowledge, the study of stability properties of optical networks was introduced in [127, 128, 158], where the authors considered optical burst scheduling under dynamic traffic in time-domain wavelength interleaved networks. Subsequent work looking at stability properties of optical networks includes: [26, 27], where scheduling algorithms were introduced for joint electronic routing and WDM layer reconfiguration under a variety of practical optical layer constraints; and [154, 155], where the stability properties of optical burst, flow, and packet switched architectures were compared.

In recent years, tremendous efforts have been made in the research towards so-called "IP-over-WDM" networks. These studies aim to improve network performance through increased electro-optical integration [58, 61, 119, 134, 153, 165]. Several studies consider Optical Burst Switching (OBS) as the mechanism for accessing the optical transport layer [119, 127, 128, 158, 162, 166]. Most solutions seek to integrate IP and Generalized Multiprotocol Label Switching (GMPLS) functionality. Our work differs from existing studies on electro-optical integration in that we are not tied to a particular protocol suite, but rather employ a "generic" architecture utilizing electronic packet switching along with a reconfigurable optical transport layer. Our approach is to determine the fundamental performance characteristics achievable in general reconfigurable optical networks having varying topology and processing functionalities.

3.1.3 Summary of contributions

In [109], logical topology reconfiguration was initiated at regular intervals in order to deal with changing traffic. Furthermore, the reconfigurations were incremental, and made no guarantees about the stability of the system. In this chapter, we provide the first systematic approach to the dynamic reconfiguration and routing problem under stochastic traffic in the presence of reconfiguration overhead. We determine stable algorithms employing IP layer routing in order to elicit an understanding of the performance trade-offs between reconfiguration at the optical layer and packet routing at the IP layer. Our major contributions are:

1. We develop mechanisms for dynamically triggering WDM reconfiguration under stochastic traffic. Our algorithms are based on maximum weight scheduling decisions, and specify precisely when and how to reconfigure the WDM layer as well as the IP routing employed between reconfigurations.

2. We demonstrate the asymptotic throughput optimality of our *frame-based* algorithms in the presence of reconfiguration overhead.

3. For multiple transceivers per node, we demonstrate the stability region by providing a novel algorithm extending Birkhoff-von Neumann matrix decompositions to this setting.

4. Using delay as a performance metric, we employ simulations to demonstrate the important trade-off between WDM reconfiguration and IP layer routing. Our simula-

tions point to the advantage of packet switching at low throughput levels and circuit switching at high throughput levels.

3.2 Reconfigurable network model

Here we provide the details of the optical network model of interest. We will use the variables and terminology introduced in Chapter 2.

We consider a reconfigurable WDM-based packet network N, consisting of n nodes (the set of nodes is V). The network symbol N refers to all physical aspects of the optical data network, including the physical topology of the network, the number of wavelengths available in each fiber link, and the number of transceivers (or ports) at each node. We assume that node $v \in V$ has P_v transceivers. The network nodes are interconnected by optical fiber, with each fiber having a single (usually bidirectional) wavelength available for transmission of data. Let $G_P = (V, E_P)$ be the directed *physical topology graph* of the network N: if there exists a fiber between nodes $v_1, v_2 \in V$ along which data can travel from node v_1 to v_2, then the directed edge (v_1, v_2) belongs to E_P.

A direct optical communication link between two nodes is called a *logical link* or a *lightpath.* Such a link consists of an all-optical path through the network N, connecting the nodes, possibly traversing multiple intermediate nodes, with no intermediate electronic processing (see for example the straight-edge links depicted in Figure 3-2). The edges of the *directed graph* $G_N = (V, E_N)$ represent the set of *logical links* that can be enabled in the network. Denote $m = |E_N|$. In general these logical links may not be able to be activated simultaneously, but resources exist to *at least* allow each link to be active individually. We assume that a lightpath can exist between any two nodes, which implies that G_N is a complete graph. At any time, the network may initiate a logical topology reconfiguration, under which existing lightpaths are torn down and new ones are set up.

Since G_N is a complete graph, for several symbols in our study it will be convenient to alternate between understanding the symbol as representing a vector or a matrix. This will always arise in the context of collections of symbols representing the possible source-destination pairs in the network. For example, we denote the collection of queue backlogs at time $t \geq 0$ as $\mathbf{Q}(t)$, which can be understood as a matrix, $\mathbf{Q}(t) = (Q_{vv'}(t), v, v' \in V)$, or as a vector, $\mathbf{Q}(t) = (Q_e(t), e \in E_N)$. These definitions are interchangeable, since we attach no meaning to diagonal entries of the matrix $\mathbf{Q}$. The other symbol that we will treat in this manner is $\boldsymbol{\lambda}$, the collection of exogenous arrival rates.

The set Π_N denotes the collection of feasible logical topologies in the network: the matrix $\boldsymbol{\pi} = (\pi_{ij}, i, j \in V) \in \Pi_N$ is a nonnegative integer matrix, where π_{ij} is the number of active logical links from node i to node j. Clearly, Π_N is constrained by the wavelength/port limitations of the network. We refer to a WDM network as *wavelength-unconstrained* when there exist sufficiently many wavelengths to allow any arbitrary logical interconnection of nodes subject to the port constraints.

Example 3.2.1 *Consider the case of a single port per node ($P_v = 1, \forall v \in V$), and assume that the network is wavelength-unconstrained. In this case, the set of $n \times n$ (sub)permutation matrices (with discarded diagonal entries) is in one-to-one correspondence with Π_N. This* *follows because $P_v = 1$ implies that no node can originate or terminate more than one lightpath. Consider a $n \times n$ (sub)permutation matrix. By letting entry (i, j) of the matrix correspond to a lightpath from node i to node j, we see clearly that at most one lightpath can originate or terminate at each node.*

We consider two levels of IP layer electronic routing capability: single-hop and multi-hop routing. Each of these mechanisms has a different set of admissible service activations, detailed in Chapter 2, and denoted $\mathcal{S}^{\text{sh}}, \mathcal{S}^{\text{mh}}$, respectively, as well as single-hop and multi-hop capacity regions, $\mathbf{\Lambda}^*_{\text{sh}}, \mathbf{\Lambda}^*_{\text{mh}}$, respectively.

As in Chapter 2, packets are assumed to have fixed size, with transmission duration of one slot. This assumption is for simplicity of exposition and can be relaxed with appropriate envelope algorithms [75]. The network allows a maximum of one packet to be transmitted across any logical link during a slot. At any time, the network may initiate a logical topology reconfiguration, under which existing lightpaths are torn down and new ones re-established to form a new logical topology. Transceivers that are tuned are forced to be idle for the reconfiguration time of δ slots, while links that are unaffected may continue to service traffic during reconfiguration. The queueing variables comprising the queue evolution equations of (2.3)-(2.6) apply to this system without loss of generality.

3.2.1 Scheduling under tuning latency, propagation delay, and distributed control

Since we are operating in a distributed mesh network environment, it may not be practical to assume that each node is synchronized to a common clock. A key aspect of the reconfiguration and routing algorithms in our packet-based WDM network is that they employ frame-based scheduling, where logical links are held fixed over *data intervals*, and the logical topology is changed over *reconfiguration intervals*. A *frame boundary* occurs at the instant when the network initiates the sequence of controls to reconfigure the logical topology. This sequence includes: 1) the time for the final packets of the terminated frame to arrive at their respective destinations, t_p (can be taken as a fixed value if we bound the delay over all possible logical links); 2) the time for information exchange in order to make a decision about the new logical topology to configure, t_c (this information exchange may have occurred prior to the frame boundary, in which case $t_c = 0$); and 3) the time for tuning the transceivers to establish a new logical topology, t_r. The value of t_p depends on the underlying fiber plant topology of the network, which in the case of WAN's is on the order of 10's of milliseconds. The value of t_r depends on the transceiver technology, with current components requiring on the order of 10's of milliseconds for reconfiguration. Thus, we designate the reconfiguration overhead $\delta = t_p + t_c + t_r$.

Using tools from standard clock synchronization algorithms [99], each node can be made aware of a common time reference. Rather than requiring that the electronics at each node be synchronized to this common reference, the reference is used to make nodes aware of frame boundaries. In the case of variable frame durations, this reference can be used to establish agreement between the nodes about each successive frame boundary. The frame boundary is initialized by having each node stop transmission of packets after the complete

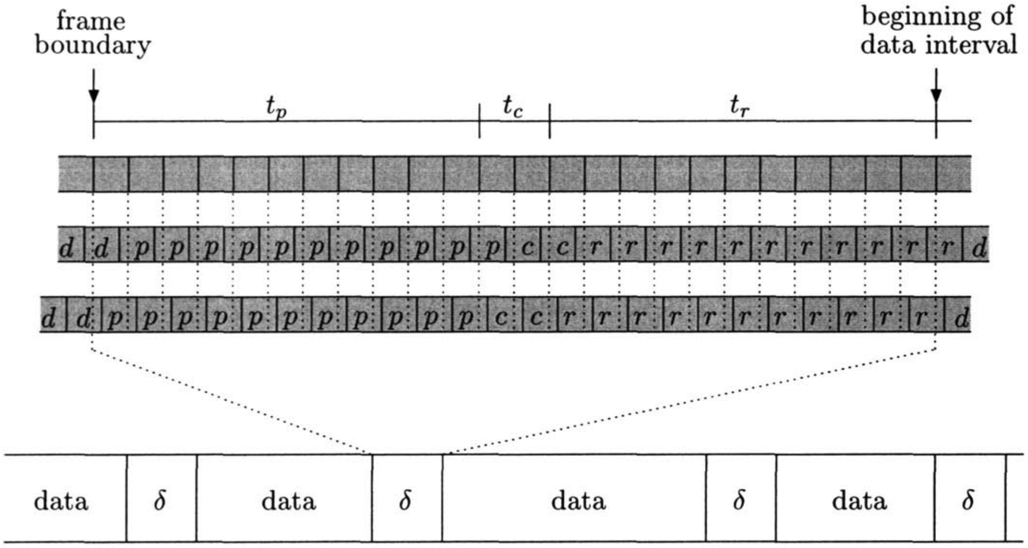

frame boundary
beginning of data interval
t_p
t_c
t_r
d d p p p p p p p p p p c c r r r r r r r r r r d
d d p p p p p p p p p p c c r r r r r r r r r r d
data
δ
data
δ
data
δ
data
δ

Figure 3-3: A reconfiguration interval is used to change the logical topology. The interval consists of t_p slots for propagation delay of the final packets of the last data interval (slots labeled p), t_c slots for passing control information in order to decide on a new logical topology (slots labeled c), and t_r slots to tune the transceivers and establish the new logical topology (slots labeled r). Slots labeled d are slots for packet transmission (corresponding to a data interval). The top sequence of slots corresponds to a common time reference according to which frame boundaries are set. The second and third sequences of slots correspond to distinct nodes in the network. As illustrated, these slots need not be synchronized to each other or to the common time reference. The frame-based scheduling is depicted at bottom, with δ used to indicate the reconfiguration interval of duration δ, and data used to indicate the data interval.

transmission of any packet being serviced at that time. We have illustrated the structure of a reconfiguration interval in Figure 3-3.

3.3 Algorithms for asymptotic throughput optimality

In Chapter 2, we detailed a general version of the algorithm of Tassiulas and Ephremides, originally introduced in [150]. The algorithmic description for scheduling in this network setting involves *maxweight decisions*, where each network configuration has associated with it a particular weight, and the maximum weighted configuration is chosen at each time. Here, we introduce two versions of this algorithm, specialized to general reconfigurable WDM-based networks. Our algorithms are valid under arbitrary wavelength/port constraints.

We begin by considering the case of no reconfiguration delay ($\delta = 0$), and introduce single-hop and multi-hop algorithms for joint WDM reconfiguration and electronic layer routing. Subsequently, for $\delta > 0$, we prove that *any* stable algorithm for the case of $\delta = 0$ may be transformed into a *frame-based* algorithm that stabilizes the network. Furthermore, we introduce a *bias-based* algorithm that makes reconfiguration decisions by taking into account the current logical topology of the network. These algorithms are a natural extension of maxweight scheduling algorithms to the case $\delta > 0$.

3.3.1 Single-hop maxweight scheduling algorithm, for $\delta = 0$

The single-hop maxweight scheduling algorithm (Algorithm SHMW) employs WDM reconfiguration and single-hop electronic layer routing. In other words, if a directed logical link exists connecting node i to node j at time t, then that link can only be used at time t to service packets at node i that are destined for node j. At time t, the algorithm selects a logical topology from Π_N whose inner product with the queue backlog vector $\mathbf{Q}(t)$ is maximum. This logical topology is used for single-hop routing of packets to their destinations. Algorithm SHMW is detailed next.

Algorithm 4 Single-hop maxweight scheduling algorithm (SHMW)

```
1: for time $t \geq 0$ do
2:     Obtain a maximum weight WDM logical configuration $\pi^* = (\pi_e^*, e \in E_N)$, using
           $\pi^* \in \arg\max_{\pi \in \Pi_N} \pi^T \mathbf{Q}(t),$
       where $\mathbf{Q}(t) = (Q_{\sigma(e)\tau(e)}(t), e \in E_N)$. Reconfigure the WDM network to this configuration
3:     Route $\min\{\pi_e^*, Q_{\sigma(e)\tau(e)}(t)\}$ packets of commodity $\sigma(e)$ from node $\sigma(e)$ to $\tau(e)$
4: end for
```

Note in step 3 that the number of packets routed is the minimum of π_e^*, which is the number of active logical links from node $\sigma(e)$ to $\tau(e)$, and $Q_{\sigma(e)\tau(e)}(t)$, which is the number of packets in queue awaiting service across edge e. Our result concerning the throughput optimality of Algorithm 1 can be applied to demonstrate that SHMW is stable over the region Λ_{sh}^*. We present this result next.

Corollary 3.3.1 *Algorithm SHMW achieves the single-hop capacity region:* $\Lambda_{\text{SHMW}} = \Lambda_{\text{sh}}^*$.

Proof: This result can be derived as an immediate consequence of [146, Lem. 5] and [49, Thm. 3]. Alternatively, our proof of Theorem 2.3.1 can be enlisted to demonstrate this result, by redefining

$$Z_{ej}(t) = \begin{cases} Q_{\sigma(e)\tau(e)}(t), & \text{if } j = \tau(e), \\ 0, & \text{otherwise}, \end{cases}$$

and assigning $Z_e^*(t) = \max_{j \in V} Z_{ej}(t)$. ■

Recall from example 3.2.1 that when the network has a single transceiver per node, and no wavelength constraint, the set Π_N is in direct correspondence with the set of permutation and subpermutation matrices (with diagonal entries discarded). Thus, Λ_{SHMW} corresponds to the convex hull of the (sub)permutation matrices, which is identical to the doubly substochastic region:

$$\Lambda^*_{\text{sh}} = \left\{ \lambda \in \mathbb{R}_+^{n \times n} : \sum_j \lambda_{i,j} \leq 1, \forall i, \ \sum_i \lambda_{i,j} \leq 1, \forall j \right\}. \tag{3.1}$$

For context, note that this is also the admissible region of an input-queued switch [97].

3.3.2 Multi-hop backpressure based algorithm, for $\delta = 0$

The multi-hop maxweight scheduling algorithm (Algorithm MHMW) employs WDM reconfiguration and multi-hop electronic layer routing. MHMW is equivalent to Algorithm 1. We present MHMW below, only modifying some of the terminology from Algorithm 1.

Algorithm 5 Multi-hop maxweight scheduling algorithm (MHMW)

```
for time $t \geq 0$ do
    For each available directed logical link $e \in E_N$ assign
        $Z_{ej}(t) \leftarrow (Q_{\sigma(e)j}(t) - Q_{\tau(e)j}(t))$
    Assign $Z^*_e(t) \leftarrow \max_j Z_{ej}(t)$
    Obtain a maximum weight WDM logical configuration $\boldsymbol{\pi}^* = (\pi^*_e, e \in E_N)$, where
        $\boldsymbol{\pi}^* \in \arg\max_{\boldsymbol{\pi} \in \Pi_N} \boldsymbol{\pi}^T \mathbf{Z}^*(t),$
    and reconfigure the WDM network to this configuration
    For each logical link $e$ where $\pi^*_e \geq 1$, choose a commodity $j^* \in \arg\max_j Z_{ej}(t)$.
    Electronically route $\min\{\pi^*_e, Q_{\sigma(e)j^*}(t)\}$ packets of commodity $j^*$ across the logical
    links from node $\sigma(e)$ to $\tau(e)$
end for
```

Since MHMW and Algorithm 1 are technically identical, we can immediately conclude that MHMW achieves 100% throughput.

Corollary 3.3.2 *Algorithm MHMW achieves 100% throughput:* $\Lambda_{\text{MHMW}} = \Lambda^*_{\text{mh}}$.

3.3.3 Frame-based scheduling framework for $\delta > 0$

Although algorithms SHMW and MHMW are specifically defined for the case $\delta = 0$, it is intuitively clear that they can be adapted to the case of $\delta > 0$ using *frame*-based schemes, where reconfiguration decisions are only made at frame boundaries. In this section, we formalize this idea by providing a result showing that the stability region achieved by these algorithms for $\delta = 0$ can be asymptotically achieved using frame-based versions of the algorithms when $\delta > 0$. The frame-based scheduling framework makes use of a frame interval $I_f \in \mathbb{Z}_+$, with a WDM topology reconfiguration decision made every I_f time slots.

The frame-based scheduling framework alternates regularly between idle and service intervals, as illustrated in Figure 3-4. The algorithm operates as follows: at each frame boundary, under backlog matrix $\mathbf{Q}$, the frame-based scheduling algorithm makes the same

Algorithm 6 Frame-based scheduling framework applied to algorithm SHMW/MHMW

```
for time slots $\{kI_f, kI_f + 1, \dots, (k+1)I_f - 1\}$, where $k \in \mathbb{Z}_+$ do
    At time $kI_f$, make a WDM reconfiguration decision according to algorithm SHMW/MHMW
    Idle through the reconfiguration interval $\{kI_f, \dots, kI_f + \delta - 1\}$
    At each time slot in $\{kI_f + \delta, \dots, (k+1)I_f - 1\}$, make an electronic routing decision according to algorithm SHMW/MHMW, subject to the fixed WDM topology configuration selected at time $kI_f$
end for
```

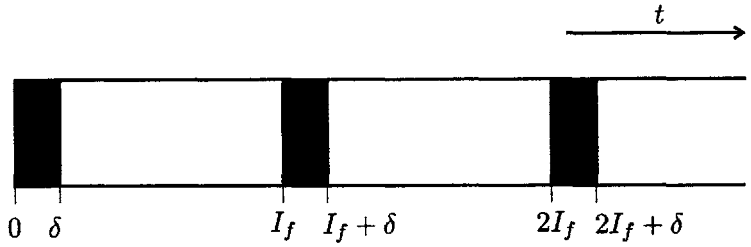


Figure 3-4: The regular on-off nature of the frame-based algorithm.

WDM reconfiguration decision that SHMW/MHMW makes under backlog **Q**. Note in step 3, the algorithm requires that the system remains completely idle while WDM reconfiguration is conducted. This could be improved to allow packets to traverse links that are not affected by the WDM reconfiguration decision. We do not consider this improved policy in the following stability analysis. The remainder of the frame is devoted to servicing packets over the fixed WDM configuration according to the maxweight decisions of algorithm SHMW/MHMW, whichever is being employed.

We next demonstrate the asymptotic throughput optimality of the frame-based scheduling framework. Our proof makes use of the *throughput parameter* $\varepsilon^*(\boldsymbol{\lambda})$, defined as follows:

$$\begin{aligned} \varepsilon^*(\boldsymbol{\lambda}) = \max & \left(1 - \sum_{\mathbf{S} \in \mathcal{S}} \alpha_{\mathbf{S}}\right) \\ \text{subject to } & \lambda_{ij} \leq \sum_{\mathbf{S} \in \mathcal{S}} \alpha_{\mathbf{S}} d_{ij}(\mathbf{S}), \quad \forall i,j \in V \\ & \sum_{\mathbf{S} \in \mathcal{S}} \alpha_{\mathbf{S}} \leq 1 \\ & \alpha_{\mathbf{S}} \geq 0, \quad \forall \mathbf{S} \in \mathcal{S} \end{aligned}$$

The variable $\varepsilon^*(\boldsymbol{\lambda})$ can be considered a measure of the "distance" of the rate vector $\boldsymbol{\lambda}$ from the outer boundary of the capacity region. As an example, if $\boldsymbol{\lambda}$ is organized as a rate matrix, then if the network N has a single port per node and no wavelength constraint, $\varepsilon^*(\boldsymbol{\lambda})$ equals the difference between the maximum row/column sum of $\boldsymbol{\lambda}$ and 1:

$$\varepsilon^*(\boldsymbol{\lambda}) = 1 - \max\left\{\max_{i \in V} \sum_{j \in V} \lambda_{ij}, \max_{j \in V} \sum_{i \in V} \lambda_{ij}\right\} \tag{3.2}$$

Theorem 3.3.1 *Consider arrival rate vector $\boldsymbol{\lambda} \in \Lambda^*$, and suppose $\varepsilon^*(\boldsymbol{\lambda}) > 0$. Then, the frame-based version of algorithm MHMW is stable for any arrival process having rate vector $\boldsymbol{\lambda}$, so long as the frame interval satisfies $I_f \geq \delta/\varepsilon^*(\boldsymbol{\lambda})$.*

Proof: See Appendix 3.A. Subsequent to the proof, Appendix 3.B provides a simple alternative demonstration of stability, for a frame-based scheduler that employs a simple batching algorithm. ■

Since Theorem 3.3.1 applies for any $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*_{\text{mh}}$ satisfying $\varepsilon^*(\boldsymbol{\lambda}) > 0$, we say that the frame version of MHMW is *asymptotically throughput optimal.* The asymptotic throughput optimality of the frame-based version of SHMW follows in a similar manner, just as the stability of SHMW followed from that of MHMW under $\delta = 0$ in Corollary 3.3.1. As for the case of the frame version of MHMW, define the single-hop throughput parameter $\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})$ according to:

$$\begin{aligned} \varepsilon^*_{\text{sh}}(\boldsymbol{\lambda}) = \max & \left(1 - \sum_{\mathbf{S} \in \mathcal{S}^{\text{sh}}} \alpha_{\mathbf{S}}\right) \\ \text{subject to } & \lambda_{ij} \leq \sum_{\mathbf{S} \in \mathcal{S}^{\text{sh}}} \alpha_{\mathbf{S}} d_{ij}(\mathbf{S}), \quad \forall i, j \in V \\ & \sum_{\mathbf{S} \in \mathcal{S}^{\text{sh}}} \alpha_{\mathbf{S}} \leq 1 \\ & \alpha_{\mathbf{S}} \geq 0, \quad \forall \mathbf{S} \in \mathcal{S}^{\text{sh}}. \end{aligned}$$

Similar to the multi-hop case, $\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})$ provides a measure of the "distance" of $\boldsymbol{\lambda}$ from the outer boundary of the single-hop admissible region $\boldsymbol{\Lambda}^*_{\text{sh}}$.

Corollary 3.3.3 *Consider arrival rate vector $\boldsymbol{\lambda} \in \Lambda^*_{\text{sh}}$, and suppose $\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda}) > 0$. Then, the frame-based version of algorithm SHMW is stable for any arrival process having rate vector $\boldsymbol{\lambda}$, so long as the frame interval satisfies $I_f \geq \delta/\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})$.*

Proof: The proof follows similarly to that of Theorem 3.3.1, only that service activations are limited to the set $\mathcal{S}^{\text{sh}}$. The only necessary modification to the proof of Theorem 3.3.1 is to redefine

$$Z_{ej}(t) = \begin{cases} Q_{\sigma(e)\tau(e)}(t), & \text{if } j = \tau(e), \\ 0, & \text{otherwise,} \end{cases}$$

and assign $Z^*_e(t) = \max_{j \in V} Z_{ej}(t)$. ■

Recall that Example 3.2.1 focused on the case of no wavelength constraint, where there are sufficiently many wavelengths available to allow configuration of any logical topology subject to the port constraint. Our next result again looks at this scenario. We find that the single-hop and multi-hop admissible regions are equal, which implies that when there is no wavelength constraint, algorithms SHMW and MHMW both achieve 100% throughput. The port constraint implies that the following set is the multi-hop admissible region $\boldsymbol{\Lambda}^*_{\text{mh}}$:

$$\boldsymbol{\Lambda}^*_{\text{mh}} = \left\{ \boldsymbol{\lambda} \in \mathbb{R}_+^{n \times n} : \sum_{j \in V} \lambda_{ij} \leq P_i \, \forall i, \sum_{i \in V} \lambda_{ij} \leq P_j \, \forall j \right\}$$

Theorem 3.3.2 *For a WDM network having no wavelength constraint and port distribution $(P_v, v \in V)$, the multi-hop admissible rate region Λ^* equals the convex hull of the link activation set Π_N: $\Lambda^* = \mathrm{conv}(\Pi_N)$.*

Proof: See Appendix 3.C. ■

The following corollaries result immediately from Theorem 3.3.2 and its proof.

Corollary 3.3.4 *For a WDM network having no wavelength constraint, the single-hop capacity region equals the multi-hop capacity region: $\mathbf{\Lambda}^*_{\text{sh}} = \mathbf{\Lambda}^*_{\text{mh}}$. Consequently, algorithms SHMW and MHMW both achieve 100% throughput when $\delta = 0$, and the frame versions of algorithms SHMW and MHMW are both asymptotically throughput optimal.*

Proof: From Theorem 3.3.2, we have that $\mathbf{\Lambda}^*_{\text{mh}} = \mathrm{conv}(\Pi_N)$. The result then follows because $\mathbf{\Lambda}^*_{\text{sh}} = \mathrm{conv}(\Pi_N)$. ■

The next corollary attempts to gain a sense of the frame interval required in stabilizing implementations of frame versions of SHMW and MHMW, when the WDM network has no wavelength constraint. Interestingly, we find that the single-hop and multi-hop throughput parameters are equal in this case, which implies that the frame intervals sufficient for stability in Theorem 3.3.1 and Corollary 3.3.3 are equal.

Corollary 3.3.5 *For a WDM network having no wavelength constraint and port distribution $(P_v, v \in V)$, and given $\boldsymbol{\lambda} \in \mathbf{\Lambda}^*_{\text{mh}}$,*

$$\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda}) = \varepsilon^*(\boldsymbol{\lambda}).$$

Proof: See Appendix 3.D. ■

While the maxweight scheduling mechanism we have proposed for WDM reconfiguration and packet routing depends upon local traffic variations, the use of a deterministic frame interval does not take traffic conditions into account. The focus of the next section is on building a frame-based scheduling framework, where the frame interval is of varying duration, based upon the local traffic conditions.

3.3.4 Additive bias-based scheduling framework

In this section, we introduce the additive bias-based scheduling framework, which provides asymptotic throughput optimality for any $\delta > 0$. Here we assume that the dissemination of control information across the network is sufficiently fast such that every node is aware of the backlog matrix at each slot. Thus, this class of algorithms is also well suited for scheduling crossbar switches with reconfiguration overhead.

The additive bias-based scheduling framework is provided below. The intuition behind the algorithm is that every decision to reconfigure should be followed by some opportunity to service packets under the logical topology selected (in essence, the algorithm has a built-in hysteresis). Under the framework, WDM reconfiguration decisions are made at each time slot. The only difference is that the weight associated with the *existing* logical topology prior

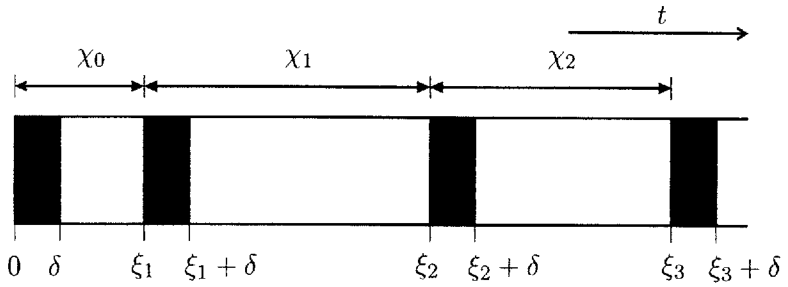


Figure 3-5: The service intervals of the additive bias-based algorithm.

to the decision instant is *biased* additively by the constant number b. This bias is chosen in such a way as to increase the expected time interval between WDM reconfiguration decisions sufficiently to ensure stability of the system for $\delta > 0$.

Algorithm 7 Additive bias-based scheduling framework applied to algorithm SHMW/MHMW

```
for time $t \in \mathbb{Z}_+$ do
  if the WDM layer is not in the process of reconfiguration then
    Denote the existing logical configuration by $\boldsymbol{\pi}(t)$. Select logical configuration $\boldsymbol{\pi}^* \in \Pi_N$ according to
    $$\boldsymbol{\pi}^* \in \begin{cases} \arg\max_{\boldsymbol{\pi}\in\Pi_N} b\mathbb{1}_{\{\boldsymbol{\pi}=\boldsymbol{\pi}(t)\}} + \boldsymbol{\pi}^T\mathbf{Q}(t), & \text{for algorithm SHMW} \\ \arg\max_{\boldsymbol{\pi}\in\Pi_N} b\mathbb{1}_{\{\boldsymbol{\pi}=\boldsymbol{\pi}(t)\}} + \boldsymbol{\pi}^T\mathbf{Z}^*(t), & \text{for algorithm MHMW} \end{cases}$$
    if $\boldsymbol{\pi}^* = \boldsymbol{\pi}(t)$ then
      Route packets across the current logical configuration according to the rules of the algorithm (SHMW or MHMW)
    else
      Initiate WDM reconfiguration to logical topology $\boldsymbol{\pi}^*$
      continue
    end if
  end if
end for
```

Figure 3-5 illustrates the intervals associated with service and reconfiguration phases of bias-based scheduling. As opposed to the frame-based scheduling policies, the service intervals are of variable duration. We denote by ξ_k the k-th reconfiguration decision instant, with $\xi_0 \triangleq 0$, and $\chi_k \triangleq \xi_{k+1} - \xi_k$.

The following lemma establishes a sufficient condition for the stability of the bias-based scheduling framework. The result makes use of the fluid limit function $\bar{F}_\delta(t)$ corresponding to the process $F_\delta(t)$, which tracks the cumulative time up to and including time t spent idle during reconfiguration intervals. This process was introduced in the proof of Theorem 3.3.1 in Appendix 3.A.

Lemma 3.3.1 *Consider an arrival process with arrival rates $\boldsymbol{\lambda} \in \Lambda^*_{\mathrm{mh}}$. If the fluid limit* *process $\bar{F}_\delta(t)$ satisfies $\dot{\bar{F}}_\delta(t) \leq \varepsilon^*(\boldsymbol{\lambda})$ for all $t \in \mathbb{R}_+$, then the additive-based based version of MHMW stabilizes the network.*

Similarly, for an arrival process with arrival rates $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*_{\text{sh}}$, if the fluid limit process $\bar{F}_\delta(t)$ satisfies $\dot{\bar{F}}_\delta(t) \leq \varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})$ for all $t \in \mathbb{R}_+$, then the additive-based based version of SHMW stabilizes the network.

Proof: The proof follows similarly to that of Theorem 3.3.1. The details can be found in Appendix 3.E. ■

Note that for $\delta = 0$, Lemma 3.3.1 immediately implies that the additive bias-based versions of SHMW and MHMW are stable, since zero time is lost to reconfiguration and thus $\bar{F}_\delta(t) = 0$ for all t. For $\delta > 0$ we now use Lemma 3.3.1 to prove the stability of a network having a single port per node and no wavelength constraint, under any joint Bernoulli arrival process.

Theorem 3.3.3 *Consider a WDM network with a single port per node, and no wavelength constraint, subject to a Bernoulli arrival process (not necessarily independent or identically distributed in time or across VOQ's) with rates $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*_{\text{sh}}$, where $\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda}) > 0$. If b is chosen to satisfy $b/n \geq 2(\delta/\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})) - \delta$, then the bias-based version of SHMW stabilizes the reconfigurable queueing network.*

Similarly, if $\lambda \in \boldsymbol{\Lambda}^*_{\text{mh}}$ with $\varepsilon^*(\boldsymbol{\lambda}) > 0$, and b is chosen to satisfy $b/n \geq 6(\delta/\varepsilon^*(\boldsymbol{\lambda})) - 3\delta$, then the bias-based version of MHMW stabilizes the reconfigurable queueing network.

Proof: See Appendix 3.F. ■

3.3.5 Comments regarding frame-based scheduling

The fixed and variable frame-based scheduling frameworks we have proposed in this section suffer a few drawbacks. First, our sufficiency conditions for stability are a function of the traffic statistics, which in general are unknown. There are several approaches to dealing with this issue:

1. the system's arrival rates can be estimated periodically, and the frame interval/additive bias adjusted accordingly;

2. a selective packet dropping mechanism can be implemented at the input and output ports, in order to guarantee a minimum value of ε^* or $\varepsilon^*_{\text{sh}}$.

A second drawback of frame-based scheduling was mentioned earlier: the system as defined does not allow packets to transfer across links that are not torn down at frame boundaries. Clearly, such a restriction need not apply in a true implementation of a fixed or variable frame-based scheduling framework.

A third drawback of the system as we have modeled it is that packets are of fixed size. Any true implementation would have to eliminate such an assumption. Fortunately, the frame-based scheduling frameworks easily admit variable-length packets, by allowing transmission of any enqueued data through to the end of the frame interval, possibly terminating transmission early in the frame when there is no packet available that can be transmitted in the remainder of the frame interval. Thus, a frame-based scheduler can be thought of as an "envelope algorithm", much like that presented in [75].

3.4 Delay performance studies

In this section, we compare the delay performance of algorithms under different traffic conditions, reconfiguration overheads, and physical topologies. Our simulations demonstrate that there exists a strong advantage to employing multi-hop routing at the IP layer under certain conditions. In particular, when there is a single transceiver per node, multi-hop routing is advantageous at low throughput levels. Also, we observe the tremendous advantage of employing mutli-hop routing in an access network scenario, where a single *hub* node has n transcievers and each of the other *local* nodes is equipped with a single transceiver.

When considering the system at the packet level, a relevant performance metric is the average service delay experienced by packets in the system. Through a straightforward application of Little's formula, the average service delay is tied to the time average aggregate queue backlog. When the WDM network is subject to an arrival process with rates $\boldsymbol{\lambda}$, and employs scheduling policy P, the time average delay is given by

$$\frac{1}{\sum_{i,j}\lambda_{ij}} \limsup_{T\to\infty} \frac{1}{T} E\left[\sum_{t=0}^{T-1}\sum_{i,j} Q_{ij}^{P}(t)\right],$$

where $\mathbf{Q}^P(t) = (Q_{ij}^P(t), i, j \in V)$ is the queue backlog matrix at time t under algorithm P. It turns out that quantifying the average delay is difficult, because of the widely varying collection of allowable traffics that have the same arrival rates. Using the theory of Lyapunov stability, the authors of [87] derive bounds on average queue occupancy (and consequently on average delay), that achieve varying degrees of tightness, depending on how correlated different arrival streams are. For this reason, this section makes use of both theory and numerical results to arrive at our conclusions.

In gigabit networks, reconfiguration delay intervals on the order of $\delta = 1,000$ to $\delta = 50,000$ time slots are reasonable values. In this section, we provide data for the case $\delta = 1,000$, though our tests for larger values of δ yield identical conclusions.

3.4.1 Zero reconfiguration delay ($\delta = 0$)

For $\delta = 0$ it is unknown whether in fact there exists any benefit to IP layer routing. We begin by showing that for $n = 3$, in the simple case of a single port per node, and no wavelength constraint, each algorithm employing packet forwarding is no better than an associated algorithm that never forwards packets.

Theorem 3.4.1 *For a WDM network having $n = 3$ nodes, a single port per node, and no wavelength constraint, any algorithm employing multi-hop routing has an associated algorithm that does not multi-hop packets with an equal or lower average aggregate backlog when $\delta = 0$, for any joint arrival distribution.*

Proof: See Appendix 3.G. ■

Essentially, we may conclude that for $n = 3$, when there is no reconfiguration overhead, there is *no benefit* from treating such a system system as *more than a switch.* For $n > 3$, it is not possible to generalize Theorem 3.4.1 directly to conclude that packet forwarding is not beneficial with respect to average delay. We leave this as an interesting open problem for future study.

3.4.2 Overview of algorithms tested

We compare several algorithms for joint WDM topology reconfiguration and IP layer routing. The algorithms are frame or bias-based versions of the following:

1. SHMW;

2. MHMW;

3. Prioritized Backpressure: This algorithm makes decisions according to MHMW for reconfiguration and routing, but priority is given to servicing packets single-hop across active links;

4. MW Minhop: This algorithm makes WDM topology reconfiguration decisions identically to SHMW, and then applies minhop routing at the electronic layer.

The algorithms Prioritized Backpressure and MW Minhop have not been introduced until now. They are heuristic algorithms that we devised in order to test the delay properties of SHMW and MHMW. Prioritized Backpressure operates on the philosophy that once MHMW has chosen a logical topology, it seems reasonable to transmit those packets that are one hop from departure prior to the multihop packets scheduled by MHMW. Thus, Prioritized Backpressure uses MHMW for joint logical topology reconfiguration decisions and IP layer routing, with the caveat that any nonempty VOQ's one hop from departure are serviced with priority.

Given δ, in our simulations we choose a frame size 10% in excess of the minimum value required for stability, in order to mitigate the probability of large deviations in the queue occupancies in our numerical simulations.

3.4.3 Circuit versus packet switching

It is certainly true that statistical multiplexing from packet switching makes efficient use of link bandwidth. However, the additional link loads from multi-hopping data across a network experiencing congestion can lead to oscillation and instability of data flows. Circuit switching is an effective solution in this situation, because heavy loads can efficiently be scheduled over the available capacity. Thus, it makes great intuitive sense that different throughput levels are well served by different degrees of circuit and packet switching. In this section we address this issue, by presenting simulation results demonstrating that our stabilizing multi-hop algorithms naturally transition between circuit and packet switching in order to achieve improved delay performance over the range of achievable throughputs.

For the simulation setup of this section, we consider a WDM-based optical network having $n = 6$ nodes, with each node having a single transceiver, and no wavelength constraint. We consider a range of throughput parameters in the interval $[0, 1]$. At each throughput level, we randomly draw 25 arrival rate matrices, where entries are chosen i.i.d. uniformly from the interval $[0, 1]$, and normalize the maximum row/column sum to the desired

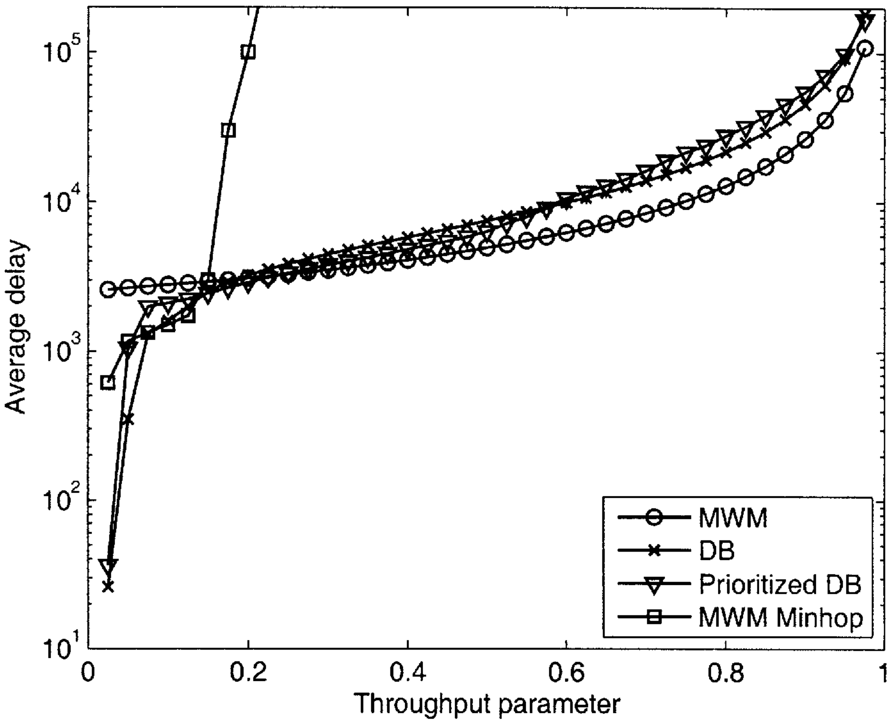


Figure 3-6: Average delay for a range of throughput levels.

throughput level. (Recall from (3.2) that when there is no wavelength constraint, and a single port per node, the throughput parameter is one minus the maximum row/column sum of the arrival rate matrix.) Each matrix is used to generate a Bernoulli arrival process that is simulated for 20×10^6 time slots, with an initial backlog of zero at each VOQ. Each point on the plots of Figures 3-6-3-8 is the mean value over the 25 sample paths generated for each arrival rate matrix.

Figure 3-6 shows the average delay for our algorithms under $\delta = 1000$. The single-hop routing algorithm (SHMW) is outperformed by all other algorithms in the low throughput regime. However, for increasing throughputs, SHMW is the algorithm with best delay performance. MW Minhop is unstable outside of the low throughput regime where the plot shows a significant jump in the delay associated with this algorithm. MHMW and Prioritized Backpressure are stable across all throughputs, though underperforming SHMW at moderate to high throughputs.

To understand the apparent performance trade-off between the circuit-centric approach (WDM reconfiguration with little or no IP layer routing) and the packet-centric approach (small amount of WDM reconfiguration with IP layer routing), we show in Figure 3-7 the average fraction of departed packets single-hopped in each time slot, and in Figure 3-8 the fraction of frames in which reconfiguration was triggered, for all algorithms. We have truncated the data in Figure 3-8 because for higher throughputs all algorithms have a fraction of approximately 1. At low throughput levels, the best performing algorithms

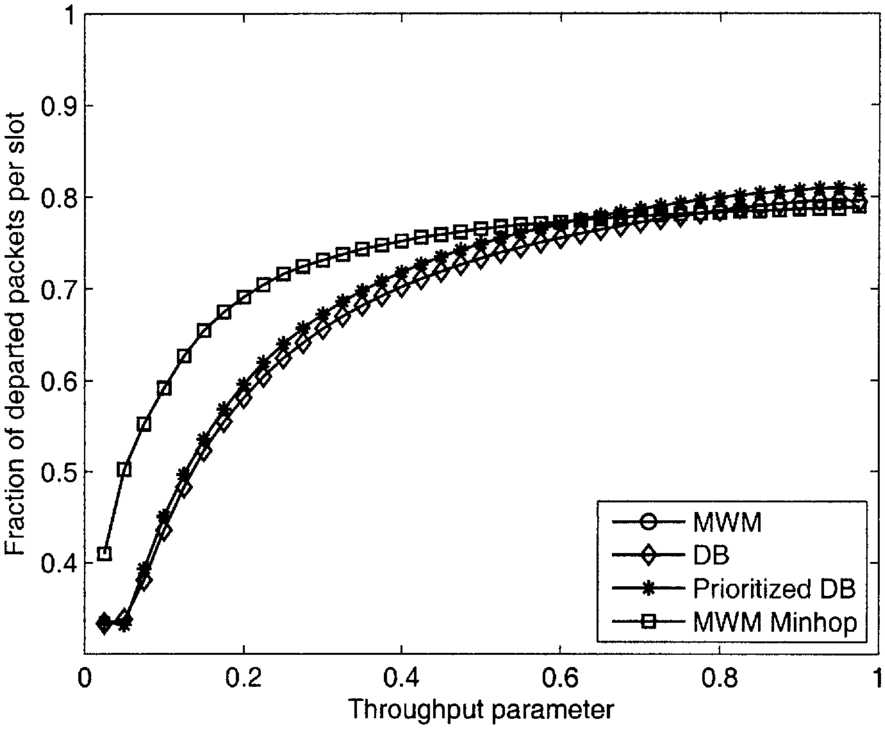


Figure 3-7: Fraction of departed packets single-hopped per time slot.

employ a large degree of IP layer routing, with a small fraction of packets single-hopped. Also, WDM layer reconfiguration is not triggered as often by the multi-hop algorithms, which implies lower delay associated with reconfiguration overhead. At high throughputs, all algorithms tend to depart more packets through single-hop routes, but the multi-hop algorithms still employ a significant amount of IP layer routing, which leads to an overall increased load and lack of performance compared to SHMW. All algorithms tend to employ WDM layer reconfiguration at each frame boundary from a relatively low throughput level and up.

We conclude that MHMW and Prioritized Backpressure are attractive algorithms, because of their ability to achieve significant gains through the use of packet routing at low throughputs and an increased tendency towards WDM reconfiguration with single-hop routing at the IP layer at high throughputs. These algorithms effectively transition between packet switching and circuit switching, and require no knowledge of the traffic arrival process other than the value of δ.

3.4.4 Frame vs. bias-based algorithms

The intuitive motivation for introducing additive bias-based algorithms is that a reconfiguration algorithm that does not make decisions at fixed intervals may be able to better adapt to actual traffic variations as they happen. Figure 3-9 provides simulation results demonstrating the validity of this argument. The simulation scenario has 6 nodes, a uni-

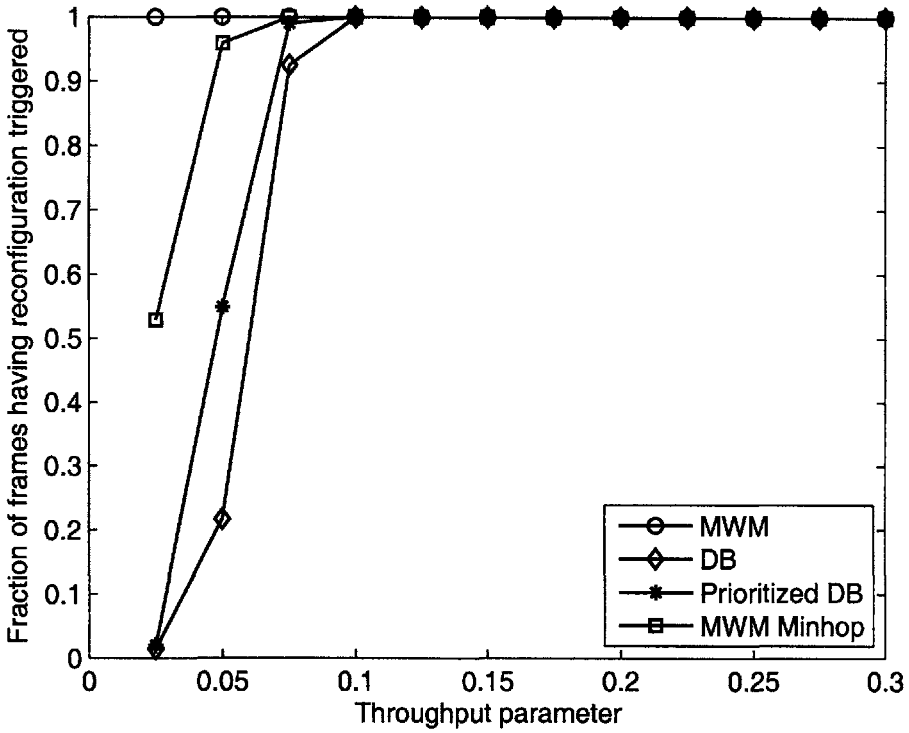


Figure 3-8: Fraction of frames in which a reconfiguration was initiated.

form arrival rate matrix of $\lambda_{i,j} = 0.04\ \forall i \neq j$ (low throughput scenario), and Bernoulli arrivals, under algorithm MHMW. Since our algorithms are intended to be implemented at a particular value of frame size I_f or bias size b, we note that for appropriately chosen bias size, there is tremendous benefit to using the bias-based algorithm in lieu of the frame-based scheme.

3.4.5 Random ring algorithms

In this section, we introduce and analyze a class of randomized algorithms from which the switch scheduling algorithms of [21] are drawn. This section considers again the scenario where there is no wavelength constraint, and a single port per node.

The class of *random ring algorithms* selects at each frame boundary a *ring* logical topology randomly with equal probability. This class of algorithms includes all possible packet routing schemes on top of the random logical topology selection.

Clearly a desirable feature of random ring algorithms is the low computational complexity associated with choosing a logical topology. Unfortunately, this results in a throughput penalty, as described in the following theorem.

Theorem 3.4.2 *For WDM networks having a single port per node, and no wavelength constraint, the class of random ring algorithms is not throughput optimal, in the sense that the stability region of any random ring algorithm has smaller volume and is a strict subset*

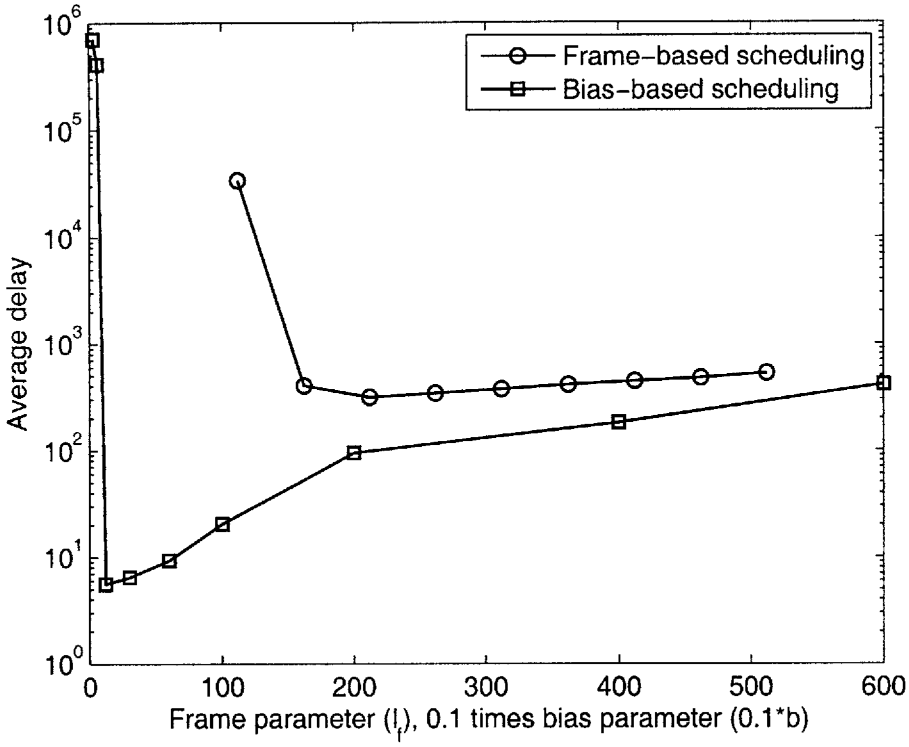


Figure 3-9: Frame/bias size versus average simulated delay.

of the doubly substochastic region.

Proof: See Appendix 3.H. ■

3.4.6 Access network

Consider an access network, where $n - 1$ of the nodes (the *local* nodes) each have a single transceiver, and one node (the *hub* node) has $P = n - 1$ ports. We assume there are n wavelengths so that the only constraints on the allowable logical topologies come from the port constraints. We consider arrival rate matrices $\boldsymbol{\lambda}$ satisfying

$$\lambda_{ij} = \begin{cases} 0, & \text{if } i = j, \\ \alpha, & \text{if } i = 1 \text{ and } j \neq i, \text{ or if } j = 1 \text{ and } i \neq j, \\ \beta, & \text{else,} \end{cases} \tag{3.3}$$

where $\alpha > 0$ and $\beta > 0$. From Theorem 3.3.2, it is easy to see that a stabilizable rate matrix for $\delta = 0$ simply must satisfy

$$\alpha + (n-2)\beta \leq 1. \tag{3.4}$$

Thus, for I_f or b chosen appropriately for their respective frame-based algorithms, we may proceed to investigate the performance trade-offs of multi-hop versus single-hop routing

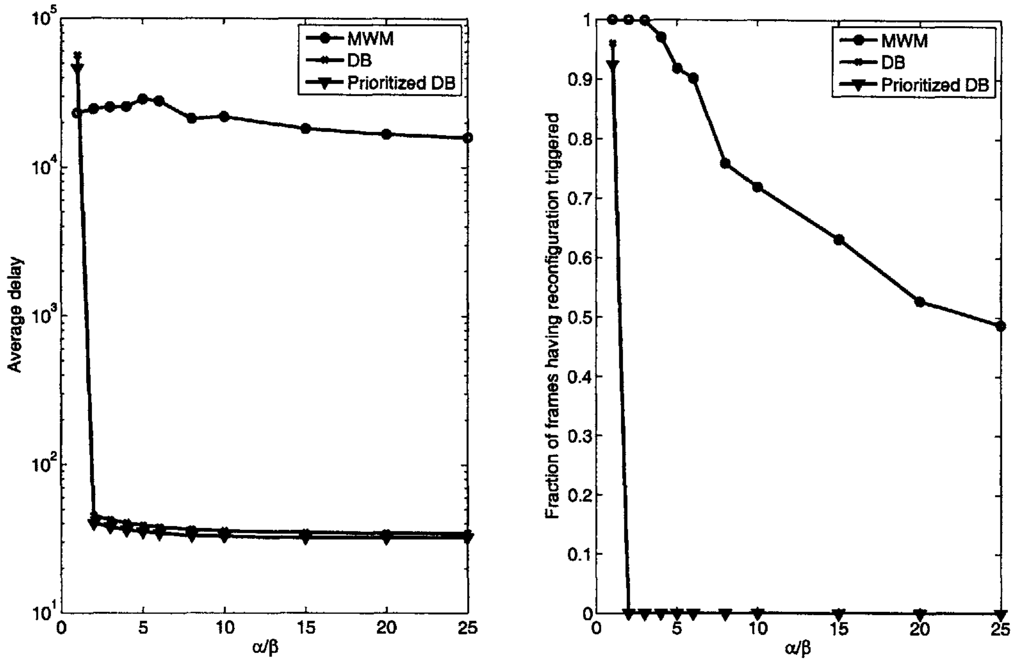


Figure 3-10: Average delay (left) and fraction of frames in which a reconfiguration was initiated (right) for a range of α/β values. $n = 6$ nodes, $\delta = 1000$ time slots. Each non-hub node has an average arrival rate of $\alpha + (n-2)\beta = 0.9$ packets per slot.

for various α, β values.

Figure 3-10 plots the data corresponding to the access network under i.i.d. Bernoulli arrivals for a range of α/β values. The plot at left of Figure 3-10 shows that the algorithms based on MHMW are far superior to SHMW for $\alpha/\beta > 1$. We plot the average fraction of frames where reconfiguration was triggered at right in Figure 3-10. It is clear that reconfiguration is in fact unnecessary in this network when the traffic is largely targeted at the hub node. Once the algorithms based on MHMW choose the logical topology directly connecting each node to the hub node, pure IP layer routing is employed thereafter. Thus, local traffic among nodes in the access network is easily served by the algorithms based on MHMW, while SHMW suffers from having to reconfigure the logical topology in order to directly service this local traffic. We have omitted the data corresponding to the MW Minhop algorithm, because of its extremely poor performance (orders of magnitude worse) next to SHMW.

3.5 Conclusions

We have studied algorithms for joint WDM reconfiguration and IP layer routing in IP-over-WDM networks. The key algorithms, SHMW and MHMW, operate based on maxweight scheduling, and are asymptotically throughput optimal in single-hop and multi-hop capable networks, respectively. We found that optical layer overhead due to reconfiguration delay is mitigated by frame-based algorithms. We provided fixed frame and variable frame duration algorithms and proved their stability properties. Our algorithms precisely dictate the control decisions made at each slot at the IP and WDM layers, with the Differential Backlog (MHMW) algorithm in general making use of both IP layer multi-hop routes and WDM reconfiguration.

In terms of delay performance, there is a great benefit from employing algorithms that tend to use multi-hop IP layer routes instead of WDM reconfiguration, when the additional load incurred from these multi-hop paths is sufficiently small. At high system loads the opposite is true, and WDM reconfiguration is preferable to additional load from multi-hop IP layer routing.

We demonstrated theoretically that multi-hop routing is of no use when reconfiguration delay is negligible, in the 3 node scenario. Further, we showed that simple algorithms employing random ring selection at the WDM layer are not capable of achieving throughput optimality.

3.5.1 Future directions

Our optical networking architecture, due to its general physical topology, and wavelength and port limitations, cannot always be considered as wavelength-unconstrained. Thus, the available configurations in the network do not correspond to matchings on a bipartite graph. This points to the challenging nature of the maxweight decision problem, which is central to SHMW and MHMW. An important future direction is to *study and develop efficient algorithms for selecting logical topology configurations under a maxweight scheduling rule.* There are many $O(n^3)$ impementations of the maximum weighted matching algorithm for bipartite graphs, including the Hungarian Method, the successive shortest path algorithm, and the relaxation algorithm [2]. Since our architecture does not necessarily admit bipartite configurations, one possible avenue is to *develop new primal-dual algorithms for maxweight scheduling.* One can treat the maxweight scheduling as a special min-cost multi-commodity integer flow problem, where link and port constraints are explicitly taken into account in the optimization. If the feasible convex set (or *polytope*) of flows under this multi-commodity integer flow problem has only integer corner points, then linear programming algorithms can be applied directly to obtain maxweight schedules. These algorithms can be combinatorialized to obtain efficient discrete routines (as in the development of the Hungarian Method [117]). It is well established however, that the general multi-commodity integer flow problem does not have exclusively integer solutions [2]. Consequently, one can determine the network conditions under which the polytope of feasible multi-commodity flows has exclusively integer corner points, and corresponding combinatorial algorithms to obtain maxweight schedules. Under network conditions where the polytope does not have integer corner points, one can develop *relaxations* on the multi-commodity integer flow problem that yield efficient approximate maxweight scheduling algorithms.

The throughput maximizing algorithms that we have considered for single-hop and multi-hop routing and reconfiguration are based on maxweight scheduling decisions. Max-weight algorithms are inherently centralized, essentially requiring that each node is made aware of all other nodes' traffic backlogs in order to make a decentralized scheduling decision.

This large degree of *communication complexity* is a highly undesirable feature, especially when considering the scalability of our network model. A very exciting and important research opportunity exists in distributed scheduling. One candidate algorithm for reducing the communication complexity in bipartite scheduling is greedy maximal weighted matching [14, 45, 71]. We study this algorithm and its performance implications in the switching and wireless settings in Chapters 6-8.

Another promising approach to achieving *optimal maxweight schedules in a distributed manner is through "belief propagation" (BP) algorithms.* Such algorithms have been tremendously successful in iterative decoding and computer vision, and have recently been demonstrated to be successful in obtaining maximum weighted matchings in simple switches [17]. BP algorithms, such as the *max-product algorithm*, seek to determine the maximum a posteriori (MAP) assignment of a probability distribution described by a *graphical model.* These algorithms are inherently *local* and thus distributed in nature. By developing an appropriate graphical model for our network architecture, BP algorithms can be used to generate optimal or approximate maxweight schedules. The challenging aspect of this problem is that convergence of BP algorithms to optimal solutions can be difficult to prove in graphs having multiple cycles, which is the case even in the simple scenario of an input-queued switch [17].

It is important to explore the *communication complexity* of our distributed algorithms. Maxweight scheduling and approximations thereof look at the scheduling problem as a throughput optimization problem, with no regard to control information dissemination. Distributed algorithms on the other hand attempt to make *local* scheduling decisions that achieve low communication complexity. Fundamentally, it is of interest to explore the trade-offs that must exist between communication complexity, delay, and throughput. One potential approach to evaluating this trade-off is to consider various implementations of primal-dual algorithms for maxweight scheduling, built with different degrees of communication complexity.

Recent results [140, 146] point to delay optimality properties of maxweight scheduling policies in the heavy traffic limiting regime. Since maxweight scheduling is at the heart of the SHMW and MHMW algorithms, these results deserve further exploration in the context of WDM networks with reconfiguration delays. Obtaining extensions of these results to WDM networks holds promise, since the results of [140, 146] essentially parallel our architecture under single-hop electronic routing. Applying similar arguments to our general multi-hop architecture should then lead to delay optimality results.

It is important to study the real-world performance implications of our joint reconfiguration and routing algorithms. For example, it is well understood that backbone network traffic is extremely aggregated, meaning traffic may be modeled as a random process with slowly-changing mean and low standard deviation. On the other hand, metropolitan area networks are subject to much more bursty traffics. These widely different traffic patterns deserve attention, because they will surely require different algorithms for achieving optimal delay performance.

Appendix

3.A Proof of Theorem 3.3.1

The proof follows closely that of Theorem 2.3.1. Note that Lemma 2.B.1 from Appendix 2.B holds true under the frame-based scheduling framework, except for the fluid model equation (2.24). We augment the lemma by adding one process, $F_\delta(t)$, which tracks the cumulative time up to and including time t spent idle during reconfiguration intervals. The linearly interpolated continuous-time version of this process can be scaled, using

$$F^r_\delta(t) = \frac{F_\delta(rt)}{r}.$$

There must then exist a subsequence of the sequence found in Lemma 2.B.1 for which the scaled functions converge uniformly on compact sets to the Lipschitz-continuous fluid limit function $\bar{F}_\delta(t)$. This function can easily be shown to satisfy the following properties:

$$\begin{aligned}\bar{F}_\delta(0) &= 0,\\ \dot{\bar{F}}_\delta(t) &= \frac{\delta}{I_f}.\end{aligned}$$

The fluid model resulting from the frame-based scheduling framework yields the following analogue of (2.24):

$$\bar{F}_\delta(t) \text{ and } \bar{F}_{\mathbf{S}}(t)\, \forall \mathbf{S} \text{ are non-decreasing,} \quad \text{and } \bar{F}_\delta(t) + \sum_{\mathbf{S}\in\mathcal{S}} \bar{F}_{\mathbf{S}}(t) = t, \quad t \geq 0.$$

Clearly also, equations (2.25)-(2.26) are unaffected by the frame-based scheduling framework. Thus, we can summarize as in (2.27):

$$\dot{\bar{F}}_\delta(t) + \sum_{\mathbf{S}\in\mathcal{S}'} \dot{\bar{F}}_{\mathbf{S}}(t) = 1.$$

Using $h(t) = (1/2)\sum_{i,j} \bar{Q}^2_{ij}(t)$, we can then apply these results to reach the inequality

$$\begin{aligned}\dot{h}(t) &\leq \sum_{\mathbf{S}\in\mathcal{S}} \alpha_{\mathbf{S}} \left(\bar{\mathbf{Z}}^*(t)\right)^T \left(\sum_j \mathbf{S}_{\cdot j}\right) - \sum_{\mathbf{S}\in\mathcal{S}'} \dot{\bar{F}}_{\mathbf{S}}(t) \left(\bar{\mathbf{Z}}^*(t)\right)^T \left(\sum_j \mathbf{S}_{\cdot j}\right)\\ &= \sum_{\mathbf{S}\in\mathcal{S}} \alpha_{\mathbf{S}} \left(\bar{\mathbf{Z}}^*(t)\right)^T \left(\sum_j \mathbf{S}_{\cdot j}\right) - \left(1 - \frac{\delta}{I_f}\right) \max_{\mathbf{S}\in\mathcal{S}} \left(\bar{\mathbf{Z}}^*(t)\right)^T \left(\sum_j \mathbf{S}_{\cdot j}\right).\end{aligned}$$

Above, we assume that the non-negative vector $\boldsymbol{\alpha}$ satisfies

$$\lambda_{ij} \le \sum_{\mathbf{S}\in\mathcal{S}} \alpha_{\mathbf{S}} d_{ij}(\mathbf{S}), \quad \forall i,j \in V, \tag{3.5}$$

$$\alpha_{\mathbf{S}} \ge 0, \quad \forall \mathbf{S} \in \mathcal{S}, \tag{3.6}$$

$$\sum_{\mathbf{S}\in\mathcal{S}} \alpha_{\mathbf{S}} = 1 - \varepsilon^*(\boldsymbol{\lambda}). \tag{3.7}$$

Denote the normalized vector $\boldsymbol{\alpha}' = (1/(1-\varepsilon^*(\boldsymbol{\lambda})))\boldsymbol{\alpha}$, and suppose that $I_f \ge \delta/\varepsilon^*(\boldsymbol{\lambda})$. Then we obtain

$$\begin{aligned}
\dot{h}(t) \le\ & (1-\varepsilon^*(\boldsymbol{\lambda}))\sum_{\mathbf{S}\in\mathcal{S}} \alpha'_{\mathbf{S}} \left(\bar{\mathbf{Z}}^*(t)\right)^T \left(\sum_j \mathbf{S}_{\cdot j}\right) - \left(1-\frac{\delta}{I_f}\right) \max_{\mathbf{S}\in\mathcal{S}} \left(\bar{\mathbf{Z}}^*(t)\right)^T \left(\sum_j \mathbf{S}_{\cdot j}\right) \\
\le\ & \left(1-\frac{\delta}{I_f}\right) \left[\sum_{\mathbf{S}\in\mathcal{S}} \alpha'_{\mathbf{S}} \left(\bar{\mathbf{Z}}^*(t)\right)^T \left(\sum_j \mathbf{S}_{\cdot j}\right) - \max_{\mathbf{S}\in\mathcal{S}} \left(\bar{\mathbf{Z}}^*(t)\right)^T \left(\sum_j \mathbf{S}_{\cdot j}\right)\right] \\
\le\ & 0.
\end{aligned}$$

The last inequality follows similarly to the final step of the proof of Theorem 2.3.1. It is immediate from the fact that $\delta \le I_f$ by definition, and the non-negative vector $\boldsymbol{\alpha}'$ sums to one.

3.B Alternative proof of stability of frame-based scheduling

The proof of Theorem 3.3.1 makes use of the powerful fluid limit technique. In this section, we demonstrate that although such a fundamental approach to proving stability is valid and correct, it may not be entirely necessary, given that algorithms SHMW and MHMW are proven to be stable for $\delta = 0$. In particular, we demonstrate that Corollaries 3.3.1 and 3.3.2 are sufficient to imply the stability of the frame-based scheduling framework.

For the purpose of simplicity, our study in this section employs a *batching* mechanism. By this, we mean that when the frame interval is I_f, and a logical link $e = (i,j)$ is active for service through the frame interval, virtual queue VOQ$_{ij}$ will not service packets across that link unless there are at least $I_f - \delta$ commodity j packets enqueued at node i.

As an example, suppose that $\delta = 1$ and $I_f = 4$. Figure 3-11 shows how exogenous arrivals for a particular VOQ are batched before being made available to that VOQ for service. All exogenous arrivals are batched and are not available for service until the frame boundary, when the maximum number of batched packets that are a multiple of $I_f - \delta = 3$ are made available to the VOQ (here, we have 3 packets made available for service at time $2I_f$ and 6 packets made available at time $3I_f$). Thus, the batch size process is nondecreasing over the frame interval, and decreases by a multiple of 3 at the frame boundaries. Because only 3 slots are allocated to servicing VOQ's within each frame, this ensures that the backlog of packets available for service at each VOQ changes by an integer multiple of 3 over every

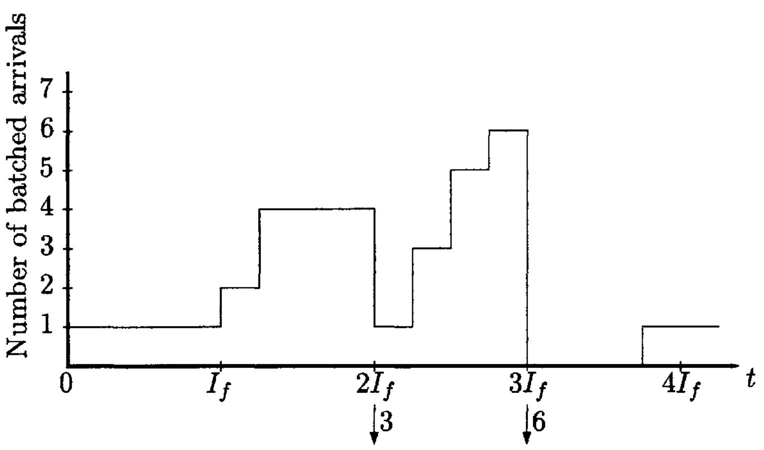


Figure 3-11: Illustration of batch size process for a particular VOQ.

frame. Thus, the frame scheme looks at the system only at the frame boundaries and considers the VOQ backlog processes divided by $I_f - \delta = 3$, and ties the resulting process back to the stabilizing scheme for $\delta = 0$.

Theorem 3.B.1 *Suppose algorithm P stabilizes the network for $\delta = 0$ for some class of arrival processes $\mathcal{A}$. Then for each $\delta > 0$, if there exists I_f such that the cumulative arrival process $(\mathbf{A}(t), t \in \mathbb{Z}_+)$ satisfies $(\tilde{\mathbf{A}}(t), t \in \mathbb{Z}_+) \in \mathcal{A}$, where*

$$\tilde{\mathbf{A}}(t) = \left\lfloor \frac{\mathbf{A}(tI_f)}{I_f - \delta} \right\rfloor,$$

then P is frame-stabilizable. *Specifically, a frame-based scheduler that makes a reconfiguration and routing decision every I_f time slots, idles for δ time slots, and subsequently services packets according to the fixed reconfiguration and routing decision made at the beginning of the frame, stabilizes the network.*

Proof: The number of batched arrivals released to the system for service at each frame boundary, kI_f for $k \in \mathbb{Z}_+$, is given by $(I_f - \delta)(\tilde{\mathbf{A}}(k) - \tilde{\mathbf{A}}(k-1))$, which is clearly an integer multiple of $(I_f - \delta)$. Thus, since the frame version of algorithm P services queues in batches of $(I_f - \delta)$ slots per frame, with the same control decision held over the duration of the frame, we are guaranteed that every virtual queue has a backlog of packets available for service that is an integer multiple of $(I_f - \delta)$.

Define the process $(\tilde{\mathbf{Q}}(t), t \in \mathbb{Z}_+)$ with $\tilde{\mathbf{Q}}(t)$ equal to $1/(I_f - \delta)$ times the backlog of packets available for service at the beginning of slot tI_f under the frame version of algorithm P. The evolution of $(\tilde{\mathbf{Q}}(t), t \in \mathbb{Z}_+)$ is defined according to the arrival process $(\tilde{\mathbf{A}}(t), t \in \mathbb{Z}_+)$ (which we assume to be a member of the set $\mathcal{A}$), and scheduling decisions according to algorithm P at each t. Thus, the process $(\tilde{\mathbf{Q}}(t), t \in \mathbb{Z}_+)$ is equivalent to the backlog process under P for $\delta = 0$ and exogenous arrival process $(\tilde{\mathbf{A}}(t), t \in \mathbb{Z}_+)$. This implies the stability of $(\tilde{\mathbf{Q}}(t), t \in \mathbb{Z}_+)$ and consequently the stability of the queue backlog process under the frame version of P. ■

Given Corollaries 3.3.1 and 3.3.2, Theorem 3.B.1 can be enlisted to infer the existence of frame-based stable scheduling policies for any $\delta > 0$. Consider the frame version of algorithm SHMW. Consider the arrival process $(\mathbf{A}(t), t \in \mathbb{Z}_+)$, having arrival rate matrix $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*_{\text{sh}}$, with $\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda}) > 0$. The class $\mathcal{A}$ of arrival processes of interest in this case is all processes that have long term arrival rates belonging to $\boldsymbol{\Lambda}^*_{\text{sh}}$. Theorem 3.B.1 then asserts the stability of frame-based scheduling if there is a frame interval I_f for which the process $(\tilde{\mathbf{A}}(t), t \in \mathbb{Z}_+) \in \mathcal{A}$, where $\tilde{\mathbf{A}}(t) = \lfloor A(tI_f)/(I_f - \delta) \rfloor$. The following sequence of equalities establishes the long-term rate of the process $(\tilde{\mathbf{A}}(t), t \in \mathbb{Z}_+)$:

$$\begin{aligned} \lim_{T\to\infty} \frac{\tilde{\mathbf{A}}(T)}{T} &= \lim_{T\to\infty} \frac{1}{T} \left\lfloor \frac{\mathbf{A}(TI_f)}{I_f - \delta} \right\rfloor \\ &= \lim_{T\to\infty} \frac{I_f}{I_f - \delta} \frac{\mathbf{A}(TI_f)}{TI_f} \\ &= \frac{I_f}{I_f - \delta} \boldsymbol{\lambda}, \quad \text{w.p.1} \end{aligned}$$

In order to satisfy $(\tilde{\mathbf{A}}(t), t \in \mathbb{Z}_+) \in \mathcal{A}$, it must then we then require that $(I_f/(I_f - \delta))\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*_{\text{sh}}$, which follows if $(I_f/(I_f - \delta)) \leq 1/(1 - \varepsilon^*_{\text{sh}}(\boldsymbol{\lambda}))$. This implies $I_f \geq \delta/\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})$. Thus, we have proved that the frame version of SHMW is stable so long as $I_f \geq \delta/\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})$. A similar proof follows for the frame version of MHMW.

3.C Proof of Theorem 3.3.2

The theorem proof is accomplished in several steps, in a similar manner as in [34, Prop. 1, Prop. 2, Alg. 1, Alg. 2]. The goal of the proof is to demonstrate that any rate $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*$ can be expressed as a convex combination of link activations from Π_N. Our approach is as follows: First, given a matrix $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*$, we find a matrix $\tilde{\boldsymbol{\lambda}}$ on the Pareto frontier of $\boldsymbol{\Lambda}^*$, which we denote by $\boldsymbol{\Lambda}^*_{\mathcal{P}}$:

$$\boldsymbol{\Lambda}^*_{\mathcal{P}} = \left\{ \boldsymbol{\lambda} : \sum_j \lambda_{ij} = P_i \forall i, \sum_i \lambda_{ij} = P_j \forall j \right\}$$

Second, an algorithm is derived for constructing a bipartite graph based on any matrix in $\boldsymbol{\Lambda}^*_{\mathcal{P}}$, with the property that the graph has a maximum matching that includes all nodes. Finally, an algorithm for expressing any matrix in $\boldsymbol{\Lambda}^*$ as a convex combination of valid link activation matrices (from the set Π_N) is provided.

3.C.1 Extending von Neumann's result

In [152], von Neumann demonstrated that any doubly substochastic matrix can be dominated by a doubly stochastic matrix. Here, we provide a methodology for finding a matrix in $\boldsymbol{\Lambda}^*_{\mathcal{P}}$ that dominates an admissible rate matrix matrix in $\boldsymbol{\Lambda}^*$.

Consider $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*$. If the summation over the elements of $\boldsymbol{\lambda}$ is less than $\sum_i P_i$, then there must exist k, l such that $\sum_j \lambda_{kj} < P_k$ and $\sum_i \lambda_{il} < P_l$. This follows easily: suppose that no such k can be found. Then $\sum_j \lambda_{kj} \geq P_k, \forall k$, which by the definition of $\boldsymbol{\Lambda}^*$ implies that $\sum_j \lambda_{kj} = P_k, \forall k$. This implies that $\sum_{kj} \lambda_{kj} = \sum_k P_k$, which violates our initial assumption.

An identical argument applies to the value of l. Thus, k, l must exist, and the entry λ_{kl} should be increased to $\lambda_{kl} + \min\{P_k - \sum_j \lambda_{kj}, P_l - \sum_i \lambda_{il}\}$. Repeating this process at most $2n-1$ times (once for each row/column with the final entry completing both a row and a column simultaneously), a matrix in $\boldsymbol{\Lambda}^*_{\mathcal{P}}$ is achieved. The following lemma summarizes this result.

Lemma 3.C.1 *Given $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*$, the above methodology yields a matrix $\tilde{\boldsymbol{\lambda}} = (\tilde{\lambda}_{ij}, i, j \in V) \in \boldsymbol{\Lambda}^*_{\mathcal{P}}$ that dominates $\boldsymbol{\lambda}$ in all entries: $\tilde{\lambda}_{ij} \geq \lambda_{ij}, \forall i, j$.*

3.C.2 Building a bipartite graph

Given matrix $\tilde{\boldsymbol{\lambda}} \in \boldsymbol{\Lambda}^*_{\mathcal{P}}$, we now construct a corresponding bipartite graph for which Hall's Theorem guarantees a maximum matching covering all nodes exists. This maximum matching can subsequently be translated to a valid link activation matrix. Designate the nodes of the two bipartitions by

$$\begin{aligned}\mathcal{P}_s &= \{s_1^1, s_1^2, \ldots, s_1^{P_1}, s_2^1, \ldots, s_2^{P_2}, \ldots, s_n^1, \ldots, s_n^{P_n}\},\\ \mathcal{P}_d &= \{d_1^1, d_1^2, \ldots, d_1^{P_1}, d_2^1, \ldots, d_2^{P_2}, \ldots, d_n^1, \ldots, d_n^{P_n}\}.\end{aligned}$$

Above, $\mathcal{P}_s$ and $\mathcal{P}_d$ represent source ports and destination ports, respectively. Algorithm 8 establishes edges between the nodes of $\mathcal{P}_s$ and $\mathcal{P}_d$.

Algorithm 8 Generates a bipartite graph from matrix $\tilde{\boldsymbol{\lambda}} \in \boldsymbol{\Lambda}^*$

```
Let $\phi = \tilde{\boldsymbol{\lambda}}$
Associate with each vertex of the bipartite graph $v$ a bin $b_v$, initially empty and having
  maximum capacity 1
for each $i, j \in V$ do
  while $\phi_{ij} > 0$ do
    Obtain $k = \min\{m : b_{s_i^m} < 1\}$, and $l = \min\{m : b_{d_j^m} < 1\}$
    Add an edge joining $s_i^k$ to $d_j^l$, if no such edge exists
    Obtain $y_{ij} = \min\{\phi_{ij}, 1 - b_{s_i^k}, 1 - b_{d_j^l}\}$
    Set $\phi_{ij} \leftarrow \phi_{ij} - y_{ij}$, $b_{s_i^k} \leftarrow b_{s_i^k} + y_{ij}$, and $b_{d_j^l} \leftarrow b_{d_j^l} + y_{ij}$
  end while
end for
```

For a matrix $\tilde{\boldsymbol{\lambda}} \in \boldsymbol{\Lambda}^*$, upon algorithm completion, it is simple to show that each bin is at capacity: Suppose $b_{s_i^k} < 1$. Then if there is no j such that $\phi_{ij} > 0$, it must be true that $\sum_j \tilde{\lambda}_{ij} \leq P_i - (1 - b_{s_i^k}) < P_i$. This follows because each time matrix entry element ϕ_{ij} is decreased, one of the bins at source i (one of $b_{s_i^1}, \ldots, b_{s_i^{P_i}}$) is increased by the same amount. Since we have assumed the entire i-th row of ϕ is zero, then the sum over the same bins must equal the initial i-th row sum of matrix ϕ, or equivalently $\sum_j \tilde{\lambda}_{ij}$. This sum must be less than P_i since all source i bins are not full, which provides a contradiction to our assumption that $\tilde{\boldsymbol{\lambda}} \in \boldsymbol{\Lambda}^*$. The argument against a value $b_{d_j^l} < 1$ upon algorithm termination follows similarly.

Alternatively, if $b_{s_i^k} < 1$ and there exists j such that $\phi_{ij} > 0$, then there must exist a value l such that $b_{d_j^l} < 1$. This follows because ϕ_{ij} has not been reduced to zero, which implies that the full column sum of P_j has not been distributed over the P_j bins corresponding to ports at destination j. Thus, the algorithm would have discovered source and destination bins with which to reduce ϕ_{ij} further, which contradicts that the algorithm has terminated.

For each $i, j \in V$, the algorithm reduces ϕ_{ij} to zero in at most $2\min\{P_i, P_j\} - 1$ steps, because this is the maximum number of times that the minimizing term y_{ij} does not have to equal ϕ_{ij}. Thus, we have shown that the algorithm terminates, and that all bins are full (at unit capacity) upon termination.

We now show that the bipartite graph constructed by the above algorithm satisfies the condition of Hall's Theorem to guarantee the existence of a perfect matching (a matching that covers every node of the graph). Take any set of source nodes $\tilde{\mathcal{P}}_s \subseteq \mathcal{P}_s$. Then we require that this set connects to at least $|\tilde{\mathcal{P}}_s|$ destination nodes in $\mathcal{P}_d$.

A useful way of considering each bin in the algorithm is as a measure of the flow departing (in the case of a source node bin) or arriving (in the case of a destination node bin) at that port. As each link is added in the algorithm, an element of matrix ϕ is reduced by some amount, and the bins associated with the source and destination nodes of that link are increased by the same amount. This captures the amount of flow serviced from the source to the destination along that link.

Upon algorithm termination, each bin is at unit capacity, which equivalently means that one unit of flow departs from each source node and arrives at each destination node. Thus, since $\tilde{\mathcal{P}}_s$ is the source of $|\tilde{\mathcal{P}}_s|$ units of flow, at least $|\tilde{\mathcal{P}}_s|$ units of flow must arrive to the destination nodes. Further, since each destination bin has unit capacity, this flow must arrive along at least $|\tilde{\mathcal{P}}_s|$ links. Thus, we have that the set of neighbor nodes to $\tilde{\mathcal{P}}_s$ must have size at least $|\tilde{\mathcal{P}}_s|$. Applying Hall's Matching Theorem [157], a perfect matching is guaranteed. The following lemma summarizes this result:

Lemma 3.C.2 *The bipartite graph generated by Algorithm 8 has a perfect matching.*

3.C.3 Translating a perfect matching on the bipartite graph into a link activation matrix

Beginning with $n \times n$ matrix $\pi = 0$, for each edge (s_i^k, d_j^l) in the perfect matching, increment π_{ij} by one. Once each edge has been considered, matrix π must have i-th row sum P_i for all i and j-th column sum P_j for all j. This follows because the matching on the bipartite graph is perfect, and thus source i is associated with P_i vertices having edges in the matching, and destination j is associated with P_j vertices having edges in the matching. Thus π corresponds to a valid logical topology under the port distribution $(P_i, i \in V)$, and given no wavelength constraint. Finally, by the construction of Algorithm 8 it is clear that a nonzero element in π implies that the corresponding entry of $\tilde{\boldsymbol{\lambda}}$ is nonzero, and conversely. The following lemma summarizes this result.

Lemma 3.C.3 *For a bipartite graph obtained according to Algorithm 8, the graph may be translated to a link activation matrix whose incidence matrix has i-th row sum equal to P_i*

and j-th column sum equal to P_j (we refer to this as a perfect link activation). Furthermore, this matrix has positive entries where $\tilde{\boldsymbol{\lambda}}$ is nonzero.

3.C.4 Proof of Theorem 3.3.2

Given $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*$, Lemma 3.C.1 guarantees the existence of a matrix $\tilde{\boldsymbol{\lambda}} \in \boldsymbol{\Lambda}^*_{calP}$ that is entry-by-entry dominant over $\boldsymbol{\lambda}$. Applying Algorithm 8 to $\tilde{\boldsymbol{\lambda}}$, Lemmas 3.C.2 and 3.C.3 guarantee the existence of a perfect link activation where each active link $i \longrightarrow j$ implies nonzero value $\tilde{\lambda}_{ij}$. Algorithm 9 (presented below) capitalizes on this to decompose $\tilde{\boldsymbol{\lambda}}$ as a convex combination of valid link activation matrices. This algorithm is the natural generalization of the decomposition presented in [34].

Algorithm 9 Decompose $\tilde{\boldsymbol{\lambda}}$ into a convex combination of link activations

```
Assign $\boldsymbol{\omega} \leftarrow \tilde{\boldsymbol{\lambda}}$
$k \leftarrow 0$
while $\boldsymbol{\omega} \neq 0$ do
    $k \leftarrow k + 1$
    For matrix $\boldsymbol{\omega}$, find a perfect link activation $\boldsymbol{\pi}^k$ according to Algorithm 8 and Lemmas 3.C.2-3.C.3
    Set $\alpha_k = \min\{\omega_{ij}/\pi^k_{ij} : \pi^k_{ij} > 0, \forall i,j \in V\}$
    Set $\boldsymbol{\omega} \leftarrow (1/(1-\alpha_k))(\boldsymbol{\omega} - \alpha_k \boldsymbol{\pi}^k)$.
end while
```

Since the link activation found for an arrival rate matrix on the Pareto frontier of $\boldsymbol{\Lambda}^*$ is perfect, step k of the algorithm reduces the i-th row sum by $\alpha_k P_i$, and the j-th column sum by $\alpha_k P_j$. Thus, all row and column sums are reduced by a factor of $1 - \alpha_k$ at each iteration. For this reason, the scale factor of $1 - \alpha_k$ is applied at each iteration to bring the matrix back to the Pareto frontier of $\boldsymbol{\Lambda}^*$. Finally, since at each iteration, α is chosen to reduce at least one matrix element to zero, with n elements reduced to zero at once in the last step, the decomposition takes at most $n^2 - n + 1$ steps to complete. $\tilde{\boldsymbol{\lambda}}$ may then be expressed as

$$\tilde{\boldsymbol{\lambda}} = \sum_{k=1}^{n^2-n+1} \left(\alpha_k \prod_{l=1}^{k-1} (1 - \alpha_l) \right) \boldsymbol{\pi}^k \tag{3.8}$$

The fact that the weights in the above decomposition sum to unity is guaranteed by the property that each link activation in the decomposition is perfect. Applying Algorithm 3, we can translate the decomposition in (3.8) to a decomposition of $\boldsymbol{\lambda}$. The only modifications to the algorithm are that initially we assign $n^* \leftarrow n^2 - n + 1$, and instead of using weights (α_k) we use weights (β_k), where $\beta_k = \alpha_k \prod_{l=1}^{k-1}(1 - \alpha_l)$ for $k = 1, \ldots, n^2 - n + 1$.

At termination, we must have

$$\boldsymbol{\lambda} = \sum_{k=1}^{n^*} \beta_k \boldsymbol{\pi}^k,$$

where $\sum_k \beta_k = 1$, and $\boldsymbol{\pi}^k$ a valid link activation for all k.

3.D Proof of Corollary 3.3.5

The port constraints provide a lower bound on $\varepsilon^*(\boldsymbol{\lambda})$:

$$\varepsilon^*(\boldsymbol{\lambda}) \geq \max\left\{\max_i \left(1 - \frac{\sum_j \lambda_{ij}}{P_i}\right), \max_j \left(1 - \frac{\sum_i \lambda_{ij}}{P_j}\right)\right\}. \tag{3.9}$$

This follows because no node i can source or terminate more than P_i packets per time slot, which means that any matrix in a decomposition of $\boldsymbol{\lambda}$ has i-th row and column sum bounded above by P_i. Let $\varepsilon_{\text{lower}}(\boldsymbol{\lambda})$ denote the term on the right in (3.9). It remains to demonstrate that $\varepsilon^*(\boldsymbol{\lambda}) \leq \varepsilon^*_{\text{lower}}(\boldsymbol{\lambda})$. We assert that the matrix $\tilde{\boldsymbol{\lambda}} = (1/(1-\varepsilon_{\text{lower}}(\boldsymbol{\lambda})))\boldsymbol{\lambda}$ must belong to $\boldsymbol{\Lambda}^*$:

$$\tilde{\boldsymbol{\lambda}} = \left(\frac{1}{\max\left\{\max_i \frac{\sum_j \lambda_{ij}}{P_i}, \max_j \frac{\sum_i \lambda_{ij}}{P_j}\right\}}\right)\boldsymbol{\lambda},$$

whose k-th row sum satisfies the inequality

$$\sum_j \tilde{\lambda}_{kj} \leq \frac{1}{\left(\frac{\sum_j \lambda_{kj}}{P_k}\right)} \sum_j \lambda_{kj} = P_k.$$

Since identical reasoning applies to any column sum, it must be true that $\tilde{\boldsymbol{\lambda}} \in \boldsymbol{\Lambda}^*$. By Theorem 3.3.2, we have that $\tilde{\boldsymbol{\lambda}} \in \text{conv}(\Pi_N)$, with decomposition weights summing to one. Consequently, $\boldsymbol{\lambda}$ can be expressed as a weighted sum of link activation matrices, with the weights summing to $(1-\varepsilon_{\text{lower}}(\boldsymbol{\lambda}))$. Thus, $\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda}) \leq \varepsilon_{\text{lower}}(\boldsymbol{\lambda})$. Since $\varepsilon^*(\boldsymbol{\lambda}) \leq \varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})$, we have that $\varepsilon^*(\boldsymbol{\lambda}) \leq \varepsilon^*_{\text{lower}}(\boldsymbol{\lambda})$, as desired.

3.E Proof of Lemma 3.3.1

The formulation that was followed in the proof of Theorem 3.3.1 also follows here. Consequently, we have that

$$\begin{aligned}
\dot{h}(t) &\leq \sum_{\mathbf{S}\in\mathcal{S}} \alpha_{\mathbf{S}} \left(\bar{\mathbf{Z}}^*(t)\right)^T \left(\sum_j \mathbf{S}_{\cdot j}\right) - \sum_{\mathbf{S}\in\mathcal{S}'} \dot{\bar{F}}_{\mathbf{S}}(t) \left(\bar{\mathbf{Z}}^*(t)\right)^T \left(\sum_j \mathbf{S}_{\cdot j}\right) \\
&= \sum_{\mathbf{S}\in\mathcal{S}} \alpha_{\mathbf{S}} \left(\bar{\mathbf{Z}}^*(t)\right)^T \left(\sum_j \mathbf{S}_{\cdot j}\right) - \left(1 - \dot{\bar{F}}_{\delta}(t)\right) \max_{\mathbf{S}\in\mathcal{S}} \left(\bar{\mathbf{Z}}^*(t)\right)^T \left(\sum_j \mathbf{S}_{\cdot j}\right) \\
&\leq (1-\varepsilon^*(\boldsymbol{\lambda})) \sum_{\mathbf{S}\in\mathcal{S}} \alpha'_{\mathbf{S}} \left(\bar{\mathbf{Z}}^*(t)\right)^T \left(\sum_j \mathbf{S}_{\cdot j}\right) - \left(1 - \dot{\bar{F}}_{\delta}(t)\right) \max_{\mathbf{S}\in\mathcal{S}} \left(\bar{\mathbf{Z}}^*(t)\right)^T \left(\sum_j \mathbf{S}_{\cdot j}\right)
\end{aligned}$$

Above, the vector $\boldsymbol{\alpha}' = (1/(1-\varepsilon^*(\boldsymbol{\lambda})))\boldsymbol{\alpha}$ satisfies (3.5)-(3.7). It follows that $\dot{h}(t) \leq 0$ when $\dot{\bar{F}}_{\delta}(t) \leq \varepsilon^*(\boldsymbol{\lambda})$. A similar proof follows for the bias-based version of algorithm SHMW.

3.F Proof of Theorem 3.3.3

We begin by considering the bias-based version of algorithm SHMW. Let $\boldsymbol{\pi}(\xi_k)$ denote the maximum weighted logical topology at time ξ_k. We will characterize the minimum time needed for another logical topology $\tilde{\boldsymbol{\pi}} \neq \boldsymbol{\pi}(\xi_k)$ to become the maximum weighted logical topology and thus trigger a WDM reconfiguration. At time ξ_k, $\boldsymbol{\pi}(\xi_k)$ satisfies

$$\tilde{\boldsymbol{\pi}}^T \mathbf{Q}(\xi_k) \leq \boldsymbol{\pi}^T(\xi_k)\mathbf{Q}(\xi_k). \tag{3.10}$$

After time ξ_k, logical topology $\boldsymbol{\pi}(\xi_k)$ will be effectively biased with b additional *dummy packets* over $\tilde{\boldsymbol{\pi}}$. Since the arrival process is Bernoulli, no more than a single packet may arrive to any VOQ at each time slot. Suppose that a single packet arrives to each of the VOQ's corresponding to logical topology $\tilde{\boldsymbol{\pi}}$ at every slot, and $\tilde{\boldsymbol{\pi}}$ does not have any lightpaths in common with $\boldsymbol{\pi}(\xi_k)$. Further suppose that there are no arrivals to VOQ's corresponding to $\boldsymbol{\pi}(\xi_k)$, and that at each slot at most one packet is removed from each of the VOQ's corresponding to $\boldsymbol{\pi}(\xi_k)$. Then, in order to have a decision to reconfigure the logical topology, the inter-reconfiguration interval χ_k must satisfy

$$\tilde{\boldsymbol{\pi}}^T \mathbf{Q}(\xi_k) + n\chi_k \geq b + \boldsymbol{\pi}^T(\xi_k)\mathbf{Q}(\xi_k) - n(\chi_k - \delta). \tag{3.11}$$

Combining (3.10) and (3.11), we obtain

$$\chi_k \geq \frac{b}{2n} + \frac{\delta}{2}. \tag{3.12}$$

Suppose $b/n \geq 2(\delta/\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})) - \delta$. Then, using (3.12), we have that $\chi_k > \delta/\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})$ for all k, which means that irrespective of the backlog process, at least $\delta/\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})$ slots pass before a reconfiguration decision. Thus, for $\kappa > 0$

$$\begin{aligned} F_\delta(r(t+\kappa)) - F_\delta(rt) &\leq \delta \left\lceil \frac{r\kappa}{\delta/\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})} \right\rceil, && (3.13) \\ &\leq r\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})\kappa + \delta. && (3.14) \end{aligned}$$

Dividing both sides of (3.14) by r, the right hand side of the inequality can be made arbitrarily close to $\varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})\kappa$ for sufficiently large integer r. This immediately implies that $\dot{\bar{F}}_\delta(t) \leq \varepsilon^*_{\text{sh}}(\boldsymbol{\lambda})$, which is sufficient for stability.

The proof for the bias-based version of MHMW follows similarly, except that rather than tracking the possible change in VOQ backlogs associated with each logical configuration, we must track the change in maximum backpressure. Again, since the arrival process is Bernoulli, no more than a single packet may arrive to any VOQ at each time slot. Suppose that $\tilde{\pi}_e = 1$, with commodity $v \in V$ maximizing backpressure across logical link e. Then the differential backlog across e increases maximally at each slot when:

1. one exogenous commodity v packet arrives at node $\sigma(e)$ in each slot;

2. one commodity v packet arrives internally at node $\sigma(e)$ in each slot;

3. one commodity v packet is serviced away from node $\tau(e)$ (and not back to node $\sigma(e)$) at each slot; and

4. there is no arrival of commodity v packet at node $\tau(e)$ in each slot.

Further, suppose that $\pi_{e'}(\xi_k) = 1$, with commodity $v' \in V$ maximizing backpressure across logical link e'. Then the backpressure across e' decreases maximally when:

1. commodity v' uniquely maximizes backpressure across e', since maximum backpressure across e' then only depends on service of commodity v' packets;

2. node $v' \neq \tau(e')$, since backpressure decreases by two units at each service in this case;

3. node $\sigma(e')$ has no exogenous arrivals of commodity v' packets

4. node $\tau(e')$ receives an exogenous arrival of one commodity v' packet

Thus, the weight associated with $\tilde{\pi}$ increases by at most $3n$ at each slot, and that associated with $\pi(\xi_k)$ decreases by at most $3n$ at each slot. Then, in order to have a decision to reconfigure the logical topology, the inter-reconfiguration interval χ_k must satisfy

$$\tilde{\pi}^T \mathbf{Z}^*(\xi_k) + 3n\chi_k \geq b + \pi^T(\xi_k)\mathbf{Z}^*(\xi_k) - 3n(\chi_k - \delta). \tag{3.15}$$

Note that at time ξ_k, $\pi(\xi_k)$ satisfies $\tilde{\pi}^T \mathbf{Z}^*(\xi_k) \leq \pi^T(\xi_k)\mathbf{Z}^*(\xi_k)$. Combining this fact with (3.15), we obtain

$$\chi_k \geq \frac{b}{6n} + \frac{\delta}{2}. \tag{3.16}$$

Thus, if we select $b/n \geq 6(\delta/\varepsilon^*(\boldsymbol{\lambda})) - 3\delta$, and follow the same steps as above for algorithm SHMW, we can conclude that $\dot{\bar{F}}_\delta(t) \leq \varepsilon^*(\boldsymbol{\lambda})$, which is sufficient for stability of the bias-based version of algorithm MHMW.

3.G Proof of Theorem 3.4.1

The proof is by induction, using a stochastic coupling argument [147]. We begin with policy $P_0 =$ MHMW, and successively refine it to a policy employing single-hop routing, with no worse average expected aggregate backlog. The recursion implies that a policy with no multi-hopping produces smaller or equal average aggregate backlog. For this proof, at step $k-1$ of the induction, assume that arrivals under policies P_{k-1} and P_k are coupled to the same queues for all time. Quantities marked with a tilde symbol, such as $\tilde{\mathbf{Q}}$, correspond to policy P_k, while those without a tilde symbol correspond to policy P_{k-1}.

For convenience, we denote by $a_{ij}(t)$ the number of exogenous arrivals to VOQ$_{ij}$ at time $t \in \mathbb{Z}_+$, and $u_{ij}(t)$ as the cumulative service of packets (departed and internally arriving) at VOQ$_{ij}$ at time $t \in \mathbb{Z}_+$. We collect these variables into vectors $\mathbf{a}(t), \mathbf{u}(t)$, respectively.

Consider policy P_{k-1} for $k \in \mathbb{Z}_+$ and time slot $k-1$. By the recursion, up to and including time $k-1$ policy P_{k-1} does not multi-hop any packets. At time k, if P_{k-1} does not multi-hop any packets, then let P_k choose the same controls as P_{k-1} for all subsequent time slots. If P_{k-1} does multi-hop one or more packets, let P_k choose the same controls as P_{k-1} up to time $k-1$. At time k, we must consider three cases. For all time after k, let P_k attempt to mimic P_{k-1} in its controls, only deviating from P_{k-1} if there simply is no packet in a queue under P_k where for the corresponding queue under P_{k-1} a packet is multi-hopped or departs the system.

Case 1. If P_{k-1} multi-hops only a single packet, along link (a,b), then note that for any link (a,b) there are only two possible logical topologies containing this link. These configurations are $\{(a,b),(b,c),(c,a)\}$ and $\{(a,b),(b,a)\}$. For either configuration, link (a,b) is being used to multi-hop a packet from VOQ$_{ac}$ to VOQ$_{bc}$. Let $\mathbf{Q}(k-1) = (Q_{ab}, Q_{bc}, Q_{ca}, Q_{ac}, Q_{cb}, Q_{ba})$ be the queue backlogs at time $k-1$. For the first configuration containing link (a,b), policy P_{k-1} results in the following queue occupancy at time k,

$$\mathbf{Q}(k) = \mathbf{Q}(k-1) + \mathbf{a}(k) + (0, -u_{bc}(k)+1, -u_{ca}(k), -1, 0, 0)\,.$$

Since $-u_{bc}(k)+1 \geq 0$, it is sufficient to let P_k employ a logical configuration that allows packets to depart from the VOQ$_{ca}$ and VOQ$_{ac}$. This is clearly an allowable control, and thus P_k results in the queue occupancy distribution

$$\tilde{\mathbf{Q}}(k) = \mathbf{Q}(k-1) + \mathbf{a}(k) + (0, 0, -u_{ca}(n), -1, 0, 0)\,.$$

For the second possible configuration containing link (a,b), the queue occupancy distributions at time k are

$$\mathbf{Q}(k) = \mathbf{Q}(k-1) + \mathbf{a}(k) + (0, 1, 0, -1, 0, -u_{ba}(k))\,,$$
$$\tilde{\mathbf{Q}}(k) = \mathbf{Q}(k-1) + \mathbf{a}(k) + (0, 0, 0, -1, 0, -u_{ba}(k))\,.$$

Here, P_k chooses the configuration that allows packets from the VOQ$_{ac}$ and VOQ$_{ba}$ to exit the system.

For either case, it is clear that P_k has an improved or equal aggregate queue occupancy at each time after k.

Case 2. If P_{k-1} multi-hops two packets, there are three possible sets of links that are used for multi-hopping: $\{(a,b),(b,c)\}$, $\{(a,b),(c,a)\}$, or $\{(a,b),(b,a)\}$. Note that each of these sets of links forces the network to a particular configuration, because of the assumption of a single port per node. We consider each of these cases in turn. If P_{k-1} multi-hops packets along links (a,b) and (b,c), then P_{k-1} has enabled logical configuration $\{(a,b),(b,c),(c,a)\}$ for single-hop service. The queue occupancy distributions under the policies are then given by

$$\mathbf{Q}(k) = \mathbf{Q}(k-1) + \mathbf{a}(k) + (0, 1, -u_{ca}(k)+1, -1, 0, -1)\,,$$
$$\tilde{\mathbf{Q}}(k) = \mathbf{Q}(k-1) + \mathbf{a}(k) + (0, 0, 0, -1, 0, -1)\,.$$

Here, policy P_k chooses the switch configuration that allows packets from VOQ$_{a,c}$ and VOQ$_{b,a}$ to exit the system.

If P_{k-1} multi-hops packets along links (a,b) and (c,a), then P_{k-1} has again chosen logical configuration $\{(a,b),(b,c),(c,a)\}$. The queue occupancy distributions under the policies are then given by

$$\mathbf{Q}(k) = \mathbf{Q}(k-1) + \mathbf{a}(k) + (1, -u_{bc}(k) + 1, 0, -1, -1, 0)\,,$$
$$\tilde{\mathbf{Q}}(k) = \mathbf{Q}(k-1) + \mathbf{a}(k) + (0, 0, 0, -1, -1, 0)\,.$$

Here, policy P_k chooses the logical configuration that allows packets from VOQ_{ac} and VOQ_{cb} to exit the system.

Finally, if P_{k-1} multi-hops packets along links (a, b) and (b, a), then P_{k-1} has chosen logical configuration $\{(a, b), (b, a)\}$. The queue occupancy distributions under the policies are then given by

$$\mathbf{Q}(k) = \mathbf{Q}(k-1) + \mathbf{a}(k) + (0, -1+1, 0, -1+1, 0, 0)\,,$$
$$\tilde{\mathbf{Q}}(k) = \mathbf{Q}(k-1) + \mathbf{a}(k) + (0, 0, 0, 0, 0, 0)\,.$$

Here, policy P_k does nothing because P_{k-1} has effectively made no change to its occupancy distribution.

It is clear that in all cases, P_k has an improved or equal aggregate queue occupancy at each time after $k-1$.

Case 3. If P_{k-1} multi-hops three packets then the logical configuration must be $\{(a, b), (b, c), (c, a)\}$. The queue occupancy distributions under the policies are then given by

$$\mathbf{Q}(k) = \mathbf{Q}(k-1) + \mathbf{a}(k) + (1, 1, 1, -1, -1, -1)\,,$$
$$\tilde{\mathbf{Q}}(k) = \mathbf{Q}(k-1) + \mathbf{a}(k) + (0, 0, 0, -1, -1, -1)\,.$$

Here, policy P_k chooses the switch configuration $\{(a, c), (c, b), (b, a)\}$ to allow packets from $\text{VOQ}_{a,c}$, $\text{VOQ}_{c,b}$, and $\text{VOQ}_{b,a}$ to exit the system. Again, it is clear that P_k results in an improved aggregate queue occupancy at each time after $k-1$.

This completes the induction.

3.H Proof of Theorem 3.4.2

For this proof, we invoke the multi-hop parameters described in Section 3.2. The proof follows for any $\delta \geq 0$. Denote by $\Pi^r \subset \Pi(V)$ the set of logical topology matrices corresponding to logical rings of size n. Recall from Definition 2.2.1 that an arrival rate matrix is stabilizable if there exists a *subprobability measure* $(\phi_{\mathbf{S}}, \mathbf{S} \in \mathcal{S})$ such that

$$\sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}} \leq 1, \tag{3.17}$$

$$\sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}} d_{ij}(\mathbf{S}) \geq \lambda_{ij}, \quad i, j \in V. \tag{3.18}$$

Since there are $(n-1)!$ different logical rings having n nodes, it is clear that under any random ring algorithm, the long-term amount of time allocated to each ring is $1/(n-1)!$. Thus, the subprobability measures $(\phi_{\mathbf{S}}, \mathbf{S} \in \mathcal{S})$ achievable under a random ring algorithm are restricted to the form

$$\phi_{\mathbf{S}} = \sum_{\pi \in \Pi^r} \frac{\phi_{\mathbf{S}|\pi}}{(n-1)!}, \quad \mathbf{S} \in \mathcal{S}$$

where $\sum_{\mathbf{S}} \phi_{\mathbf{S}|\pi} = 1$ for all $\pi \in \Pi^r$, and $\phi_{\mathbf{S}|\pi} > 0$ only if $\mathbf{S}$ is an allowed activation matrix under logical ring π.

For $i, j \in V$, we may now express the left hand side of (3.18) as

$$\sum_{\mathbf{S} \in \mathcal{S}} \sum_{\pi \in \Pi^r} \frac{\phi_{\mathbf{S}|\pi}}{(n-1)!} d_{ij}(\mathbf{S}) = \frac{1}{(n-1)!} \sum_{\pi \in \Pi^r} \sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}|\pi} d_{ij}(\mathbf{S}). \tag{3.19}$$

Now $(\phi_{\mathbf{S}|\pi}, \mathbf{S} \in \mathcal{S})$ has no restrictions other than to be a subprobability measure restricted to logical ring π. Consider the set of arrival rate matrices that are dominated by the *inner summation* in (3.19), as we range over the compact set of feasible subprobability measures $(\phi_{\mathbf{S}|\pi}, \mathbf{S} \in \mathcal{S})$. This set of arrival rate matrices must be equal to the stability region corresponding to electronic routing over a *fixed* logical ring. Thus, the set of stabilizable arrival rate matrices for the class of random ring algorithms has outer bound equal to the average over the $(n-1)!$ fixed-ring stability regions. Since each fixed-ring stability region clearly has smaller volume than the doubly substochastic region, the result follows.

Chapter 4

Achieving 100% throughput in reconfigurable optical networks: The single-wavelength case

In this chapter, we continue our study of the optical networking architecture introduced in Chapter 3. Chapter 3 focused on developing scheduling algorithms for addressing delays associated with reconfiguration in networks with no wavelength constraints. In this Chapter, we quantify the impact of wavelength constraints on the network throughput properties. We determine the performance penalty associated with wavelength constraints, and we characterize the performance gap between architectures that employ single-hop versus multi-hop routing at the electronic layer.

4.1 Overview and summary of contributions

A major contribution of this chapter is a characterization of the capacity region for single-wavelength optical networks through a linkage to the Routing and Wavelength Assignment (RWA) problem for WDM networks. This characterization allows us to derive fundamental geometric properties of the capacity region for optical networks of arbitrary topologies. In this chapter, we primarily focus on single-wavelength optical networks. The single wavelength topology is commonly used in traditional metropolitan and access networks operating on one frequency (*e.g.* 1.3nm systems). Moreover, our single-wavelength treatment simplifies the presentation considerably and can be extended, by appropriate scaling of the capacity region, to multi-wavelength optical networks.

Our work is conceptually related to Birkhoff-von Neumann (BvN) decompositions, particularly as applied to switching theory [34,152]. The set of switch configurations (or *service configurations*) available to an $n \times n$ input-queued switch is typically represented by the set of permutation matrices of size n. The result of [150] implies that the convex hull of these service configurations equals the capacity region of the input-queued switch. BvN decompositions draw on these concepts to express any stabilizable rate matrix as a convex combination of permutation matrices (service configurations) [34]. An alternative characterization employs a result of Birkhoff [22] to state that the convex hull of the service matrices (permutation matrices) equals the doubly substochastic region [97]. Like BvN decompositions for input-queued switches, our work seeks to express any stabilizable rate matrix as a convex combination of service configurations. Unlike input-queued switches, our optical networking architecture has physical constraints, such as port and wavelength limitations, that affect the set of service configurations. For example, the set of service configurations may not include the full set of permutation matrices, and may include non-permutation matrices. Thus, while the work of [150] allows us to express the capacity region as the convex hull of available service configurations, this description can have limited value in providing an understanding of the geometric properties of the capacity region. This is in contrast to the case of the input-queued switch, where a result of Birkhoff [22] has been applied to demonstrate that the convex hull of the service matrices (permutation matrices) equals the doubly substochastic region [97]. Recently, the study of [92] has developed order bounds, based on uniform multi-commodity flow, for maximum achievable throughput performance in general network settings. In this chapter, we develop a theory of *RWA decompositions* that enables us to *exactly* elicit geometric properties of the capacity region of single-wavelength optical networks having general topologies.

4.1.1 Simple motivating example

Consider a unidirectional ring network having 3 nodes, as depicted in Figure 4-1(a). Suppose this network is restricted to a *single wavelength per optical fiber*, with lightpaths routed *only in the clockwise direction.* These constraints restrict the network to four *maximal* logical topologies[1]. These topologies are illustrated in Figure 4-1.

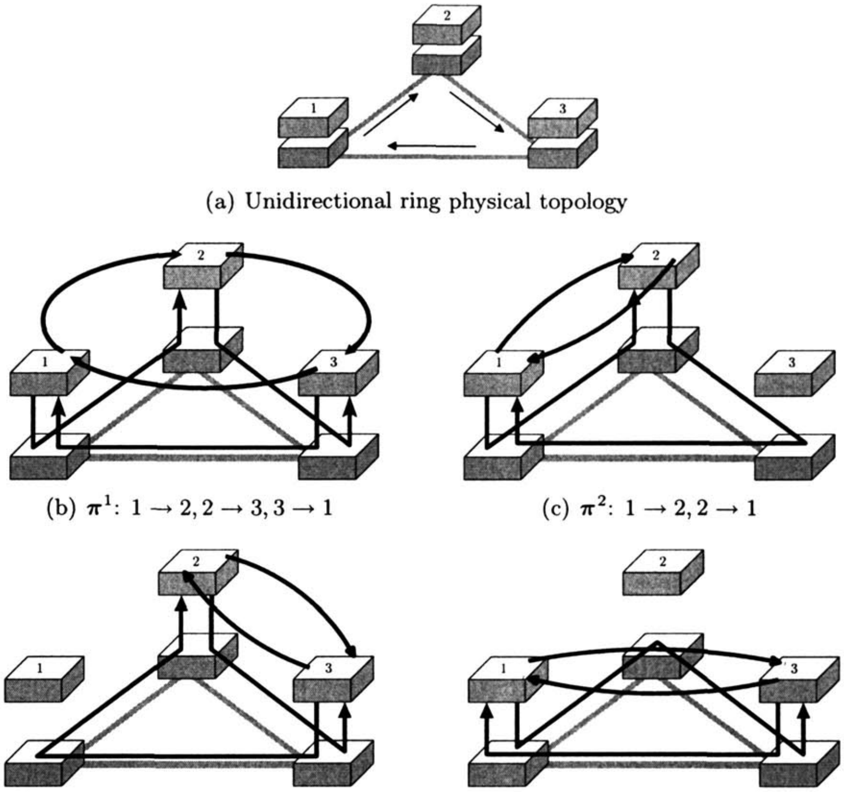
Consider the traffic matrix $\boldsymbol{\lambda}$, given by

$$\boldsymbol{\lambda} = \begin{bmatrix} \cdot & 0 & \theta \\ \theta & \cdot & 0 \\ 0 & \theta & \cdot \end{bmatrix}, \tag{4.1}$$

where the (i, j)-th entry of $\boldsymbol{\lambda}$ is equal to the average arrival rate of packets to node i destined for node j. We wish to determine the maximum value of θ that the network can support, given that only one packet can be serviced along a logical link per time slot. If we restrict the network to only use single-hop electronic-layer routes, the maximum value of θ is $1/3$. This follows because logical links $1 \to 3$, $2 \to 1$, and $3 \to 2$ each traverse two fibers, which due to the single-wavelength constraint means that only one of these links can be served at a time. Sharing time equally between the three links affords a maximum of $1/3$ of the proportion of time to service each link. Thus, $\theta = 1/3$ is the maximum value such that the traffic rate matrix $\boldsymbol{\lambda}$ can be supported.

Suppose instead that we allow the network to make use of multi-hop electronic-layer routes. In this case, a simple policy that maintains logical topology π^1 (Figure 4-1(b)) for all time and multi-hops packets along the electronic layer leads to a link load of 2θ on each

[1] Every valid logical topology is either equal to, or has some subset of *logical links* from, one of the maximal topologies.



(a) Unidirectional ring physical topology

(b) π^1: $1 \to 2, 2 \to 3, 3 \to 1$

(c) π^2: $1 \to 2, 2 \to 1$

(d) π^3: $2 \to 3, 3 \to 2$

(e) π^4: $3 \to 1, 1 \to 3$

Figure 4-1: There are four maximal logical topology configurations for the unidirectional three-node ring having a single wavelength per optical fiber. The logical configurations are depicted as lightpath routings (straight-edge links with corners) with corresponding logical topology graph overlaid (curved links).

logical link. Since no more than 1 unit of traffic per time slot can be supported on each wavelength, this policy can support any $\theta \leq 1/2$. This is a clear improvement over the achievable traffic rate matrix supported under single-hop routing. The value $\theta = 1/2$ is also the maximum value achievable, which is easily seen by noting that each physical link has 2θ units of traffic demand that it must service.

For comparison, consider the wavelength-unconstrained case [26, 27, 128], which in the case of the 3-node unidirectional ring topology implies that there exist at least three wavelengths per optical fiber. The maximum value of θ that is supported in this case is $\theta = 1$, which is achievable by maintaining for all time the logical configuration $1 \to 3, 2 \to 1, 3 \to 2$.

This example highlights three important points. First, the wavelength constraint has been shown to reduce the maximum throughput achievable under single-hop and multi-hop routing. This is an example of the intuitively obvious fact that wavelength constraints often lead to throughput penalties. Second, there is a throughput performance gap between electronic layers employing multi-hop versus exclusively single-hop routing. Again, this is intuitively obvious in light of the optical-layer constraints, but this is in contrast to the case of unconstrained networks, where single-hop and multi-hop algorithms are identical in terms of throughput performance [26]. Finally, note that both the single-hop and multi-hop cases have made use of service configurations that cannot be equated to permutation matrices, where each input port is always connected to a single output port, each output port is always connected to a single input port, and the connections are exclusively used for single-hop service of packets. This points to the fact that a direct application of BvN decompositions does not apply in constrained network scenarios. These observations suggest three important goals of this chapter:

1. to develop a theory of generalized decompositions analogous to BvN decompositions for port and wavelength constrained networks;
2. to explore the throughput penalty of constrained versus unconstrained optical networks; and
3. to determine the throughput gap between single-hop and multi-hop electronic-layer routing algorithms.

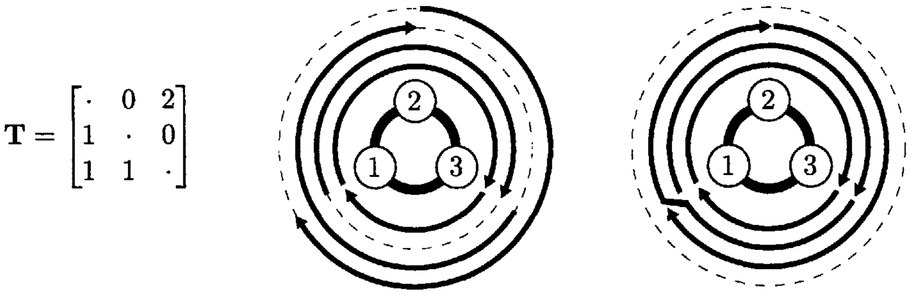
4.2 RWA decompositions

In this section, we demonstrate that in any optical network having a single wavelength per physical fiber link, the question of stability for a particular arrival rate matrix can be directly tied to the RWA problem on the same physical topology graph. Note that our work considers capacity properties of single-wavelength optical networks. Yet, we use properties of the RWA for multi-wavelength optical networks to characterize the capacity region of single-wavelength optical networks. We directly relate the RWA problem with no wavelength conversion to the set of achievable rates using only single-hop electronic routing, and the RWA problem with wavelength conversion to the set of achievable rates using multi-hop electronic layer routes.

4.2.1 The RWA problem

The objective of the RWA problem is to minimize the number of wavelengths needed to set up a certain set of lightpaths for a given physical topology. We consider two versions of the RWA problem: RWA with no wavelength conversion capability and RWA with full wavelength conversion capability.

Let $\mathbf{T} = (T_{ij})$ be a non-negative $n \times n$ integer lightpath demand matrix, where T_{ij} is the number of lightpaths, originating at node i and terminating at node j, that must be assigned. In the case of no wavelength conversion capability, the RWA is subject to the *wavelength continuity constraint*, which requires that no lightpath makes use of more than a single color from its source to its destination. In this case, let $W^{\mathrm{nc}}(\mathrm{T})$ be the minimum number of wavelengths required to service the demands of matrix $\mathbf{T}$ with no wavelength conversion (see Appendix 4.A for details). As an example, consider the 3-node *unidirectional* ring physical topology having a single wavelength per optical fiber, and the lightpath demand matrix $\mathbf{T}$ given in Figure 4-2(a). A valid RWA with no wavelength conversion is provided in Figure 4-2(b), It is easy to see for this network that $W^{\mathrm{nc}}(\mathbf{T}) = 4$.


$$\mathbf{T} = \begin{bmatrix} \cdot & 0 & 2 \\ 1 & \cdot & 0 \\ 1 & 1 & \cdot \end{bmatrix}$$

(a) Demand matrix (b) Without (c) With

Figure 4-2: RWAs with and without wavelength conversion for traffic **T**. The physical topology is a unidirectional ring (clockwise oriented). A dashed line indicates an idle wavelength on the corresponding fiber links.

A network node having full wavelength conversion capability can transform any pass-through lightpath, in the optical domain, from its incident wavelength to any other wavelength. In this case, we define $W^{\mathrm{c}}(\mathbf{T})$ to be the minimum number of wavelengths required to service the demands of **T** with wavelength conversion (see Appendix 4.A for details). Since using a single color per lightpath is accommodated by the RWA with wavelength conversion, it is clear for any physical topology that $W^{\mathrm{c}}(\mathbf{T}) \leq W^{\mathrm{nc}}(\mathbf{T})$ for all **T**. For the trivial case of $\mathbf{T} = 0$, we define (for technical reasons) that $W^{\mathrm{nc}}(0) = W^{\mathrm{c}}(0) = 1$. For the traffic demand **T** of Figure 4-2(a), Figure 4-2(c) depicts the RWA employing wavelength conversion. In this case, $W^{\mathrm{c}}(\mathbf{T}) < W^{\mathrm{nc}}(\mathbf{T})$ (the inequality is strict).

4.2.2 Examples of RWA decompositions

In the RWA problem, multiple *single-wavelength* logical configurations are *multiplexed* through the use of frequency division (WDM). In our reconfigurable network setting, restricted to a single wavelength per optical fiber, multiple single-wavelength logical configurations are multiplexed through the use of time division (by enabling logical reconfiguration and adjustable electronic-layer routing over time). Through careful interchange of time and frequency, we can conceptually link the RWA problem to the stability issue in our reconfigurable network. Consequently, we will demonstrate how to transform a RWA for a particular wavelength traffic demand into a sequence of arrival rate matrices belonging to the network capacity region, when the network N has a single wavelength per optical fiber. We next demonstrate this relationship with examples for both the single-hop and multi-hop scenarios.

Single-hop RWA decompositions

Consider the RWA with no wavelength conversion for traffic **T** in Figure 4-2(a). The RWA of Figure 4-2(b) multiplexes the traffic demand **T** over 4 wavelengths. This RWA can be expressed as a *decomposition* of **T** into a superposition of single-wavelength logical topology configurations (expressed in matrix form) as follows,

$$\mathbf{T} = \begin{bmatrix} \cdot & 0 & 1 \\ 0 & \cdot & 0 \\ 1 & 0 & \cdot \end{bmatrix} + \begin{bmatrix} \cdot & 0 & 1 \\ 0 & \cdot & 0 \\ 0 & 0 & \cdot \end{bmatrix} + \begin{bmatrix} \cdot & 0 & 0 \\ 0 & \cdot & 0 \\ 0 & 1 & \cdot \end{bmatrix} + \begin{bmatrix} \cdot & 0 & 0 \\ 1 & \cdot & 0 \\ 0 & 0 & \cdot \end{bmatrix}, \tag{4.2}$$

where the matrices from left to right represent the rings depicted in Figure 4-2(a) in order of increasing radius. Note that each of the matrices in the decomposition of (4.2) is a valid single-wavelength logical configuration.

Assuming there is a constant number $W \geq 4$ wavelengths available in each optical fiber, then we can say that each single-wavelength logical configuration in the RWA utilizes a fraction of $1/W$ of the total available *multiplexing resources* in the network. The *utilization* of the multiplexing resources is then given by $4/W \leq 1$.

We also consider *time as a multiplexing resource*; however, since we consider the evolution of our system over an infinite horizon, the time resource is normalized to unity. Consequently, when a particular single-wavelength logical configuration utilizes a fraction of the time resource, this is a measure of the long-term fraction of time spent servicing that logical configuration.

Consider equation (4.2), the valid RWA for traffic matrix $\mathbf{T}$ on the physical topology G_P, and re-interpret each wavelength configuration as utilizing $1/W$ of the available time resources in a single-wavelength network N. We have established that each wavelength configuration from the RWA is a valid single-wavelength logical topology and that the total utilization of multiplexing resources can be no more than 1. Consequently, we have validly multiplexed time in the single-wavelength network N. The resulting rate matrix corresponding to time sharing of service configurations is given for $W \geq 4$ by

$$\boldsymbol{\lambda}_W = \frac{1}{W}\mathbf{T} = \frac{1}{W} \begin{bmatrix} \cdot & 0 & 2 \\ 1 & \cdot & 0 \\ 1 & 1 & \cdot \end{bmatrix}.$$

Using (4.2), we have an explicit decomposition of $\boldsymbol{\lambda}_W$ into a convex combination of valid single-hop service matrices, subject to a single-wavelength per optical fiber,

$$\boldsymbol{\lambda}_W = \frac{1}{W} \begin{bmatrix} \cdot & 0 & 1 \\ 0 & \cdot & 0 \\ 1 & 0 & \cdot \end{bmatrix} + \frac{1}{W} \begin{bmatrix} \cdot & 0 & 1 \\ 0 & \cdot & 0 \\ 0 & 0 & \cdot \end{bmatrix} + \frac{1}{W} \begin{bmatrix} \cdot & 0 & 0 \\ 0 & \cdot & 0 \\ 0 & 1 & \cdot \end{bmatrix} + \frac{1}{W} \begin{bmatrix} \cdot & 0 & 0 \\ 1 & \cdot & 0 \\ 0 & 0 & \cdot \end{bmatrix} + \frac{W-4}{W} \begin{bmatrix} \cdot & 0 & 0 \\ 0 & \cdot & 0 \\ 0 & 0 & \cdot \end{bmatrix}. \tag{4.3}$$

From the decomposition of (4.3), we can immediately conclude that $\boldsymbol{\lambda}_W \in \boldsymbol{\Lambda}^*_{\text{sh}}$ for $W \geq 4$ (this follows directly from the definition of $\boldsymbol{\Lambda}^*_{\text{sh}}$). In words, the arrival rate matrix $\boldsymbol{\lambda}_W$ belongs to the single-hop capacity region for any $W \geq 4$. We call this decomposition a *single-hop RWA decomposition* of $\boldsymbol{\lambda}_W$. In summary, by interchanging frequency and time, we have used a RWA for a particular wavelength traffic demand to produce a sequence of arrival rate matrices belonging to the single-hop capacity region of N, when N has a single wavelength per optical fiber.

Multi-hop RWA decompositions

For the RWA with wavelength conversion, each wavelength routing can be considered a valid single-wavelength logical configuration. The difference from the RWA with no wavelength conversion is that lightpaths on a particular wavelength can have endpoints on that wavelength, corresponding to the use of a wavelength converter. We can re-interpret the RWA problem in our reconfigurable setting by noting that while the RWA problem uses wavelength converters to take advantage of available resources at *different frequencies* (equivalently, wavelengths), our reconfigurable network uses electronic-layer queues to take advantage of available resources at *different times.* Thus, wherever RWA invokes a wavelength converter, the reconfigurable network can be understood to terminate a lightpath at that node and electronically enqueue the carried data for multi-hop transmission to its destination at a different time.

We demonstrate multi-hop RWA decompositions in the following example. Consider the RWA problem for the wavelength traffic demand $\mathbf{T}$ in Figure 4-2(a). We have established that wavelength conversion can be used to service $\mathbf{T}$ with only 3 wavelengths, as depicted in Figure 4-2(c). This RWA can be expressed as the following decomposition, with the matrices successively representing the rings depicted in Figure 4-2(b) in order of increasing radius,

$$\mathbf{T} = \begin{bmatrix} \cdot & 0 & 1 \\ 0 & \cdot & 0 \\ 1 & 0 & \cdot \end{bmatrix} + \begin{bmatrix} \cdot & -1 & 1 \\ 0 & \cdot & 0 \\ 0 & 1 & \cdot \end{bmatrix} + \begin{bmatrix} \cdot & 1 & 0 \\ 1 & \cdot & 0 \\ 0 & 0 & \cdot \end{bmatrix}. \tag{4.4}$$

The above decomposition can be interpreted as follows. The first wavelength fully services demands $\{1 \longrightarrow 3, 3 \longrightarrow 1\}$. The second wavelength services demand $1 \longrightarrow 3$ and services demand $3 \longrightarrow 2$ *only up to node* 1. Consequently, the '-1' in the second matrix of (4.4) represents the $3 \longrightarrow 2$ traffic that is enqueued for multi-hop transmission at node 1. The third wavelength services the remainder of demand $3 \longrightarrow 2$ from node 1 to node 2 as well as fully servicing demand $2 \longrightarrow 1$.

Thus, for $W \geq 3$, the arrival rate matrix $\boldsymbol{\lambda}_W = (1/W)\mathbf{T}$ can be expressed using (4.4) as a convex combination of valid single-wavelength multi-hop service matrices,

$$\boldsymbol{\lambda}_W = \frac{1}{W}\begin{bmatrix} \cdot & 0 & 1 \\ 0 & \cdot & 0 \\ 1 & 0 & \cdot \end{bmatrix} + \frac{1}{W}\begin{bmatrix} \cdot & -1 & 1 \\ 0 & \cdot & 0 \\ 0 & 1 & \cdot \end{bmatrix} + \frac{1}{W}\begin{bmatrix} \cdot & 1 & 0 \\ 1 & \cdot & 0 \\ 0 & 0 & \cdot \end{bmatrix} + \frac{W-3}{W}\begin{bmatrix} \cdot & 0 & 0 \\ 0 & \cdot & 0 \\ 0 & 0 & \cdot \end{bmatrix}.$$

We conclude that $\boldsymbol{\lambda}_W \in \boldsymbol{\Lambda}^*_{\mathrm{mh}}$ for $W \geq 3$.

4.3 Capacity regions from RWA decompositions in single-wavelength networks

The examples of the previous section have shown how the RWA with and without wavelength conversion *for a single traffic demand* $\mathbf{T}$ can be translated to a sequence of arrival rate matrices belonging to the single-hop and multi-hop capacity regions, respectively. In this section, we will demonstrate that the single-hop and multi-hop capacity regions for single-wavelength optical networks can be fully described by the RWA functions W^{nc} and W^{c}, respectively.

4.3.1 Single-hop capacity region

We begin by considering the single-hop capacity region. In networks with no wavelength constraints, this region is characterized in Chapter 3 and [26,27]. In [154,155], this region is studied as the capacity region of general optical flow switched networks. Our characterization, which is exclusive to single-wavelength networks, is useful in our subsequent development of geometric properties of the capacity region. In particular, it allows us to express the entire capacity region as a collection of limit points based on the solution to the RWA problem.

The example of Section 4.2.2 provided a sequence of arrival rates belonging to $\mathbf{\Lambda}^*_{\mathrm{sh}}$ for a single integer traffic demand matrix $\mathbf{T}$ in the RWA problem with no conversion. In this section we consider all such arrival rates, gathered over all possible integer traffics $\mathbf{T}$ in the RWA problem. Let $\mathcal{R}^{\mathrm{nc}}$ be the set of all such arrival rates,

$$\mathcal{R}^{\mathrm{nc}} = \left\{ \boldsymbol{\lambda} = \frac{1}{W}\mathbf{T} : \mathbf{T} \in \mathbb{Z}_+^{n \times n}, W \in \mathbb{Z}_+, W \geq W^{\mathrm{nc}}(\mathrm{T}) \right\}. \tag{4.5}$$

Recall that we are restricting attention to joint optical reconfiguration and electronic layer routing algorithms where the optical layer has only a single wavelength available in each optical fiber. Consequently, $\mathbf{\Lambda}^*_{\mathrm{sh}}$ is the single-hop capacity region of the single-wavelength network N.

For the set $\mathcal{R}$, let $\mathrm{cl}(\mathcal{R})$ represent the closure[2] of $\mathcal{R}$. We next establish that every matrix in $\mathrm{cl}(\mathcal{R}^{\mathrm{nc}})$ belongs to $\mathbf{\Lambda}^*_{\mathrm{sh}}$, and conversely, that every matrix in $\mathbf{\Lambda}^*_{\mathrm{sh}}$ belongs to $\mathrm{cl}(\mathcal{R}^{\mathrm{nc}})$.

Theorem 4.3.1 $\mathbf{\Lambda}^*_{\mathrm{sh}} = \mathrm{cl}(\mathcal{R}^{\mathrm{nc}})$

Proof: See Appendix 4.B. ■

4.3.2 Multi-hop capacity region

The multi-hop capacity region is characterized in a similar manner. In [26,27], this region is characterized for networks having no wavelength constraints, where it is shown that the single-hop and multi-hop capacity regions are equal. In [154,155], a queueing model is enlisted to study the throughput properties of optical packet switched networks (OPS). The OPS capacity region of [154,155] is related to the multi-hop capacity region of our reconfigurable optical network, with differences arising depending on the set of available optical layer network configurations Π_N. Our characterization in the single-wavelength setting is tailored to our subsequent analysis of geometric properties of the multi-hop capacity region.

[2] An accumulation point of $\mathcal{R}$ is such that there exist other points of $\mathcal{R}$ arbitrarily close by. The closure of $\mathcal{R}$ is then given by the union of $\mathcal{R}$ and all its accumulation points [93].

Here, we gather all possible arrival rates generated by multi-hop RWA decompositions over all possible traffic demand matrices T into the set $\mathcal{R}^{\mathrm{c}}$,

$$\mathcal{R}^{\mathrm{c}} = \left\{ \boldsymbol{\lambda} = \frac{1}{W}\mathbf{T} : \mathbf{T} \in \mathbb{Z}_{+}^{n \times n}, W \in \mathbb{Z}_{+}, W \geq W^{\mathrm{c}}(\mathbf{T}) \right\}. \tag{4.6}$$

Through similar steps as in the single-hop case, we can establish the following theorem.

Theorem 4.3.2 $\boldsymbol{\Lambda}^{*}_{\mathrm{mh}} = \mathrm{cl}(\mathcal{R}^{\mathrm{c}})$

Proof: The proof is similar to the proof of Theorem 4.3.1, and is provided, with important details specific to the multi-hop scenario, in Appendix 4.C. ■

4.4 Geometric properties of the capacity region

While the capacity properties of our dynamically reconfigurable electronic-over-optical network are well characterized in the multi-hop and single-hop cases through equations (2.7) and (2.9), respectively, these expressions do not easily yield simple geometric properties of the capacity regions. This is in contrast to the characterization of the input-queued switch capacity region of equation (3.1).

The remainder of this work is dedicated to extracting geometric properties of the single-hop and multi-hop capacity regions in the wavelength-constrained WDM network setting.

In what follows, we will occasionally refer to the wavelength-unconstrained network setting. From our assumption that node $v \in V$ has P_v transceivers available, when the network has no wavelength constraint, the capacity region (single-hop and multi-hop) equals [26,27]

$$\boldsymbol{\Lambda}_{\mathrm{port}} = \left\{ \boldsymbol{\lambda} : \textstyle\sum_j \lambda_{ij} \leq P_i \,\forall i,\, \sum_i \lambda_{ij} \leq P_j \,\forall j \right\}. \tag{4.7}$$

4.4.1 Maximum uniform arrival rate matrices

In this section, we make use of RWA decompositions to establish geometric properties of the single-hop and multi-hop capacity regions. Define $\mathbf{J}$ as the $n \times n$ matrix having (i,j) entry equal to 1 if $i \neq j$:

$$\mathbf{J} = \begin{bmatrix} \cdot & 1 & \cdots & 1 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \cdots & 1 & \cdot \end{bmatrix}.$$

We then seek to determine the maximum values $\theta^{\mathrm{sh}}, \theta^{\mathrm{mh}}$ such that $\theta^{\mathrm{sh}}\mathbf{J}$ belongs to the single-hop capacity region, and $\theta^{\mathrm{mh}}\mathbf{J}$ belongs to the multi-hop capacity region.

Theorem 4.4.1 *For network N having a single wavelength per optical fiber, let $\theta^{\mathrm{sh}} = \sup\{\theta : \theta\mathbf{J} \in \boldsymbol{\Lambda}^{*}_{\mathrm{sh}}\}$. Then,*

$$\theta^{\mathrm{sh}} = \limsup_{k \to \infty} \frac{k}{W^{\mathrm{nc}}(k\mathbf{J})}. \tag{4.8}$$

Table 4.1: Maximum values $\theta^{\text{sh}}, \theta^{\text{mh}}$ for various physical topologies having a single wavelength per optical fiber. The corresponding wavelength-unconstrained values are listed under $\theta^{\max}$, along with the resulting throughput performance gap.

Phy. topology G_P	$P_i, \forall i$	$W^{\text{nc}}(l\mathbf{J}) = W^{\text{c}}(l\mathbf{J})$	$\theta^{\text{sh}} = \theta^{\text{mh}}$	$\theta^{\max}$	Performance gap $\frac{\theta^{\text{mh}}}{\theta^{\max}}$
Tree $\mathcal{T}$	1	$l \max_{e \in \mathcal{T}} \lvert\mathcal{N}_{e,1}\rvert\lvert\mathcal{N}_{e,2}\rvert$	$1/(\max_{e \in \mathcal{T}} \lvert\mathcal{N}_{e,1}\rvert\lvert\mathcal{N}_{e,2}\rvert)$	$1/(n-1)$	$\frac{n-1}{\max_{e \in \mathcal{T}} \lvert\mathcal{N}_{e,1}\rvert\lvert\mathcal{N}_{e,2}\rvert}$
Unidir. ring	1	$ln(n-1)/2$	$2/(n^2-n)$	$1/(n-1)$	$2/n = O(1/n)$
Bidir. ring					
n odd	2	$l(n^2-1)/8$	$8/(n^2-1)$	$2/(n-1)$	$4/(n+1) = O(1/n)$
n even	2	$\lceil ln^2/8 \rceil$	$8/n^2$	$2/(n-1)$	$4(n-1)/n^2 = O(1/n)$
2D Torus R rows, C cols (R, C div. by 4)	4	$\lceil lRC(R+C)/16 \rceil$	$16/(RC(R+C))$	$4/(n-1)$	$\frac{4(n-1)}{n(R+C)} = O(1/(R+C))$
Bin. hypercube	$\log_2 n$	$ln/2$	$2/n$	$(\log_2 n)/(n-1)$	$\frac{2(n-1)}{n \log_2 n} = O(1/\log_2 n)$

For the multi-hop scenario, let $\theta^{\text{mh}} = \sup\{\theta : \theta\mathbf{J} \in \mathbf{\Lambda}^*_{\text{mh}}\}$. *Then,*

$$\theta^{\text{mh}} = \limsup_{k \to \infty} \frac{k}{W^{\text{c}}(k\mathbf{J})}. \tag{4.9}$$

Proof: See Appendix 4.D. ■

Equations (4.8) and (4.9) essentially capture the maximum ratio of the uniform traffic load l to the number of wavelengths needed to support that traffic demand. These values are a measure of the most efficient way that the uniform traffic demand l can be packed over network N, with or without wavelength conversion.

Theorem 4.4.1 allows us to draw on the literature regarding RWA algorithms for various physical topologies to obtain geometric properties of the single-hop and multi-hop capacity regions. As an example, consider the unidirectional ring having a single transceiver per node ($P_i = 1$). In this case, it can be shown that the minimum numbers of wavelengths required to service traffic $l\mathbf{J}$ with or without wavelength conversion are equal: $W^{\text{nc}}(l\mathbf{J}) = W^{\text{c}}(l\mathbf{J}) = ln(n-1)/2$. Applying (4.8) and (4.9) we obtain a maximum uniform arrival rate of $\theta^{\text{sh}} = \theta^{\text{mh}} = 2/(n^2-n)$. Thus, there is no single-hop versus multi-hop performance gap for uniform arrival rates under the unidirectional ring. However, noting in the wavelength-unconstrained case (see (4.7)), the maximum uniform arrival rate is given by $\theta^{\max} = 1/(n-1)$, we find a constrained versus unconstrained performance gap of $2/n = O(1/n)$.[3]

We draw the RWA values $W^{\text{nc}}(l\mathbf{J}), W^{\text{c}}(l\mathbf{J})$ from [41,131–133], and summarize the single-hop and multi-hop maximum uniform arrival rates for several physical topologies in Table 4.1. The table lists the maximum uniform arrival rates achievable in the single-wavelength setting, as well as the corresponding maximum uniform arrival rate achievable in the wavelength-unconstrained case, $\theta^{\max}$, and the implied unconstrained versus constrained

[3] We employ O-notation to represent an *asymptotically tight bound* [45] on the performance gap.

performance gap. For the tree topology $\mathcal{T}$, denote $\mathcal{N}_{e,1}, \mathcal{N}_{e,2}$ as the node sets in the cut corresponding to edge $e \in \mathcal{T}$.

A remarkable property evident from Table 4.1 is that for all physical topologies considered, there is no single-hop versus multi-hop performance gap with respect to uniform arrival rates. This follows for all physical topologies considered in Table 4.1, because under uniform traffic demand, RWA with and without wavelength conversion can achieve the same minimum number of wavelengths. It is conjectured in [131] that this result holds generally over all physical topologies.

Note that *the geometric properties listed in the table are exact.* For physical topologies besides rings, trees, tori, hypercubes, and others where the solution to the RWA problem is known, the exact characterizations of (4.8) and (4.9) can be approximated through evaluation of the RWA functions over multiple all-to-all integer traffic demands. Techniques for solving the integer RWA problem are well-studied in the literature. In [43], various RWA methodologies are classified, based on their optimization criteria, and their approach to solving the problem. Additional comments regarding the solution to the RWA problem can be found in Appendix 4.A. The computability of θ^{sh} and θ^{mh} is explored further in Chapter 5.

4.4.2 Maximum scaled doubly substochastic set

In this section, we take advantage of RWA decompositions to derive bounds on the maximum scaling that can be applied to the set of doubly substochastic matrices, such that every matrix in the scaled set is contained within the capacity region. For a mathematical description of this property we require the following definitions.

Definition 4.4.1 *For matrix* $\mathbf{A}$, *let the maximum row/column sum of* $\mathbf{A}$ *be given by* $\|\mathbf{A}\|_{\max}$:

$$\|\mathbf{A}\|_{\max} = \max\left\{\max_i \textstyle\sum_j A_{ij}, \max_j \sum_i A_{ij}\right\}.$$

Definition 4.4.2 *Let the set* $\mathcal{D}_s$ *denote the doubly substochastic region, scaled by factor* s,

$$\mathcal{D}_s = \left\{\boldsymbol{\lambda} \in \mathbb{R}_+^{n \times n} : \|\boldsymbol{\lambda}\|_{\max} \leq s\right\}.$$

We seek the maximum values $\alpha^{\mathrm{sh}}, \alpha^{\mathrm{mh}}$ such that the sets $\mathcal{D}_{\alpha^{\mathrm{sh}}}, \mathcal{D}_{\alpha^{\mathrm{mh}}}$ are respectively subsets of the single-hop and multi-hop capacity regions. We will demonstrate that there are cases in which the multi-hop capacity region provides improved performance over the single-hop capacity region, in terms of this geometric property. Consequently, we can conclude that there are indeed cases in which multi-hop routing can provide a strict throughput performance improvement over algorithms that exclusively employ single-hop routes. This is in contrast to the case of a crossbar switch, where single-hop algorithms can achieve the capacity region.

Definition 4.4.3 *The integer matrix* $\mathbf{T} = (T_{ij}) \in \mathbb{Z}_+^{n \times n}$ *is called* k*-allowable if it satisfies* $\|\mathbf{T}\|_{\max} \leq k$. *Let* $\mathcal{K}_k$ *be the set of all* k*-allowable matrices.*

Let $\mathcal{W}^{\text{nc}}(k)$ be the minimum number of wavelengths required to service *any* k-allowable traffic matrix in the RWA with no conversion: $\mathcal{W}^{\text{nc}}(k) = \max_{\mathbf{T}\in\mathcal{K}_k} \mathcal{W}^{\text{nc}}(\mathbf{T})$. Similarly, let the corresponding value with wavelength conversion be $\mathcal{W}^{\text{c}}(k)$. The RWA problem for k-allowable matrices was introduced in [60] and subsequently studied in [41, 131–133]. These papers seek to understand the values of the quantities $\mathcal{W}^{\text{nc}}(k), \mathcal{W}^{\text{c}}(k)$ for various physical topologies. The bidirectional ring with no wavelength conversion is considered in [60, 133], tree topologies with no wavelength conversion were considered in [60, 132], and ring and torus topologies with wavelength conversion were considered in [41]. Additional results for k-allowable traffics can be found in [131].

The following theorem establishes the quantity α^{sh} as the maximum scale factor on the substochastic region, such that the scaled region is a subset of the single-hop capacity region. The analogous result for the multi-hop case is also provided.

Theorem 4.4.2 *Let* $\alpha^{\text{sh}} = \sup\{\alpha : \mathcal{D}_\alpha \subseteq \mathbf{\Lambda}^*_{\text{sh}}\}$. *Then,*

$$\alpha^{\text{sh}} = \limsup_{k\to\infty} \frac{k}{\mathcal{W}^{\text{nc}}(k)}. \tag{4.10}$$

Similarly, let $\alpha^{\text{mh}} = \sup\{\alpha : \mathcal{D}_\alpha \subseteq \mathbf{\Lambda}^*_{\text{mh}}\}$. *Then,*

$$\alpha^{\text{mh}} = \limsup_{k\to\infty} \frac{k}{\mathcal{W}^{\text{c}}(k)}. \tag{4.11}$$

Proof: See Appendix 4.E. ∎

Equations (4.10) and (4.11) provide the limiting ratios of k to the worst-case number of wavelengths required to support any k-allowable traffic, in their respective RWA problems. This is a measure of the most efficient way that the worst-case k-allowable traffic can be packed over network N, in the limit of large k.

Applying Theorem 4.4.2, we can use results from the RWA literature [40,41,108,131–133] to characterize the values $\alpha^{\text{sh}}, \alpha^{\text{mh}}$ for various physical topology configurations. Consider for example the bidirectional ring having an even number $n \geq 8$ nodes. For the RWA with no wavelength conversion, the worst-case k-allowable traffic requires $\lceil kn/3 \rceil$ wavelengths, resulting in a maximum scaling of $\alpha^{\text{sh}} = 3/n$. The RWA with wavelength conversion requires at most $\lceil kn/4 \rceil$ wavelengths for any k-allowable traffic, yielding $\alpha^{\text{mh}} = 4/n$. Consequently, we have a single-hop versus multi-hop performance gap of 3/4, irrespective of the number of nodes in the network. Designating the maximum scale value achievable in the wavelength-unconstrained case by α^{max}, we note that the bidirectional ring has $\alpha^{\text{max}} = 2$, since the architecture employs two transceivers per node (one for each incident fiber). This yields a constrained versus unconstrained performance gap in the unidirectional ring of $2/n$. Our results for various physical topologies are summarized in Table 4.2. Note that the value of $\mathcal{W}^{\text{c}}(k)$ for a bidirectional ring when n is odd remains an open problem. Consequently, Table 4.2 provides the tightest known interval in which this value resides [40], and the interval in which α^{mh} resides. The lower limit of this interval is derived based on the next theorem (see Theorem 4.4.3 and the subsequent discussion). Also note that for the tree

Table 4.2: Maximum values $\alpha^{\rm sh}, \alpha^{\rm mh}$ for various physical topologies having a single wavelength per optical fiber. Also listed for each topology is the single-hop versus multi-hop performance gap, as well as the constrained versus unconstrained performance gap.

Phy. topology G_P	$\mathcal{W}^{\rm nc}(k)$	$\alpha^{\rm sh}$	$\mathcal{W}^{\rm c}(k)$	$\alpha^{\rm mh}$	$\alpha^{\rm sh}/\alpha^{\rm mh}$	$\alpha^{\rm mh}/\alpha^{\rm max}$
Star	k	1	k	1	1	1
Tree $\mathcal{T}$	$kc_{\mathcal{T}}$	$1/c_{\mathcal{T}}$	$kc_{\mathcal{T}}$	$1/c_{\mathcal{T}}$	1	$1/c_{\mathcal{T}}$
Unidir. ring	kn	$1/n$	$k(n-1)$	$1/(n-1)$	$1-1/n$	$1/(n-1)$
Bidir. ring ($n \geq 7$)						
n odd	$\lceil kn/3 \rceil$	$3/n$	$\left\lceil \frac{k(n-1)}{4} \right\rceil \leq \mathcal{W}^{\rm c}(k) \leq \lceil \frac{kn}{4} \rceil$	$\frac{4n}{n^2-1} \leq \alpha^{\rm mh} \leq \frac{4}{n-1}$	$\leq \frac{3}{4} - \frac{3}{4n^2}$	$\leq \frac{2}{n-1}$
n even	$\lceil kn/3 \rceil$	$3/n$	$\lceil kn/4 \rceil$	$4/n$	$3/4$	$2/n$

network, throughput performance depends on the tree topology employed, and particularly on the worst-case cut that maximizes the number of nodes on the smaller side of the cut. We call this number $c_{\mathcal{T}}$. Recalling our definition of $\mathcal{N}_{e,1}, \mathcal{N}_{e,2}$ as the node sets in the cut corresponding to edge e, we have $c_{\mathcal{T}} \triangleq \max_{e \in \mathcal{T}} \min\{|\mathcal{N}_{e,1}|, |\mathcal{N}_{e,2}|\}$.

Theorem 4.4.2 provides an *exact* characterization of the maximum scaled doubly substochastic region fully contained within $\Lambda^*_{\rm mh}$. If an order bound is sufficient, then we can use [92, Lem. 1] to provide the following connection between the geometric properties studied in this section.

Theorem 4.4.3 $n\theta^{\rm mh}/2 \leq \alpha^{\rm mh} \leq (n-1)\theta^{\rm mh}$

Proof: Lemma 1 of [92] can be understood in our reconfigurable WDM network setting as follows: if $\theta \mathbf{J} \in \Lambda^*_{\rm mh}$, then $\mathcal{D}_\alpha \subseteq \Lambda^*_{\rm mh}$ when $\alpha \leq n\theta/2$. The lower bound follows. The upper bound follows since $(\alpha^{\rm mh}/(n-1))\mathbf{J} \in \mathcal{D}_{\alpha^{\rm mh}} \subseteq \Lambda^*_{\rm mh}$, which implies $\theta^{\rm mh} \geq \alpha^{\rm mh}/(n-1)$. ■

Theorem 4.4.3 allows us to obtain a refined bound on $\alpha^{\rm mh}$ for the bidirectional ring when n is odd. For this physical topology, Theorem 4.4.3 provides that $\alpha^{\rm mh} \geq 4n/(n^2-1)$. Based only on the fact (from Table 4.2) that $\lceil k(n-1)/4 \rceil \leq \mathcal{W}^{\rm c}(k) \leq \lceil kn/4 \rceil$, we find that $4/n \leq \alpha^{\rm mh} \leq 4/(n-1)$. However, since $4n/(n^2-1) > 4/n$ for $n \geq 2$, we can obtain the refined bound, $4n/(n^2-1) \leq \alpha^{\rm mh} \leq 4/(n-1)$.

A similar statement to Theorem 4.4.3 cannot be made for the quantity $\alpha^{\rm sh}$, because the argument of [92, Lem. 1] is inherently a multi-hop result.

4.5 Conclusions

In this chapter, we have studied the optimal throughput performance properties of reconfigurable WDM-based packet networks. We considered networks having arbitrary physical topologies, and general node architectures.

In general, the capacity region of joint arrival rates that can be supported in a particular network is described as a convex combination of available service configurations (joint routing and WDM configurations) in that network. However, this typical characterization provides little insight into the physical attributes of the capacity region, particularly of important performance metrics.

We thus undertook a study of geometric properties of the capacity region in networks having general topologies. The work of this chapter focused on networks having a single-wavelength per optical fiber. We developed a theory of RWA decompositions that establishes the entire capacity region under any physical topology in terms of the RWA properties of the same physical topology graph. The RWA problem with no conversion was tied to the single-hop capacity region of the reconfigurable network, while the RWA problem with conversion was tied to the multi-hop capacity region.

This characterization enabled us to *exactly* determine certain geometric properties of the capacity region under any physical topology, restricted to a single-wavelength per optical fiber: the maximum all-to-all arrival rate and maximum doubly substochastic region that can be supported by the network. We presented closed-form solutions for certain network topologies such as rings, trees, and tori. For any other physical topology, the characterization of these geometric properties in terms of the RWA problem can be approximated through numerical evaluation of the RWA problem.

These geometric properties provide a measure of the optimal achievable throughput under any physical topology. Consequently, a network designer could use such a metric in comparing and evaluating network topologies and/or varying node functionality. For example, we have *exactly* demonstrated the throughput performance gap between wavelength-limited and wavelength-unconstrained networks having particular physical topologies. Additionally, we have exactly characterized the throughput performance gap between networks employing exclusively single-hop routing and those employing multi-hop routing. In the case of the bidirectional ring, we have observed a performance improvement of 33% of multi-hop over single-hop enabled networks.

The contributions of this chapter are primarily theoretical in nature, but we have laid out the essential considerations for network designers seeking to understand the performance limits of future configurable optical networks. Naturally, the single-wavelength constraint we adopted is not realistic in many practical settings. However, the development of multi-wavelength capacity properties is quite similar to the single-wavelength case. We consider multi-wavelength networks in the next chapter.

Appendix

4.A The RWA optimization

The RWA problem with full wavelength conversion is an integer multicommodity flow problem, which can be formulated as follows [106]. Let $\mathbf{T} = (T_{ij}) \in \mathbb{Z}_+^{n \times n}$ represent the set of lightpath demands, and let f_{ij}^e be a flow variable that represents the number of lightpaths from node i to node j that cross the fiber link e. For physical topology graph G_P, let E_v^σ be the set of edges originating at node v: $E_v^\sigma = \{e \in E_P : \sigma(e) = v\}$. Similarly, let E_v^τ denote the set of edges terminating at node v: $E_v^\tau = \{e \in E_P : \tau(e) = v\}$.

$$\begin{aligned} \min \quad & W && (4.12)\\ \text{s.t.} \quad & W \geq \textstyle\sum_{i,j \in V} f_{ij}^e, \quad \forall e \in E_P && (4.13)\\ & \sum_{e \in E_v^\sigma} f_{ij}^e - \sum_{e \in E_v^\tau} f_{ij}^e = \begin{cases} T_{ij} & v = i \\ -T_{ij} & v = j \\ 0 & \text{else} \end{cases} \quad \forall v, i, j \in V && (4.14)\\ & f_{ij}^e \in \mathbb{Z}_+, \quad \forall i, j \in V,\, e \in E_P && (4.15) \end{aligned}$$

The minimum value W reached by the optimization is $W^{\mathrm{c}}(\mathbf{T})$.

The RWA problem with no wavelength conversion can be formulated through the addition of the following constraints in the optimization (4.12)-(4.15), which impose the *wavelength-continuity constraint* on the RWA problem.

$$\begin{aligned} & f_{ij}^e = \textstyle\sum_{w=1}^{W} c_{ij}^{e,w} \quad \forall i, j \in V, e \in E_P \\ & \sum_{e \in E_v^\sigma} c_{ij}^{e,w} - \sum_{e \in E_v^\tau} c_{ij}^{e,w} \begin{cases} \geq 0 & v = i \\ \leq 0 & v = j \\ = 0 & \text{else} \end{cases} \quad \forall v, i, j \in V \\ & c_{ij}^{e,w} \in \{0, 1\} \quad \forall i, j \in V, e \in E_P, w \in \{1, \ldots, W\} \end{aligned}$$

The minimum value W reached by this optimization is $W^{\mathrm{nc}}(\mathbf{T})$.

Commonly, the RWA problem is solved in two stages, first by solving the lightpath routing problem, followed by obtaining a wavelength assignment for the routing determined in the first step [43]. The routing problem can be solved sequentially using shortest-path algorithms, or through standard integer programming solution methods such as randomized rounding. The wavelength assignment algorithm is typically studied as a graph coloring algorithm, with common approaches to the problem including sequential assignment, genetic algorithms, simulated annealing, and randomized rounding. See [43] and the references contained therein for details.

4.B Proof of Theorem 4.3.1

Here we divide the proof as follows. First we demonstrate that $\text{cl}(\mathcal{R}^{\text{nc}}) \subseteq \mathbf{\Lambda}^*_{\text{sh}}$, and second we prove $\mathbf{\Lambda}^*_{\text{sh}} \subseteq \text{cl}(\mathcal{R}^{\text{nc}})$.

Proof that $\text{cl}(\mathcal{R}^{\text{nc}}) \subseteq \mathbf{\Lambda}^*_{\text{sh}}$*:* Suppose $\boldsymbol{\lambda} \in \mathcal{R}^{\text{nc}}$. Then from (4.5) there must exist $\mathbf{T}, W$ such that $\boldsymbol{\lambda} = (1/W)\mathbf{T}$, with $\mathbf{T} \in \mathbb{Z}^m_+$ and $W \geq W^{\text{nc}}(\mathbf{T})$. We establish a RWA decomposition for $\boldsymbol{\lambda}$ as a subconvex combination of $W^{\text{nc}}(\mathbf{T})$ matrices as follows. For each index $i = 1, \ldots, W^{\text{nc}}(\mathbf{T})$, we construct the matrix $\tilde{\boldsymbol{\pi}}^i$, corresponding to a valid single-wavelength logical topology configuration: let $\tilde{\pi}^i_{kl} = 1$ if logical link $k \to l$ is enabled on the i-th color of the RWA of $\mathbf{T}$ employing $W^{\text{nc}}(\mathbf{T})$ wavelengths, and $\tilde{\pi}^i_{kl} = 0$ otherwise. Clearly $\tilde{\boldsymbol{\pi}}^i$ is a valid logical topology subject to the single-wavelength constraint, since the same configuration had a valid routing on the i-th color under the RWA of $\mathbf{T}$. Let the elements of Π_N be indexed by $\boldsymbol{\pi}^1, \ldots, \boldsymbol{\pi}^{|\Pi_N|}$, where $|\Pi_N|$ is the cardinality of Π_N. Thus, it must be true that

$$\begin{aligned}
\boldsymbol{\lambda} &= \frac{1}{W} \sum_{j=1}^{W^{\text{nc}}(\mathbf{T})} \tilde{\boldsymbol{\pi}}^j, \\
&= \sum_{i=1}^{|\Pi_N|} \frac{\sum_{j=1}^{W^{\text{nc}}(\mathbf{T})} \mathbb{1}_{\{\tilde{\boldsymbol{\pi}}^j = \boldsymbol{\pi}^i\}}}{W} \boldsymbol{\pi}^i, \\
&= \sum_{i=1}^{|\Pi_N|} \alpha^i \boldsymbol{\pi}^i, \qquad (4.16)
\end{aligned}$$

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function and for all i,

$$\alpha^i \triangleq \left(\sum_{j=1}^{W^{\text{nc}}(\mathbf{T})} \mathbb{1}_{\{\tilde{\boldsymbol{\pi}}^j = \boldsymbol{\pi}^i\}} \right) / W.$$

By definition we have that $\alpha^i \geq 0, \forall i$, and since $W \geq W^{\text{nc}}(\mathbf{T})$, $\sum_i \alpha^i \leq 1$. We conclude that $\boldsymbol{\lambda} \in \mathbf{\Lambda}^*_{\text{sh}}$.

Next, suppose $\boldsymbol{\lambda} \in \text{cl}(\mathcal{R}^{\text{nc}}_{\mathcal{P}}) \setminus \mathcal{R}^{\text{nc}}$. By the definition of the closure of a set, there must exist a sequence $\{\boldsymbol{\lambda}^k\}$, with $\boldsymbol{\lambda}^k \in \mathcal{R}^{\text{nc}}$ for all k, such that $\boldsymbol{\lambda}^k \to \boldsymbol{\lambda}$ as $k \to \infty$. From (4.16), each $\boldsymbol{\lambda}^k$ has a RWA decomposition given by

$$\boldsymbol{\lambda}^k = \sum_{i=1}^{|\Pi_N|} \alpha^i_k \boldsymbol{\pi}^i.$$

For each k, the vector $(\alpha^1_k, \ldots, \alpha^{|\Pi_N|}_k)$ belongs to the compact set of non-negative real vectors having L^1 norm no greater than one. Using this compactness property, the Bolzano-Weierstrass Theorem [93] guarantees the existence of a vector $(\alpha^1, \ldots, \alpha^{|\Pi_N|})$ and a subsequence $\{k_j\}_{j=1}^{\infty}$ with

$$\alpha^i_{k_j} \to \alpha^i \text{ as } j \to \infty, \quad \text{for } i = 1, \ldots, |\Pi_N|. \qquad (4.17)$$

To demonstrate that $\boldsymbol{\lambda} = \sum_i \alpha^i \pi^i$, we make use of the following chain of relations. Let $\varepsilon > 0$, and let $\|\cdot\|$ be the L^1 norm operator.

$$\begin{aligned}\left\|\boldsymbol{\lambda} - \sum_i \alpha^i \pi^i\right\| &\leq \left\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^{k_j}\right\| + \left\|\boldsymbol{\lambda}^{k_j} - \sum_i \alpha^i \pi^i\right\|, \\ &= \left\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^{k_j}\right\| + \left\|\sum_{i=1}^{|\Pi_N|} (\alpha^i_{k_j} - \alpha^i)\pi^i\right\|, \\ &< \varepsilon. \end{aligned} \tag{4.18}$$

Equation (4.18) follows for j sufficiently large from the convergence property of the sequence $\{\boldsymbol{\lambda}^k\}$ and by (4.17). Finally, we have that $\alpha^i_k \geq 0, \forall k, i$, and that $\sum_i \alpha^i_k \leq 1, \forall k$, from which it must be true that the limiting quantities $\alpha^1, \ldots, \alpha^{|\Pi_N|}$ satisfy $\alpha^i \geq 0, \forall i$, and $\sum_i \alpha^i \leq 1$. This implies $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*_{\text{sh}}$.

Proof that $\boldsymbol{\Lambda}^*_{\text{sh}} \subseteq \text{cl}(\mathcal{R}^{\text{nc}})$*:* Suppose $\boldsymbol{\lambda} \notin \text{cl}(\mathcal{R}^{\text{nc}})$. Then we must show that $\boldsymbol{\lambda} \notin \boldsymbol{\Lambda}^*_{\text{sh}}$. Suppose $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*_{\text{sh}}$. Then there exist $\alpha^1, \ldots, \alpha^{|\Pi_N|}$ such that $\boldsymbol{\lambda} = \sum_i \alpha^i \pi^i$. Define α^i_k to be the value α^i truncated to k decimal places. This truncation ensures that $\alpha^i_k \geq 0, \forall i, k$, $\sum_i \alpha^i_k \leq 1, \forall k$, and that $\alpha^i_k \to \alpha^i$ as $k \to \infty$ for $i = 1, \ldots, |\Pi_N|$. For each k, define $\boldsymbol{\lambda}^k = \sum_i \alpha^i_k \pi^i$, $\mathbf{T}^k = 10^k \boldsymbol{\lambda}^k$, and $W_k = \sum_i 10^k \alpha^i_k$. Clearly $\mathbf{T}^k$ is an integer matrix for every k. The decomposition property of $\boldsymbol{\lambda}^k$ implies

$$\mathbf{T}^k = \sum_{i=1}^{|\Pi_N|} 10^k \alpha^i_k \pi^i. \tag{4.19}$$

Since $10^k \alpha^i_k$ is an integer for all i, k, we may interpret (4.19) as a valid RWA for traffic $\mathbf{T}^k$ using $W_k \leq 10^k$ wavelengths. This follows because each π^i can be routed on a single wavelength. By definition, it must be true that $W_k \geq W^{\text{nc}}(\mathbf{T}^k)$. Thus, $\boldsymbol{\lambda}^k = \mathbf{T}^k / W_k \in \mathcal{R}^{\text{nc}}$ for each k. Since $\boldsymbol{\lambda}^k \to \boldsymbol{\lambda}$, then $\boldsymbol{\lambda} \in \text{cl}(\mathcal{R}^{\text{nc}})$, which is a contradiction.

4.C Proof of Theorem 4.3.2

Proof that $\text{cl}(\mathcal{R}^{\text{c}}) \subseteq \boldsymbol{\Lambda}^*_{\text{mh}}$*:* Suppose $\boldsymbol{\lambda} \in \mathcal{R}^{\text{c}}$, which by definition implies there must exist $\mathbf{T}, W$ such that $\boldsymbol{\lambda} = (1/W)\mathbf{T}$, with $\mathbf{T} \in \mathbb{Z}^m_+$ and $W \geq W^{\text{c}}(\mathbf{T})$. We establish a RWA decomposition for $\boldsymbol{\lambda}$ from the RWA for $\mathbf{T}$ using $W^{\text{c}}(\mathbf{T})$ wavelengths as follows.

Suppose the RWA for $\mathbf{T}$ employs logical link $j \to k$ on wavelength i. Starting with $m \times n$ matrix $\tilde{\mathbf{S}}^i = 0$, we build the service activation matrix corresponding to the i-th wavelength as follows. Suppose index l corresponds to link $j \to k$.

1. If $j \to k$ is the terminal fragment of an end-to-end lightpath (whether or not wavelength conversion occurs on the lightpath), then assign $\tilde{S}^i_{lk} \leftarrow 1$.

2. If $j \to k$ is not a terminal fragment, and instead wavelength conversion at node k is employed, with the ultimate destination of the multi-color lightpath being node v, then assign $\tilde{S}^i_{lv} \leftarrow 1$.

Applying this procedure to all logical links on all wavelengths $i = 1, \ldots, W^{\mathrm{c}}(\mathbf{T})$, we now claim that the $\tilde{\mathbf{S}}^i$ are valid service activation matrices, subject to the single wavelength constraint, and that the (j,k)-th element of $\mathbf{T}$ can be expressed as

$$T_{jk} = \sum_{i=1}^{W^{\mathrm{c}}(\mathrm{T})} \mathbf{R}^k_{j:} \tilde{\mathbf{S}}^i_{:k} = \sum_{i=1}^{W^{\mathrm{c}}(\mathrm{T})} d_{jk}(\tilde{\mathbf{S}}^i). \tag{4.20}$$

Since $\tilde{\mathbf{S}}^i$ is built from the RWA on wavelength i, $\tilde{\mathbf{S}}^i$ must be a valid single-wavelength service activation matrix for each i. Consider the value T_{jk}. We need only consider values l such that $\tilde{S}^i_{lk} = 1$. In this case, by the definition of matrix $\mathbf{R}^k$, if the source node of link l is j then $R^k_{jl} = 1$, and if the destination node of link l is j then $R^k_{jl} = -1$, and otherwise $R^k_{jl} = 0$. Thus the quantity at right in (4.20) is equivalent to

$$\sum_{i=1}^{W^{\mathrm{c}}(\mathrm{T})} \left(\sum_{\{l:\sigma(l)=j\}} \tilde{S}^i_{lk} - \sum_{\{l:\tau(l)=j\}} \tilde{S}^i_{lk} \right). \tag{4.21}$$

Equation (4.21) assigns unit weight to each logical link sourced at node j that is carrying traffic destined to node k, and assigns weight negative one to each logical link that terminates at node j and carries traffic destined to node k. Since this sum is carried out over all activation matrices corresponding to the RWA of $\mathbf{T}$, this ensures that only traffic sourced at node j for node k is counted in the overall sum in (4.21). Since this holds for all j, k, it must be true that (4.21) equals $\mathbf{T}$, as desired. Thus, for $W \geq W^{\mathrm{c}}(\mathbf{T})$, the (j,k)-th entry of matrix $\boldsymbol{\lambda}$ can be expressed as

$$\begin{aligned} \lambda_{jk} &= \sum_{i=1}^{W^{\mathrm{c}}(\mathrm{T})} \frac{1}{W} d_{jk}(\tilde{\mathbf{S}}^i), \\ &= \sum_{i=1}^{|\mathcal{S}^{\mathrm{mh}}|} \frac{\sum_{j=1}^{W^{\mathrm{c}}(\mathrm{T})} \mathbb{1}_{\{\tilde{\mathbf{S}}^j = \mathbf{S}^i\}}}{W} d_{jk}(\mathbf{S}^i), \\ &= \sum_{i=1}^{|\mathcal{S}^{\mathrm{mh}}|} \alpha^i d_{jk}(\mathbf{S}^i), \end{aligned} \tag{4.22}$$

where for all i, $\alpha^i \triangleq (1/W) \sum_{j=1}^{W^{\mathrm{c}}(\mathrm{T})} \mathbb{1}_{\{\tilde{\mathbf{S}}^j = \mathbf{S}^i\}}$. By definition we have that $\alpha^i \geq 0, \forall i$, and since $W \geq W^{\mathrm{c}}(\mathbf{T})$, $\sum_i \alpha^i \leq 1$. Consequently, (4.22) implies that $\boldsymbol{\lambda}$ is an admissible arrival rate matrix, and we conclude that $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*_{\mathrm{mh}}$.

Next, suppose $\boldsymbol{\lambda} \in \mathrm{cl}(\mathcal{R}^{\mathrm{c}}) \backslash \mathcal{R}^{\mathrm{c}}$. By a similar argument used in the proof of Theorem 4.3.1, there must exist sequences $\{\boldsymbol{\lambda}^k\}$ and $\{(\alpha^1_k, \ldots, \alpha^{|\mathcal{S}^{\mathrm{mh}}|}_k)\}$ such that $\boldsymbol{\lambda}^k \to \boldsymbol{\lambda}$, and $\alpha^i_k \to \alpha^i$ for $i = 1, \ldots, |\mathcal{S}^{\mathrm{mh}}|$, as $k \to \infty$. Furthermore, it can be shown that the limits of these sequences satisfy $\lambda_{jk} = \sum_i \alpha^i d_{jk}(\mathbf{S}^i)$, for $j, k \in V$. This establishes that $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*_{\mathrm{mh}}$.

Proof that $\boldsymbol{\Lambda}^* \subseteq \mathrm{cl}(\mathcal{R}^{\mathrm{c}})$*:* Suppose $\boldsymbol{\lambda} \notin \mathrm{cl}(\mathcal{R}^{\mathrm{c}})$. By a similar argument to the proof of Theorem 4.3.1, if $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*_{\mathrm{mh}}$, we can construct a decimal-truncated sequence of arrival rate matrices $\{\boldsymbol{\lambda}^k\}$, satisfying $\boldsymbol{\lambda}^k \to \boldsymbol{\lambda}$. Defining $\mathbf{T}^k = 10^k \boldsymbol{\lambda}^k$, $W_k = \sum_i 10^k \alpha^i_k$, and making use of the decomposition property

$$T^k_{jj'} = \sum_{i=1}^{|\mathcal{S}^{\mathrm{mh}}|} 10^k \alpha^i_k d_{jj'}(\mathbf{S}^i), \quad j, j' \in V, \tag{4.23}$$

where $10^k\alpha^i_k$ is an integer for all i, k, it is clear that (4.23) is a valid RWA for traffic $\mathbf{T}^k$ using $W_k \leq 10^k$ wavelengths. This follows because each $\mathbf{S}^i$ can be routed on a single wavelength. By definition, it must be true that $W_k \geq W^{\mathrm{c}}(\mathbf{T}^k)$. Thus, $\boldsymbol{\lambda}^k = (1/W_k)\mathbf{T}^k \in \mathcal{R}^{\mathrm{c}}$ for each k. Since $\boldsymbol{\lambda}^k \to \boldsymbol{\lambda}$, then $\boldsymbol{\lambda}^k \in \mathrm{cl}(\mathcal{R}^{\mathrm{c}})$, which is a contradiction.

4.D Proof of Theorem 4.4.1

We consider the single-hop case only, since the multi-hop case follows similarly. Denote

$$\theta^* = \limsup_{k\to\infty} k/W^{\mathrm{nc}}(k\mathbf{J}).$$

From the definition of θ^*, there must exist a sequence $\{k_l\}$ such that $k_l \to \infty$ as $l \to \infty$, and

$$\frac{k_l}{W^{\mathrm{nc}}(k_l\mathbf{J})} \to \theta^*. \tag{4.24}$$

Define the uniform arrival rate matrix $\boldsymbol{\lambda}^l = k_l\mathbf{J}/W^{\mathrm{nc}}(k_l\mathbf{J})$. From the definition of the set $\mathcal{R}^{\mathrm{nc}}$, we have that $\boldsymbol{\lambda}^l \in \mathcal{R}^{\mathrm{nc}}$ for all l. Due to the convergence property (4.24), it must be true that $\theta^*\mathbf{J} \in \mathrm{cl}(\mathcal{R}^{\mathrm{nc}})$. By Theorem 4.3.1 we then have that $\theta^*\mathbf{J} \in \mathbf{\Lambda}^*_{\mathrm{sh}}$.

Conversely, suppose that $\boldsymbol{\lambda}$ is a uniform arrival rate matrix, with uniform arrival rate $r > \theta^*$, for which $\boldsymbol{\lambda} \in \mathbf{\Lambda}^*_{\mathrm{sh}}$. Theorem 4.3.1 provides that $\boldsymbol{\lambda} \in \mathrm{cl}(\mathcal{R}^{\mathrm{nc}})$. Thus, there must exist a sequence of matrices $\{\boldsymbol{\lambda}^k\}$ such that $\boldsymbol{\lambda}^k \to \boldsymbol{\lambda}$ as $k \to \infty$, and $\boldsymbol{\lambda}^k \in \mathcal{R}^{\mathrm{nc}}$ for all k. Consequently, by the definition of the set $\mathcal{R}^{\mathrm{nc}}$, there must exist a sequence of traffics $\{\mathbf{T}^k\}$ and integers $\{W_k\}$ such that $\boldsymbol{\lambda}^k = \mathbf{T}^k/W_k$ with $W_k \geq W^{\mathrm{nc}}(\mathbf{T}^k)$ for all k. Define the sequence of traffics $\{\tilde{\mathbf{T}}^k\}$, with $\tilde{\mathbf{T}}^k = (\min_{i\neq j} T^k_{ij})\mathbf{J}$. Since $\lambda^k_{ij} \to r$ for all $i \neq j$, it must be true that $(\min_{i\neq j} \lambda^k_{ij}) \to r$. This implies that $\tilde{\mathbf{T}}^k/W_k = (\min_{i\neq j} T^k_{ij}/W_k)\mathbf{J} \to r\mathbf{J}$. Clearly, since the traffic $\tilde{\mathbf{T}}^k$ is integer and fully dominated (entry-by-entry) by $\mathbf{T}^k$, it must be true that $\tilde{\mathbf{T}}^k$ can be satisfied using W_k wavelengths. This follows by using the RWA for $\mathbf{T}^k$ using W_k wavelengths in order to build a RWA for $\tilde{\mathbf{T}}^k$ using W_k wavelengths. Since $r > \theta^*$, there must exist k^* such that when $k > k^*$, for $i \neq j$, $\tilde{T}^k_{ij}/W_k > \theta^*$. Since W_k wavelengths are sufficient for a RWA with no conversion of traffic $\tilde{\mathbf{T}}^k$, we must have that $W_k \geq W^{\mathrm{nc}}(\tilde{\mathbf{T}}^k)$. Thus for all $i \neq j$, $\tilde{T}^k_{ij}/W^{\mathrm{nc}}(\tilde{\mathbf{T}}^k) > \theta^*$, which implies by the definition of $\tilde{\mathbf{T}}^k$ that for $k > k^*$,

$$\frac{\min_{i\neq j} T^k_{ij}}{W^{\mathrm{nc}}((\min_{i\neq j} T^k_{ij})\mathbf{J})} > \theta^*. \tag{4.25}$$

For integer $c > 0$, the traffic $c\tilde{\mathbf{T}}^k$ can be satisfied using cW_k wavelengths, by simply repeating the RWA for traffic $\tilde{\mathbf{T}}^k$ a total of c times. Consequently, we must have $W^{\mathrm{nc}}(c\tilde{\mathbf{T}}^k) \leq cW_k$.

Combining this fact with (4.25), we have for any $k > k^*$, and any $c \geq 1$,

$$\frac{c \min_{i \neq j} T_{ij}^k}{W^{\text{nc}}(c(\min_{i \neq j} T_{ij}^k)\mathbf{J})} > \theta^*.$$

This violates the definition of θ^* and provides a contradiction.

4.E Proof of Theorem 4.4.2

In this appendix, we focus on the single-hop quantity, α^{sh}. The proof for the multi-hop quantity α^{mh} follows identically. Denote $\alpha^* = \limsup_{k \to \infty} k/\mathcal{W}^{\text{nc}}(k)$.

Definition 4.E.1 *Let the set $\partial\mathcal{D}_s$ denote the set of doubly substochastic matrices having at least one row or column sum equal to s: $\partial\mathcal{D}_s = \{\boldsymbol{\lambda} \in \mathcal{D}_s : \|\boldsymbol{\lambda}\|_{\max} = s\}$.*

Proof that $\alpha^{\text{sh}} \geq \limsup_{k \to \infty} k/\mathcal{W}^{\text{nc}}(k)$: Suppose $\boldsymbol{\lambda} \in \mathcal{D}_{\alpha^*}$, with $\boldsymbol{\lambda} \neq 0$ (since $\boldsymbol{\lambda} = 0$ has a trivial RWA decomposition). Define the sequence of integer traffic matrices $\{\mathbf{T}^k\}$, such that for $i \neq j$,

$$T_{ij}^k = (\lfloor \lambda_{ij} \mathcal{W}^{\text{nc}}(k) - \eta_k \rfloor)^+ .$$

Here, the operator $(\cdot)^+$ sets to zero any negative elements of its matrix operand, and $\lfloor \cdot \rfloor$ is the floor operator. We seek to ensure that $\mathbf{T}^k \in \mathcal{K}_k$, $\forall k$. To this end, consider the following series of relations. For sequence $\{\eta_k\}$, which we define subsequently, and k sufficiently large,

$$\begin{aligned} \|\mathbf{T}^k\|_{\max} &= \left\|(\lfloor \boldsymbol{\lambda} \mathcal{W}^{\text{nc}}(k) - \eta_k \mathbf{J} \rfloor)^+\right\|_{\max} \\ &\leq \left\|(\boldsymbol{\lambda} \mathcal{W}^{\text{nc}}(k) - \eta_k \mathbf{J})^+\right\|_{\max} && (4.26) \\ &\leq \left\|\boldsymbol{\lambda} \mathcal{W}^{\text{nc}}(k)\right\|_{\max} - \eta_k && (4.27) \\ &\leq \alpha^* \mathcal{W}^{\text{nc}}(k) - \eta_k && (4.28) \\ &\leq k + \varepsilon_k \mathcal{W}^{\text{nc}}(k) - \eta_k, && (4.29) \end{aligned}$$

where for $k \in \mathbb{Z}_+$,

$$\varepsilon_k = \sup_{\tilde{k} \geq k} \left| \frac{\tilde{k}}{\mathcal{W}^{\text{nc}}(\tilde{k})} - \alpha^* \right| .$$

In (4.26), if we assume that $\eta_k/\mathcal{W}^{\text{nc}}(k) \to 0$ as $k \to \infty$, then (4.27) follows for k sufficiently large, since there is at least one non-zero element on the row/column having maximum sum in $\boldsymbol{\lambda}$. Note that $\mathcal{W}^{\text{nc}}(k)$ increases at least linearly in k. Since $\boldsymbol{\lambda} \in \mathcal{D}_{\alpha^*}$, (4.28) must follow. By (4.10) we then have (4.29).

To ensure $\mathbf{T}^k \in \mathcal{K}_k$, we simply choose $\eta_k = \varepsilon_k \mathcal{W}^{\text{nc}}(k)$. Clearly, $\eta_k/\mathcal{W}^{\text{nc}}(k) \to 0$ as $k \to \infty$, since (4.10) implies that $\varepsilon_k \to 0$ as $k \to \infty$. Next, define $\boldsymbol{\lambda}^k = (1/\mathcal{W}^{\text{nc}}(k))\mathbf{T}^k$. Since $\mathbf{T}^k \in \mathcal{K}_k$, it must be true that $\boldsymbol{\lambda}^k \in \mathcal{R}^{\text{nc}}$. To demonstrate that $\boldsymbol{\lambda}$ has a RWA decomposition, we need to show that $\boldsymbol{\lambda}^k \to \boldsymbol{\lambda}$ as $k \to \infty$. Since $\eta_k/(\mathcal{W}^{\text{nc}}(k)) \to 0$ as $k \to \infty$, this is clearly true. Thus, $\boldsymbol{\lambda} \in \text{cl}(\mathcal{R}^{\text{nc}})$, which implies by Theorem 4.3.1 that $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_{\text{sh}}^*$. Since this holds for all $\boldsymbol{\lambda} \in \mathcal{D}_{\alpha^*}$, it must be true that $\alpha^{\text{sh}} \geq \alpha^*$.

Proof that $\alpha^{\text{sh}} \leq \limsup_{k\to\infty} k/\mathcal{W}^{\text{nc}}(k)$: Suppose there exists $\alpha > \alpha^*$ such that $\mathcal{D}_\alpha \subseteq \boldsymbol{\Lambda}^*_{\text{sh}}$. Consider any positive integer u. Let $\boldsymbol{\lambda}^{u,1}, \ldots, \boldsymbol{\lambda}^{u,K_u}$ be a finite set of matrices belonging to $\partial\mathcal{D}_\alpha$, such that

$$\partial\mathcal{D}_\alpha \subseteq \bigcup_{l=1}^{K_u} \left\{\boldsymbol{\lambda} : |\lambda_{ij} - \lambda_{ij}^{u,l}| \leq 1/u,\ \forall i,j \in V\right\}.$$

In words, the set of points $\{\boldsymbol{\lambda}^{u,1}, \ldots, \boldsymbol{\lambda}^{u,K_u}\}$ are the center locations of a set of $(1/u)$-balls that cover the outer boundary $\partial\mathcal{D}_\alpha$. The compactness of $\mathcal{D}_\alpha$ is sufficient to ensure the existence of a covering such that K_u is finite-valued [93].

Since $\boldsymbol{\lambda}^{u,l} \in \mathcal{D}_\alpha$, and by our assumption that $\mathcal{D}_\alpha \subseteq \boldsymbol{\Lambda}^*_{\text{sh}}$, Theorem 4.3.1 provides that there must exist a set of integer traffics $\{\mathbf{T}^{u,1}, \ldots, \mathbf{T}^{u,K_u}\}$, and a set of positive integers $\{W^{u,1}, \ldots, W^{u,K_u}\}$ such that for $l = 1, \ldots, K_u$,

$$\frac{1}{W^{u,l}}\mathbf{T}^{u,l} \in \left\{\boldsymbol{\lambda} = (\lambda_{ij}) : |\lambda_{ij} - \lambda_{ij}^{u,l}| \leq 1/u,\ \forall i,j \in V\right\}. \tag{4.30}$$

where $W^{u,l} \geq W^{\text{nc}}(\mathbf{T}^{u,l})$ for all l. Since K_u is finite and $\mathbf{T}^{u,l}$ is an integer matrix for all l, there must exist integers $\kappa_1^u, \ldots, \kappa_{K_u}^u$ and k_u^*, such that for $l = 1, \ldots, K_u$,

$$\kappa_l^u \|\mathbf{T}^{u,l}\|_{\max} = k_u^*.$$

The integer traffic $\kappa_l^u \mathbf{T}^{u,l}$ must have a RWA using $\kappa_l^u W^{u,l}$ wavelengths. This RWA is constructed by repeating the RWA for traffic $\mathbf{T}^{u,l}$, that makes use of $W^{u,l}$ wavelengths, a total of κ_l^u times over $\kappa_l^u W^{u,l}$ wavelengths. While the maximum row/column sum of $\boldsymbol{\lambda}^{u,l}$ is α, that of $(\kappa_l^u/W^{u,l})\mathbf{T}^{u,l}$ is $k_u^*/W^{u,l}$ for each l. Applying (4.30), we then have for $l = 1, \ldots, K_u$,

$$\left|\alpha - \frac{k_u^*}{\kappa_l^u W^{u,l}}\right| \leq \frac{n-1}{u}. \tag{4.31}$$

Consider any traffic $\mathbf{T} \in \mathcal{K}_{k_u^*}$, with maximum row/column sum equal to k_u^*. Then $(\alpha/k_u^*)\mathbf{T} \in \partial\mathcal{D}_\alpha$, which implies there exists l^* such that for all $i,j \in V$,

$$\left|\frac{\alpha}{k_u^*}T_{ij} - \lambda_{ij}^{u,l^*}\right| \leq \frac{1}{u}. \tag{4.32}$$

Combining (4.30) with (4.32), we have

$$\left|\frac{\alpha}{k_u^*}T_{ij} - \frac{T_{ij}^{u,l^*}}{W^{u,l^*}}\right| \leq \frac{2}{u}. \tag{4.33}$$

Multiplying (4.33) through by $\kappa_{l^*}^u W^{u,l^*}$ provides

$$\left|\frac{\alpha\kappa_{l^*}^u W^{u,l^*}}{k_u^*}T_{ij} - \kappa_{l^*}^u T_{ij}^{u,l^*}\right| \leq \frac{2}{u}\kappa_{l^*}^u W^{u,l^*}. \tag{4.34}$$

Note that if $a > 0$, and $|ax - y| \leq c$, then if $ax - y > 0$, we have $x - y \leq c/a + ((1-a)/a)y$, and if $ax - y < 0$, we have $x - y \leq ((1-a)/a)y$. Consequently, equation (4.34) implies

$$\left|T_{ij} - \kappa_{l^*}^u T_{ij}^{u,l^*}\right| \leq \frac{2}{u}\kappa_{l^*}^u W^{u,l^*} \frac{k_u^*}{\alpha \kappa_{l^*}^u W^{u,l^*}} + \kappa_{l^*}^u T_{ij}^{u,l^*} \left(\frac{k_u^*}{\alpha \kappa_{l^*}^u W^{u,l^*}} - 1\right).$$

The difference between the integer traffic demand matrix $\mathbf{T}$ and the matrix $\kappa_{l^*}^u \mathbf{T}^{u,l^*}$ can then be bounded as

$$\begin{aligned}\sum_{i,j} \left|T_{ij} - \kappa_{l^*}^u T_{ij}^{u,l^*}\right| &\leq n(n-1)\frac{2}{u}\kappa_{l^*}^u W^{u,l^*} \frac{k_u^*}{\alpha \kappa_{l^*}^u W^{u,l^*}} + n k_u^* \left(\frac{k_u^*}{\alpha \kappa_{l^*}^u W^{u,l^*}} - 1\right) \\ &\triangleq \omega_{u,l^*}.\end{aligned}$$

Then,

$$\frac{\omega_{u,l^*}}{k_u^*} = n(n-1)\frac{2}{\alpha u} + n\left(\frac{k_u^*}{\alpha \kappa_{l^*}^u W^{u,l^*}} - 1\right).$$

Applying (4.31), it is clear that $\omega_{u,l^*}/k_u^* \to 0$ as $u \to \infty$.

If each additional integer demand in traffic $\mathbf{T}$ over that in traffic $\kappa_{l^*}^u \mathbf{T}^{u,l^*}$ is serviced using a unique wavelength, the value of ω_{u,l^*} can be used to infer an upper bound on the minimum number of wavelengths required to service $\mathbf{T}$. This holds, given the appropriate choice for the index l^*, for any $\mathbf{T} \in \mathcal{K}_{k_u^*}$ having maximum row/column sum of k_u^*, from which we obtain,

$$\mathcal{W}^{\text{nc}}(k_u^*) \leq \max_l (\kappa_l^u W^{u,l} + \omega_{u,l}).$$

Thus, we obtain

$$\frac{k_u^*}{\mathcal{W}^{\text{nc}}(k_u^*)} \geq \frac{k_u^*}{\max_l \kappa_l^u W^{u,l} + \max_l \omega_{u,l}}. \tag{4.35}$$

Applying (4.31), the right side of (4.35) must converge to α as $u \to \infty$. Thus, there must exist $\bar{u}$ such that for $u \geq \bar{u}$,

$$\frac{k_u^*}{\mathcal{W}^{\text{nc}}(k_u^*)} \geq \frac{\alpha + \alpha^{\text{sh}}}{2} > \alpha^{\text{sh}}.$$

Clearly, if $k_u^* \to \infty$, this is in violation of (4.10), which provides a contradiction. Thus, it remains to show that $k_u^* \to \infty$ as $u \to \infty$. Suppose this is not true, and there exists integer $\tilde{k}^*$ such that $k_u^* \leq \tilde{k}^*$ for all u. We can then bound the cardinality of $\mathcal{K}_{\tilde{k}^*}$ as $|\mathcal{K}_{\tilde{k}^*}| \leq (\tilde{k}^*)^{n(n-1)}$. The number of distinct $(1/u)$-balls required to cover $\partial\mathcal{D}_\alpha$ must increase with u. This can be seen as follows: consider any two neighboring (sharing the same face) non-zero vertices of $\mathcal{D}_\alpha$. The line segment joining these two vertices is completely contained in $\partial\mathcal{D}_\alpha$. This line segment is isomorphic to an interval of equal length on the real line, for which a covering by $(1/u)$-balls clearly requires an increasing number of balls as u increases. Furthermore, since the line segment is not collinear with the origin (this would violate that one of the end points is a vertex of $\mathcal{D}_\alpha$), the number of covering $(1/u)$-balls that exist such that no two balls contain *any* matrices that are scaled versions of one another, is also increasing with u. Consequently, for sufficiently large u, there must be more than $(\tilde{k}^*)^{n(n-1)}$ traffics in the set

$\{\mathbf{T}^{u,1}, \ldots, \mathbf{T}^{u,K_u}\}$ that are not scaled versions of one another. Since the line joining each of these traffics to the origin has a unique direction, the common boundary that these traffics will be scaled to (using the integers from the set $\{\kappa_l^u\}$) must contain more than $(\tilde{k}^*)^{n(n-1)}$ integer matrices. This however, is in violation of our assumption that $|\mathcal{K}_{k_u^*}| \leq (\tilde{k}^*)^{n(n-1)}$ for all u. Thus, $k_u^* \rightarrow \infty$ as $u \rightarrow \infty$.

Chapter 5

Achieving 100% throughput in reconfigurable optical networks: Extensions

In this chapter, we consider again the optical networking architecture introduced in Chapter 3. Chapter 4 focused on quantifying the throughput properties of single-wavelength WDM-based packet networks. Here, we seek to understand the computability of the geometric properties studied in Chapter 4, as well as to generalize those results to the multi-wavelength scenario.

Our analysis begins by demonstrating that the uniform all-to-all geometric property characterized in Theorem 4.4.1 can be determined efficiently in the multi-hop scenario, and is likely to be difficult to compute efficiently in the single-hop scenario.

Subsequently, we generalize the mathematical characterizations of the capacity regions as well as their geometric properties to multi-wavelength networks.

When every fiber link is equipped with $w \geq 1$ wavelengths, we seek to understand the capacity region of the network in terms of the single-wavelength capacity region, characterized in Chapter 4. We study the *round-up property* of fractional chromatic number, and characterize the capacity region of multi-wavelength networks adhering to this property.

5.1 Computability of geometric properties

In Chapter 4, we characterized the maximum uniform all-to-all arrival rate matrix that can be stabilized in single-wavelength networks. This geometric property was labeled θ^{sh} for single-hop networks, and θ^{mh} for multi-hop networks, and serves as a natural measure for assessing the throughput capabilities of single-wavelength networks. This property is linked in Theorem 4.4.1 to the RWA problem, which is known to be NP-hard[1]. The following theorems establish the computational complexity associated with determining θ^{sh} and θ^{mh}.

Theorem 5.1.1 *θ^{mh} can be determined in polynomial time. In particular, $1/\theta^{\mathrm{mh}} = \bar{W}^{c}(\mathbf{J})$,*

[1] See [42], where it is shown that the wavelength assignment problem, a subproblem of the RWA problem, is NP-complete.

where $\bar{W}^{\mathrm{c}}(\mathbf{J})$ is the solution to the relaxed version of the RWA problem with wavelength conversion.

Proof: See Appendix 5.A. ■

The importance of Theorem 5.1.1 is its implication that determining the maximum uniform arrival rate matrix supportable in the reconfigurable WDM network is computationally feasible. Interestingly, the single-hop geometric property θ^{sh} turns out to be computationally difficult to obtain in general.

Theorem 5.1.2 *Determining θ^{sh} is an NP-hard problem.*

Proof: See Appendix 5.B. ■

The proofs of Theorems 5.1.1 and 5.1.2 suggest that the capacity regions $\mathbf{\Lambda}^*_{\mathrm{mh}}$ and $\mathbf{\Lambda}^*_{\mathrm{sh}}$ have as natural counterparts the relaxed versions of the RWA with and without wavelength conversion, respectively.

5.2 Generalized traffic decompositions

Our network model in Section 3.2 is sufficiently general that it applies much more broadly than in WDM networks having a single wavelength per optical fiber. In particular, the model can accommodate any number of wavelengths available in each fiber, and other architectural assumptions that affect the logical topologies and electronic routing allowed in the network. For such a network, designate by $\mathcal{S}$ the set of allowable service activation matrices. Recall from Section 3.2 that every matrix belonging to $\mathcal{S}$ jointly represents a valid logical topology and electronic routing. As in definitions 2.2.1 and 2.2.4, we designate by $\mathbf{\Lambda}^*$ the capacity region of arrival rates that can be rate stabilized when the service activation set in network N is $\mathcal{S}$.

The following definition generalizes the RWA functions $W^{\mathrm{nc}}, W^{\mathrm{c}}$ to this more general network setting.

Definition 5.2.1 *For the non-negative integer matrix $\mathbf{T} = (T_{ij})$, let $\chi(\mathbf{T})$ equal the minimum number of service activation matrices belonging to $\mathcal{S}$ required to decompose $\mathbf{T}$:*

$$\chi(\mathbf{T}) = \min\left\{k : \exists \mathbf{S}^1, \ldots, \mathbf{S}^k \in \mathcal{S},\, \mathbf{T} \leq \sum_{l=1}^{k} \mathbf{d}(\mathbf{S}^l)\,\forall i,j\right\}$$

The following theorem generalizes Theorems 4.3.1 and 4.3.2 to multi-wavelength networks. Its proof follows identically to the single-wavelength proof in Appendix 4.B, only replacing the RWA function W^{nc} with χ.

Theorem 5.2.1 *Define the set $\mathcal{R}$ as the set of integer traffic matrices scaled by their respective χ values,*

$$\mathcal{R} = \left\{\boldsymbol{\lambda} = \frac{1}{W}\mathbf{T} : \mathbf{T} \in \mathbb{Z}_+^m,\, W \in \mathbb{Z}_+,\, W \geq \chi(\mathbf{T})\right\}.$$

Then $\mathbf{\Lambda}^* = \mathrm{cl}(\mathcal{R})$.

While Theorem 5.2.1 broadens the class of networks to which generalized RWA decompositions can be applied to characterize network capacity properties, its key drawback is that it relies on the function χ, which does not in general tie to a well-studied optimization problem. This is in contrast to the special case of single-wavelength networks, which we studied in Chapter 4, where the capacity region was fully characterized in terms of the well-known RWA problem.

5.3 Additional geometric properties

Theorem 4.4.2 can be extended to provide for any polytope the maximum scale factor such that the scaled polytope remains within the capacity region. Define the set of integer matrices in the scaled region $k\mathcal{P}$ as $\mathcal{K}_k^{\mathcal{P}}$:

$$\mathcal{K}_k^{\mathcal{P}} = \mathbb{Z}_+^m \cap k\mathcal{P}.$$

Further, define the maximum value of the function χ achieved over the set $\mathcal{K}_k^{\mathcal{P}}$ as $\mathcal{W}_{\mathcal{P}}(k)$:

$$\mathcal{W}_{\mathcal{P}}(k) = \max_{\mathbf{T}\in\mathcal{K}_k^{\mathcal{P}}} \chi(\mathbf{T}).$$

Finally, we call a set $\mathcal{P} \subseteq \mathbb{R}_+^{n\times n}$ *Pareto*, if for each $\tilde{\boldsymbol{\lambda}} \in \mathcal{P}$, if $\boldsymbol{\lambda} \leq \tilde{\boldsymbol{\lambda}}$ (entry-by-entry dominance) and $\boldsymbol{\lambda} \geq 0$, then $\boldsymbol{\lambda} \in \mathcal{P}$.

Theorem 5.3.1 *Let $\mathcal{P}$ be a convex, compact, full-dimensional, Pareto subset of $\mathbb{R}_+^m$, and $\alpha_{\mathcal{P}} = \sup\{\alpha : \alpha\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*, \forall\boldsymbol{\lambda} \in \mathcal{P}\}$. Then*

$$\alpha_{\mathcal{P}} = \limsup_{k\to\infty} \frac{k}{\mathcal{W}_{\mathcal{P}}(k)}.$$

Proof: The proof methodology is similar to that of Theorem 4.4.2, though it requires additional technical maneuvers. For completeness, the proof has been included in Appendix 5.C ■

Theorem 5.3.1 can be used to recover the result of Theorem 4.4.1, as follows. Consider the m-dimensional cube with edge lengths equal to unity, denoted by $\mathcal{P}_{\text{box}}$:

$$\mathcal{P}_{\text{box}} = \{\boldsymbol{\lambda} \in \mathbb{R}_+^m : \lambda_{ij} \leq 1, \forall i, j\}.$$

Then, if we define $\mathcal{K}_k^{\text{box}} = \mathbb{Z}_+^m \cap k\mathcal{P}_{\text{box}}$, it follows that

$$\mathcal{W}_{\mathcal{P}_{\text{box}}}^{\text{nc}}(k) \triangleq \max_{\mathbf{T}\in\mathcal{K}_k^{\text{box}}} W^{\text{nc}}(\mathbf{T}) = W^{\text{nc}}(k\mathbf{J}).$$

This is because $k\mathbf{J}$ is entry-by-entry greater or equal to each element of $\mathcal{K}_k^{\text{box}}$. Consequently, Theorem 5.3.1 implies that

$$\alpha_{\mathcal{P}_{\text{box}}}^{\text{sh}} = \limsup_{k\to\infty} k/W^{\text{nc}}(k\mathbf{J}) = \theta^{\text{sh}}.$$

An identical result follows in the multi-hop case.

Similarly, Theorem 5.3.1 can be used to recover the result of Theorem 4.4.2, by employing the set $\mathcal{P}_{\text{ds}} = \left\{\boldsymbol{\lambda} \in \mathbb{R}_+^{n\times n} : \|\boldsymbol{\lambda}\|_{\max} \leq 1\right\}$.

5.4 Connecting the multi-wavelength and single-wavelength capacity regions

While the characterizations of the multi-wavelength capacity region and stability properties of the previous sections are perfectly valid, they make use of the function χ, which is not well studied in the literature. This is in contrast to the single-wavelength scenario, where the capacity properties can be tied to the classical RWA problem, with or without wavelength conversion.

In this section, we take a different approach, and attempt to understand multi-wavelength capacity properties in terms of the single-wavelength characterization, studied in Chapter 4. This provides a picture of how the capacity regions expand from the single-wavelength scenario studied in Chapter 4 to the wavelength-unconstrained scenario studied in Chapter 3.

Intuitively, one might hope that the multi-wavelength case can be extended from the single-wavelength case through simple scaling. Thus, if the single-wavelength capacity region (single-hop or multi-hop) is given by the set $\boldsymbol{\Lambda}_1^*$, the w-wavelength capacity region, which we denote $\boldsymbol{\Lambda}_w^*$, would be given by $w\boldsymbol{\Lambda}_1^* = \{w\boldsymbol{\lambda} : \boldsymbol{\lambda} \in \boldsymbol{\Lambda}_1^*\}$. The set $w\boldsymbol{\Lambda}_1^*$ turns out to be an important element in the characterization of the w-wavelength capacity region. The following section provides illustrative examples showing how the set $w\boldsymbol{\Lambda}_1^*$ must be refined in order to attain the w-wavelength capacity region.

5.4.1 Scaling the single-wavelength capacity region: an example

Consider the network links depicted in Figure 5-1. Here, there are two source-destination pairs of interest: $(1, 2)$ and $(1, 3)$. The network admits a single logical link from node 1 to node 3, and two logical links from node 1 to node 2. Each $1 \to 2$ link shares a fiber with the $1 \to 3$ link, but does not share a fiber with the other $1 \to 2$ link.

Consider first the case $P_1 = 3$, $P_2 = P_3 = 2$, by which we mean that node 1 is equipped with 3 ports (transceivers), while nodes 2, 3 are equipped with 2 ports each. In the single-wavelength case, the set of available service configurations is given by

$$\{(0,0), (0,1), (1,0), (2,0)\},$$

where configuration (a, b) indicates that a logical links are active from node 1 to node 2, simultaneously with b links active from node 1 to node 3. The single-wavelength capacity region $\boldsymbol{\Lambda}_1^*$ is then depicted in Figure 5-2(a). We now consider the case of $w = 2$ wavelengths per optical fiber. Under the same port configuration, Figure 5-2(b) contains the scaled region $2\boldsymbol{\Lambda}_1^*$, along with the wavelength-unconstrained capacity region $\boldsymbol{\Lambda}_{\text{port}}^*$ with a dashed boundary. Clearly, the wavelength unconstrained case must serve as an outer bound on the capacity region in the 2 wavelength scenario. The intersected region $2\boldsymbol{\Lambda}_1^* \cap \boldsymbol{\Lambda}_{\text{port}}$ turns out

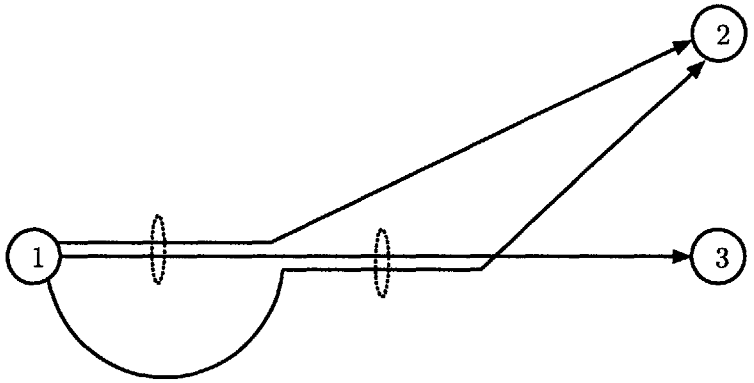


Figure 5-1: Network links considered in Section 5.4.1. Each 1 → 2 link interferes (shares a fiber) with the 1 → 3 link, but never with the other 1 → 2 path.

to be the capacity region in this scenario, because the set of 2 wavelength configurations that satisfy the port constraint is given by:

$$\{(0,0),(0,1),(0,2),(1,0),(1,1),(2,0),(2,1)\}.$$

These configurations are depicted as dots in Figure 5-2(b). Clearly, the convex hull of this set is the intersected region $2\Lambda_1^* \cap \Lambda_{\text{port}}$.

The intersection of the scaled single-wavelength region with the wavelength-unconstrained capacity region does not completely characterize the capacity region, as the following example demonstrates. Consider again the set of links depicted in Figure 5-1, with the following port values: $P_1 = 5, P_2 = P_3 = 3$. Once again, the single-wavelength capacity region Λ_1^* in this scenario is depicted in Figure 5-2(a). Figure 5-2(c) depicts the region $2\Lambda_1^*$ along with the port-loaded region Λ_{port} with a dashed boundary. In this case, the intersected region contains a non-integer corner point, which clearly cannot correspond to a valid service activation. The set of valid configurations in this case is depicted as dots in Figure 5-2(c) and is given by

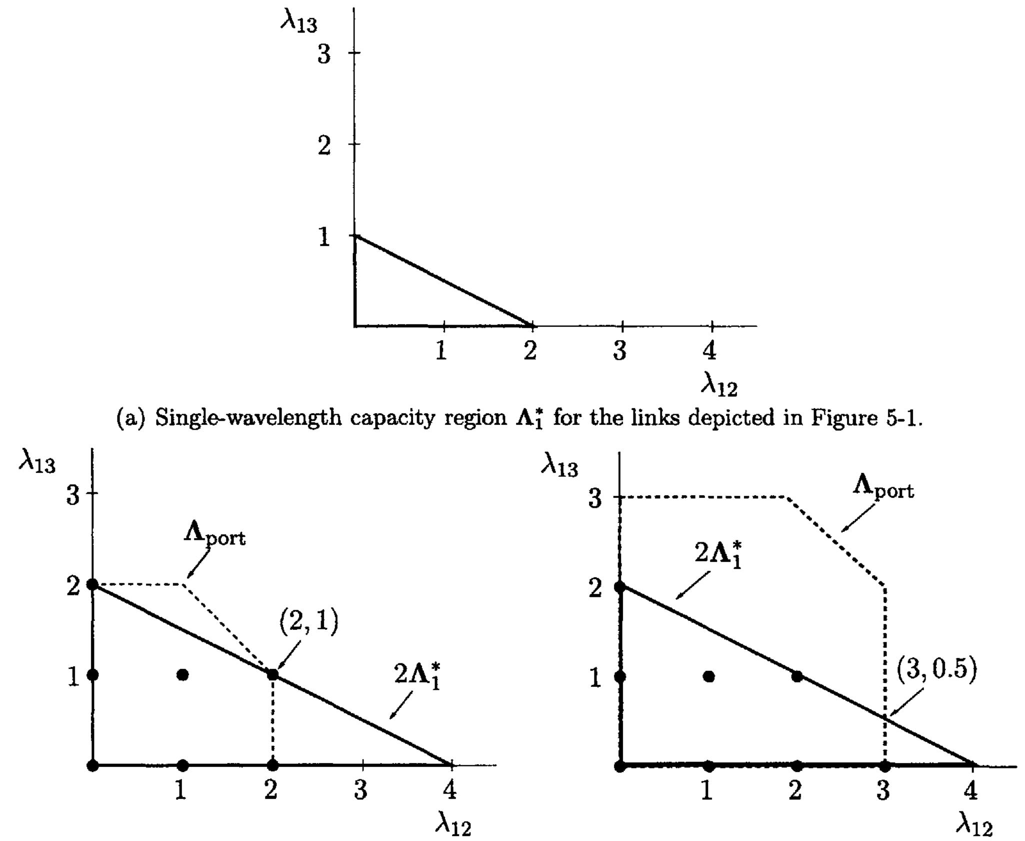
$$\{(0,0),(0,1),(0,2),(1,0),(1,1),(2,0),(2,1),(3,0)\}.$$

Thus, the 2-wavelength capacity region Λ_2 is given by the convex hull of the integer points contained in the intersected region $2\Lambda_1^* \cap \Lambda_{\text{port}}$. We next study the validity of this characterization of the general w-wavelength capacity region.

5.4.2 The w-wavelength capacity region

The above example suggests that the integer points contained in the region $w\Lambda_1^* \cap \Lambda_{\text{port}}$ form the set of service configurations available in the w-wavelength scenario. In this section we demonstrate the truth of this statement, subject to a property of the RWA function $W(\cdot)$, which we call the *round-up property*. We deliberately avoid designating $W(\cdot)$ as corresponding to wavelength conversion capability or a lack thereof, since the analysis in either case follows identically.

Definition 5.4.1 *The RWA function W satisfies the round-up property if for each* $\mathbf{T} \in$



(a) Single-wavelength capacity region Λ_1^* for the links depicted in Figure 5-1.

(b) 2 wavelength scenario under $P_1 = 3, P_2 = P_3 = 2$.

(c) 2 wavelength scenario under $P_1 = 5, P_2 = P_3 = 3$.

Figure 5-2: Towards an understanding of the 2-wavelength capacity region for the link structure of Figure 5-1.

$\Lambda_{\rm port} \cap \mathbb{Z}_+^m$, *there exists* $\varepsilon > 0$ *such that*

$$W(\mathbf{T}) - \inf_{k \in \mathbb{Z}_+} \frac{W(k\mathbf{T})}{k} \leq 1 - \varepsilon.$$

In words, satisfaction of the round-up property implies that for any non-negative integer traffic matrix $\mathbf{T}$, the number of wavelengths required in a static RWA for $\mathbf{T}$ should differ from the number of wavelengths required in a static RWA for $k\mathbf{T}$, normalized by k, by a value strictly less than 1, for every integer $k > 0$.

The following theorem, assuming the round-up property, states that the w-wavelength capacity region is given by the convex hull of the integer points contained in the intersected region $w\Lambda^* \cap \Lambda_{\rm port}$.

Theorem 5.4.1 *If the RWA function* $W^{\rm nc}$ *satisfies the round-up property, then for any integer* $w > 0$ *the* w*-wavelength single-hop capacity region, where the network has no wavelength conversion capability, is given by*

$$\Lambda_{w,\rm sh}^* = \operatorname{conv}\left(w\Lambda_{\rm sh}^* \cap \Lambda_{\rm port} \cap \mathbb{Z}_+^m\right). \qquad (5.1)$$

Similarly, if the RWA function W^c satisfies the round-up property, the w-wavelength multi-hop capacity region, where the network allows full wavelength conversion, is given by

$$\mathbf{\Lambda}^*_{w,\mathrm{mh}} = \mathrm{conv}\left(w\mathbf{\Lambda}^*_{\mathrm{mh}} \cap \mathbf{\Lambda}_{\mathrm{port}} \cap \mathbb{Z}^m_+\right). \tag{5.2}$$

Proof: See Appendix 5.D. ■

An immediate result of the proof of Theorem 5.4.1 is that in the general setting, namely irrespective of the satisfaction of the round-up property, the w-wavelength capacity regions can be bounded as follows.

$$\begin{aligned}\mathbf{\Lambda}^*_{w,\mathrm{sh}} &\subseteq \mathrm{conv}\left(w\mathbf{\Lambda}^*_{\mathrm{sh}} \cap \mathbf{\Lambda}_{\mathrm{port}} \cap \mathbb{Z}^m_+\right)\\ \mathbf{\Lambda}^*_{w,\mathrm{mh}} &\subseteq \mathrm{conv}\left(w\mathbf{\Lambda}^*_{\mathrm{mh}} \cap \mathbf{\Lambda}_{\mathrm{port}} \cap \mathbb{Z}^m_+\right).\end{aligned}$$

The above analysis of the multi-wavelength capacity region characterizes the throughput properties of the most and least capable joint electronic and optical systems. In particular, Theorem 5.4.1 provides a characterization of the capacity region of a network having both full wavelength conversion and multi-hop capability. The theorem also characterizes the capacity region of a network having no wavelength conversion capability and employing exclusively single-hop routing. Clearly, the capacity region of a network with single-hop and partial/full wavelength conversion capabilities and that of a network with multi-hop and partial/no wavelength conversion capability falls somewhere between the regions we have characterized.

Clearly, in the above characterization of the w-wavelength capacity region, when w becomes sufficiently large, we observe that the region $\mathbf{\Lambda}_{\mathrm{port}}$ becomes the binding element in each of the intersections $w\mathbf{\Lambda}^*_{\mathrm{sh}} \cap \mathbf{\Lambda}_{\mathrm{port}}$ and $w\mathbf{\Lambda}^*_{\mathrm{mh}} \cap \mathbf{\Lambda}_{\mathrm{port}}$. Thus, we can observe the natural transition from the single-wavelength capacity region $\mathbf{\Lambda}^*_1$ through to the wavelength-unconstrained capacity region $\mathbf{\Lambda}_{\mathrm{port}}$.

5.5 Conclusions

In this chapter, we considered several natural avenues of research stemming from our throughput study of single-wavelength reconfigurable optical networks of the previous chapter. We demonstrated that the multi-hop maximum all-to-all traffic supported by the network can be computed in polynomial time, through a relaxation of a multicommodity flow optimization. Interestingly, we found that the computation of the corresponding single-hop geometric property is in general an NP-hard problem.

Subsequently, we demonstrated how the results of Chapter 4 can be extended in a straightforward manner to multi-wavelength optical networks. Rather than connecting the multi-wavelength capacity region to the classical RWA problem, we defined an analogous quantity (χ) representing the minimum number of elements in a valid multi-wavelength decomposition of any given traffic matrix.

Finally, we set out to quantify the transition from the single-wavelength capacity region Λ_1^*, characterized in Chapter 4 to the wavelength-unconstrained region Λ_{port}. To that end, we studied the connection between the w-wavelength capacity region and the single-wavelength capacity region. We demonstrated that when the network is subject to a property called the round-up property, then we can provide an exact characterization of this transition.

5.5.1 Future directions

In the single-wavelength scenario, there were only two relevant network settings whose throughput properties we set out to quantify in Chapter 4: single-hop and multi-hop capable networks. When we consider multi-wavelength networks, there can be more grades of functionality, with each grade achieving its own throughput performance. At the extremes are: a network allowing only single-hop routes and employing no wavelength conversion; and a network allowing any multi-hop routes and employing full wavelength conversion. Simply alternating these combinations of conversion and hop capabilities introduces two intermediate grades of network functionality. Our study in this chapter can only capture the throughput gaps between the two extremes of network functionality. Thus, determining the throughput performance capabilities of all grades of network functionality remains an interesting problem of future study.

A related point concerns our discovery in Chapter 4 of a 33% performance gap between single-hop and multi-hop capable algorithms for a particular network (the bidirectional ring having $n \geq 7$, n even). An important question is: Can the throughput performance gap between different grades of network functionality be arbitrarily large, or is there a fundamental limit on this gap?

The round-up property may be an overly strict requirement in establishing an *exact* connection between the single- and multi-wavelength capacity regions. An interesting question is: Under what conditions, if at all, do equations (5.1)-(5.2) fail? We feel that it is probable that there are networks in which these equations do fail to hold. The most likely avenue for demonstrating this failure is to discover a network and traffic demand under which one of these equalities fails. This remains an open question.

Appendix

5.A Proof of Theorem 5.1.1

Note that $1/\theta^{\mathrm{mh}} = \liminf_{l\to\infty} W^{\mathrm{c}}(l\mathbf{J})/l$. Suppose $\bar{W}^{\mathrm{c}}(\mathbf{J}) > 1/\theta^{\mathrm{mh}}$. Then there exists an integer $l \geq 1$ such that $W^{\mathrm{c}}(l\mathbf{J})/l < \bar{W}^{\mathrm{c}}(\mathbf{J})$. Thus, there exists an RWA for the traffic demand $l\mathbf{J}$ requiring fewer than $l\bar{W}^{\mathrm{c}}(\mathbf{J})$ wavelengths. For such an RWA, let f^e_{ij} be the number of lightpaths originating at node i and terminating at node j, that traverse fiber e. Then, dividing each quantity f^e_{ij} by l, $(f^e_{ij}/l, i, j \in V, e \in E_P)$ is a multicommodity flow (MCF) that satisfies (4.13) and (4.14) with $W = W^{\mathrm{c}}(l\mathbf{J})/l < \bar{W}^{\mathrm{c}}(\mathbf{J})$. This contradicts $\bar{W}^{\mathrm{c}}(\mathbf{J})$ as the optimal value to the relaxed version of the ILP (4.12)-(4.15).

Conversely, suppose $\bar{W}^{\mathrm{c}}(\mathbf{J}) < 1/\theta^{\mathrm{mh}}$, with the MCF $(f^e_{ij}, i, j \in V, e \in E_P)$ achieving $\bar{W}(\mathbf{J})$ as the optimal cost of the relaxed version of the ILP (4.12)-(4.15). Thus for integer $k \geq 1$, the MCF $\mathbf{f}^k = (10^k f^e_{ij}, i, j \in V, e \in E_P)$ is feasible for the traffic demand $10^k\mathbf{J}$ in the relaxed ILP. For each source-destination pair i, j, each path p through the network from i to j carries some non-negative amount of the total 10^k units of flow satisfied by $\mathbf{f}^k$. Let this flow associated with path p for traffic from node i to node j be ρ^p_{ij}. Clearly $\sum_p \rho^p_{ij} = 10^k$. Consider next the truncated flow $\lfloor \rho^p_{ij} \rfloor$ for each path p. The floor operation implies $\rho^p_{ij} - \lfloor \rho^p_{ij} \rfloor < 1$. The integer flows described by $(\lfloor \rho^p_{ij} \rfloor)$ directly translates to a MCF $(\tilde{f}^e_{ij})$, according to $\tilde{f}^e_{ij} = \sum_{\{p:e\in p\}} \lfloor \rho^p_{ij} \rfloor$. Clearly $(\tilde{f}^e_{ij})$ must be a feasible integer flow in (4.13)-(4.15). Further, since no more than $m!$ distinct paths exist in the network, then $(\tilde{f}^e_{ij})$ must at least satisfy the demand matrix $(10^k - m!)\mathbf{J}$. We thus obtain

$$\begin{aligned} W^{\mathrm{c}}((10^k - m!)\mathbf{J}) &\leq \max_{e\in E_P} \sum_{ij} \tilde{f}^e_{ij} \\ &\leq \max_{e\in E_P} \sum_{ij} 10^k f^e_{ij} && (5.3) \\ &= 10^k \bar{W}(\mathbf{J}) && (5.4) \end{aligned}$$

Above, (5.3) follows because

$$\tilde{f}^e_{ij} = \sum_{\{p:e\in p\}} \lfloor \rho^p_{ij} \rfloor \leq \sum_{\{p:e\in p\}} \rho^p_{ij} = 10^k f^e_{ij}$$

Since $\bar{W}^{\mathrm{c}}(\mathbf{J}) < 1/\theta^{\mathrm{mh}}$, there must exist k^* sufficiently large such that for all $k \geq k^*$, $10^k > m!$ and $(10^k/(10^k - m!))\bar{W}^{\mathrm{c}}(\mathbf{J}) < 1/\theta^{\mathrm{mh}}$. By (5.4), this implies for $k \geq k^*$ that $\theta^{\mathrm{mh}} < (10^k - m!)/W^{\mathrm{c}}((10^k - m!)\mathbf{J})$, which is a contradiction.

5.B Proof of Theorem 5.1.2

We begin with several definitions, and a useful theorem.

Definition 5.B.1 (Chromatic number [137]) *A k-coloring of a graph G is an assignment of one of k colors to each vertex so that adjacent vertices receive different colors. The* *chromatic number of G, denoted $C(G)$, is the least k for which G has a k-coloring.*

Definition 5.B.2 (Fractional chromatic number [137]) *A b-fold coloring of a graph G assigns to each vertex of G a set of b colors so that adjacent vertices receive disjoint sets of colors. We say that G is $a{:}b$-colorable if it has a b-fold coloring in which the colors are drawn from a palette of a colors. The least a for which G is $a{:}b$-colorable is denoted $C_b(G)$. The fractional chromatic number is defined as $C_f(G) = \liminf_{b\to\infty} C_b(G)/b$.*

Theorem 5.B.1 ([137, Thm. 3.9.2]) *For every real number $r > 2$, the problem of determining whether a graph G has $C_f(G) \leq r$ is NP-complete.*

To prove that determining θ^{sh} is NP-hard, we will show that determining whether a graph G has $C_f(G) \leq r$ is polynomial-time reducible to the problem of determining $\theta^{\mathrm{sh}} \leq 1/r$.

Consider a graph G. From G, we will build a physical topology graph G_P. Associate with each vertex of G a lightpath having fixed routing (there is no other available lightpath for that source-destination pair through the network). In this construction, no two lightpaths share any nodes in common. Each edge (v_1, v_2) of G implies that the lightpaths represented by vertices v_1, v_2 share at least one fiber link. We assume that the lightpaths associated with v_1 and v_2 only share a single fiber, and that no other lightpaths traverse the same fiber. Proceeding in this manner for all edges of G, we obtain a set of lightpaths whose incidence with one another is represented by G. Let V be the set of nodes terminating the lightpaths we have constructed thus far. For each directed pair of nodes $v_1, v_2 \in V$, if we have not yet constructed a lightpath from v_1 to v_2, define a new fiber link from v_1 to v_2, and let the lightpath from v_1 to v_2 traverse the new fiber link. We have thus determined a physical topology G_P and fixed lightpath routing associated with each possible source-destination pair. This is clearly a polynomial-time operation.

It remains to show that $C_f(G) \leq r$ if and only if $\theta^{\mathrm{sh}} \geq 1/r$. Consider the following set of equality statements.

$$\begin{aligned}
1/\theta^{\mathrm{sh}} &= \liminf_{l\to\infty} W^{\mathrm{nc}}(l\mathbf{J})/l && (5.5)\\
&= \liminf_{l\to\infty} C_l(G)/l && (5.6)\\
&= C_f(G) && (5.7)
\end{aligned}$$

Above, (5.5) follows from (5.6) because $W^{\mathrm{nc}}(l\mathbf{J}) = C_l(G)$. To see why this is so, note first that by definition, $C_l(G) \geq l$, and that any lightpath not associated with a node in G has no overlap with other lightpaths, and thus requires exactly l wavelengths to satisfy l units of demand. By definition, $W^{\mathrm{nc}}(l\mathbf{J})$ is the minimum number of wavelengths to route l lightpaths between each source-destination pair in the network. Since each lightpath is forced (by definition) to have fixed routing, we are assigning to each lightpath a total of l colors, so that no two lightpaths that overlap share any colors in common. This is clearly equal to $C_l(G)$, since any lightpath that overlaps with another lightpath is represented as an edge in G. The equality (5.7) implies that $C_f(G) \leq r$ if and only if $\theta^{\mathrm{sh}} \geq 1/r$, as desired.

5.C Proof of Theorem 5.3.1

Denote $\alpha^* = \limsup_{k\to\infty} k/\mathcal{W}_{\mathcal{P}}(k)$.

Proof that $\alpha_{\mathcal{P}} \geq \limsup_{k\to\infty} k/\mathcal{W}_{\mathcal{P}}(k)$ *:* Suppose $\boldsymbol{\lambda} \in \alpha^*\mathcal{P}$, with $\boldsymbol{\lambda} \neq 0$ (since $\boldsymbol{\lambda} = 0$ has a trivial RWA decomposition). Define the sequence of integer traffic matrices $\{\mathbf{T}^k\}$, such that for $i \neq j$, $T_{ij}^k = (\lfloor \lambda_{ij}\mathcal{W}_{\mathcal{P}}(k) - \eta_k \rfloor)^+$. We seek to define the sequence $\{\eta_k\}$ to ensure that $\mathbf{T}^k \in \mathcal{K}_k^{\mathcal{P}}$, $\forall k$. To this end, it is straightforward to demonstrate that

$$\begin{aligned} \lambda_{ij}\mathcal{W}_{\mathcal{P}}(k) &= \frac{\lambda_{ij}}{\alpha^*}\alpha^*\mathcal{W}_{\mathcal{P}}(k) \\ &\leq \frac{\lambda_{ij}}{\alpha^*}\left(k + \varepsilon_k \mathcal{W}_{\mathcal{P}}(k)\right) \end{aligned} \tag{5.8}$$

where for $k \in \mathbb{Z}_+$,

$$\varepsilon_k = \sup_{\tilde{k}\geq k}\left|\frac{\tilde{k}}{\mathcal{W}_{\mathcal{P}}(\tilde{k})} - \alpha^*\right|.$$

To ensure $\mathbf{T}^k \in \mathcal{K}_k^{\mathcal{P}}$, we simply choose

$$\eta_k = \left(\max_{ij}\lambda_{ij}\right)\frac{\varepsilon_k\mathcal{W}_{\mathcal{P}}(k)}{\alpha^*}.$$

To see why this is so, note that this choice of η_k ensures $T_{ij}^k \leq \lambda_{ij}k/\alpha^*$ for all i, j, as follows:

$$\begin{aligned} T_{ij}^k &= (\lfloor \lambda_{ij}\mathcal{W}_{\mathcal{P}}(k) - \eta_k \rfloor)^+ \\ &\leq (\lambda_{ij}\mathcal{W}_{\mathcal{P}}(k) - \eta_k)^+ \\ &\leq \left(\frac{\lambda_{ij}}{\alpha^*}(k - \varepsilon_k\mathcal{W}_{\mathcal{P}}(k)) - \eta_k\right)^+ && (5.9) \\ &= \left(\frac{\lambda_{ij}}{\alpha^*}(k - \varepsilon_k\mathcal{W}_{\mathcal{P}}(k)) - \left(\max_{ij}\lambda_{ij}\right)\frac{\varepsilon_k\mathcal{W}_{\mathcal{P}}(k)}{\alpha^*}\right)^+ \\ &\leq \left(\frac{\lambda_{ij}}{\alpha^*}k\right)^+ \\ &= \frac{\lambda_{ij}}{\alpha^*}k && (5.10) \end{aligned}$$

Above, (5.9) follows from (5.8). Our assumption of $\boldsymbol{\lambda} \in \alpha^*\mathcal{P}$ implies that $(k/\alpha^*)\boldsymbol{\lambda} \in k\mathcal{P}$. By (5.10), we have that $(k/\alpha^*)\boldsymbol{\lambda}$ dominates $\mathbf{T}^k$, entry-by-entry, from which we can conclude that $\mathbf{T}^k \in k\mathcal{P}$. Finally, since $\mathbf{T}^k$ is an integer matrix, it must belong to $\mathcal{K}_k^{\mathcal{P}}$.

Clearly, $\eta_k/\mathcal{W}_{\mathcal{P}}(k) \to 0$ as $k \to \infty$, since the limsup definition of α^* implies that $\varepsilon_k \to 0$ as $k \to \infty$. Next, define $\boldsymbol{\lambda}^k = (1/\mathcal{W}_{\mathcal{P}}(k))\mathbf{T}^k$. Since $\mathbf{T}^k \in \mathcal{K}_k^{\mathcal{P}}$, it must be true that $\boldsymbol{\lambda}^k \in \mathcal{R}$. To demonstrate that $\boldsymbol{\lambda}$ has a RWA decomposition, we need to show that $\boldsymbol{\lambda}^k \to \boldsymbol{\lambda}$ as $k \to \infty$. Since $\eta_k/\mathcal{W}_{\mathcal{P}}(k) \to 0$ as $k \to \infty$, this is clearly true. Thus, $\boldsymbol{\lambda} \in \mathrm{cl}(\mathcal{R})$, which implies by Theorem 5.2.1 that $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*$. Since this holds for all $\boldsymbol{\lambda} \in \alpha^*\mathcal{P}$, it must be true that $\alpha_{\mathcal{P}} \geq \alpha^*$.

Proof that $\alpha_{\mathcal{P}} \leq \limsup_{k\to\infty} k/\mathcal{W}_{\mathcal{P}}(k)$ *:* Suppose there exists $\alpha > \alpha^*$ such that $\alpha\mathcal{P} \subseteq \boldsymbol{\Lambda}^*$. Denote by $\alpha\mathcal{P} \setminus \alpha^*\mathcal{P}$ the portion of region $\alpha\mathcal{P}$ that is disjoint from $\alpha^*\mathcal{P}$. Since $\mathcal{P}$ is Pareto, $\alpha\mathcal{P} \subset \alpha^*\mathcal{P}$. Consider any positive integer u. By Theorem 5.2.1, and since $\mathcal{P}$ is convex, compact, and full-dimensional, there must exist a non-negative integer K_u, non-negative integer matrices $\mathbf{T}^{u,1}, \ldots, \mathbf{T}^{u,K_u}$, and integers $W^{u,1}, \ldots, W^{u,K_u}$, such that $W^{u,l} \geq \chi(\mathbf{T}^{u,l})$ for all l, $(1/W^{u,l})\mathbf{T}^{u,l} \in \alpha_1\mathcal{P} \setminus \alpha_2\mathcal{P}$ for all l, where $\alpha > \alpha_1 > \alpha_2 > \alpha^*$, and

$$\alpha\mathcal{P} \setminus \alpha^*\mathcal{P} \subseteq \bigcup_{l=1}^{K_u} \left\{\boldsymbol{\lambda} : \left|\lambda_{ij} - T_{ij}^{u,l}/W^{u,l}\right| \leq 1/u, \forall i,j \in V\right\}. \tag{5.11}$$

In words, the set of points $\{(1/W^{u,1})\mathbf{T}^{u,1}, \ldots, (1/W^{u,K_u})\mathbf{T}^{u,K_u}\}$ are the center locations of a set of $(1/u)$-balls that cover the region $\alpha\mathcal{P} \setminus \alpha^*\mathcal{P}$. The compactness of $\mathcal{P}$ is sufficient to ensure the existence of a covering such that K_u is finite-valued [93]. For $l \in \{1, \ldots, K_u\}$, let $\kappa_l^u = \prod_{\tilde{l} \neq l} W^{u,l}$. By definition we then have for any integer $r \geq 0$,

$$10^r \kappa_l^u \mathbf{T}^{u,l} \in \left(10^r \textstyle\prod_{\tilde{l}} W^{u,\tilde{l}}\right)(\alpha_1\mathcal{P} \setminus \alpha_2\mathcal{P}) \subseteq \left\lceil 10^r \alpha_1 \textstyle\prod_{\tilde{l}} W^{u,\tilde{l}} \right\rceil \mathcal{P} \setminus \left\lfloor 10^r \alpha_2 \textstyle\prod_{\tilde{l}} W^{u,\tilde{l}} \right\rfloor \mathcal{P}$$

Since $\alpha > \alpha_1 > \alpha_2 > \alpha^*$, there must exist an integer $r_u^* \geq 0$ such that for $r \geq r_u^*$,

$$\left\lceil 10^r \alpha_1 \textstyle\prod_{\tilde{l}} W^{u,\tilde{l}} \right\rceil < \alpha 10^r \textstyle\prod_{\tilde{l}} W^{u,\tilde{l}}, \tag{5.12}$$

$$\left\lfloor 10^r \alpha_2 \textstyle\prod_{\tilde{l}} W^{u,\tilde{l}} \right\rfloor > \alpha^* 10^r \textstyle\prod_{\tilde{l}} W^{u,\tilde{l}}, \tag{5.13}$$

and such that any matrix on the Pareto boundary of $\mathcal{K}^{\mathcal{P}}_{\left\lceil 10^r \alpha_1 \prod_{\tilde{l}} W^{u,\tilde{l}} \right\rceil}$ resides in the region $\left\lceil 10^r \alpha_1 \prod_{\tilde{l}} W^{u,\tilde{l}} \right\rceil \mathcal{P} \setminus \left\lfloor 10^r \alpha_2 \prod_{\tilde{l}} W^{u,\tilde{l}} \right\rfloor \mathcal{P}$. Denote $k_u^* = \lceil 10^{r_u^*} \alpha_1 \prod_{\tilde{l}} W^{u,l} \rceil$. Finally, observe that for any integer $r > 0$, $\chi(10^r \kappa_l^u \mathbf{T}^{u,l}) \leq 10^r \kappa_l^u W^{u,l}$, since the decomposition of $\mathbf{T}^{u,l}$ can be repeated κ_l^u times.

Consider any traffic $\tilde{\mathbf{T}} \in \mathcal{K}^{\mathcal{P}}_{k_u^*}$. Let $\mathbf{T}$ be an integer matrix that dominates $\tilde{\mathbf{T}}$, entry-by-entry, and that resides on the Pareto boundary of $\mathcal{K}^{\mathcal{P}}_{k_u^*}$. Recall that we have selected k_u^* such that $\mathbf{T} \in \left\lceil 10^{r_u^*} \alpha_1 \prod_{\tilde{l}} W^{u,\tilde{l}} \right\rceil \mathcal{P} \setminus \left\lfloor 10^{r_u^*} \alpha_2 \prod_{\tilde{l}} W^{u,\tilde{l}} \right\rfloor \mathcal{P}$. Applying (5.12)-(5.13), we must have that

$$\frac{1}{10^{r_u^*} \prod_{\tilde{l}} W^{u,\tilde{l}}} \mathbf{T} \in \alpha\mathcal{P} \setminus \alpha^*\mathcal{P}.$$

Applying (5.11), there must exist an integer l^*, where $1 \leq l^* \leq K_u$, such that

$$\left| \frac{T_{ij}}{10^{r_u^*} \prod_{\tilde{l}} W^{u,\tilde{l}}} - \frac{T_{ij}^{u,l^*}}{W^{u,l^*}} \right| \leq \frac{1}{u}, \quad \forall i,j.$$

Multiplying through by $10^{r_u^*} \prod_{\tilde{l}} W^{u,\tilde{l}}$, we obtain

$$\left| T_{ij} - 10^{r_u^*} \kappa_{l^*}^u T_{ij}^{u,l^*} \right| \leq \frac{10^{r_u^*} \prod_{\tilde{l}} W^{u,\tilde{l}}}{u}.$$

Summing over all indices i, j, we obtain

$$\sum_{ij} \left| T_{ij} - 10^{r_u^*} \kappa_{l^*}^u T_{ij}^{u,l^*} \right| \leq \frac{n(n-1)10^{r_u^*} \prod_{\bar{l}} W^{u,\bar{l}}}{u}.$$

The above serves as an upper bound on the number of connection requests in $\mathbf{T}$ in excess of those in the traffic $10^{r_u^*} \kappa_{l^*}^u \mathbf{T}^{u,l^*}$. Thus, at worst each such excess request requires a single service matrix in a valid RWA decomposition of $\mathbf{T}$. Since $\mathbf{T}$ dominates $\tilde{\mathbf{T}}$, entry-by-entry, we can now upper bound $\chi(\tilde{\mathbf{T}})$ as follows

$$\begin{aligned} \chi(\tilde{\mathbf{T}}) \leq \chi(\mathbf{T}) &\leq \chi(\mathbf{T}^{u,l^*}) + n(n-1)10^{r_u^*} \textstyle\prod_{\bar{l}} W^{u,\bar{l}}/u, \\ &\leq 10^{r_u^*} \kappa_{l^*}^u W^{u,l^*} + n(n-1)10^{r_u^*} \textstyle\prod_{\bar{l}} W^{u,\bar{l}}/u, \\ &= 10^{r_u^*} \left(\textstyle\prod_{\bar{l}} W^{u,\bar{l}} \right) + n(n-1)10^{r_u^*} \left(\textstyle\prod_{\bar{l}} W^{u,\bar{l}} \right) /u \end{aligned}$$

It immediately follows that

$$\mathcal{W}_{\mathcal{P}}(k_u^*) \leq 10^{r_u^*} \left(\textstyle\prod_{\bar{l}} W^{u,\bar{l}} \right) + n(n-1)10^{r_u^*} \left(\textstyle\prod_{\bar{l}} W^{u,\bar{l}} \right) /u.$$

Consequently,

$$\begin{aligned} \frac{k_u^*}{\mathcal{W}_{\mathcal{P}}(k_u^*)} &\geq \frac{\left\lceil 10^{r_u^*} \alpha_1 \prod_{\bar{l}} W^{u,\bar{l}} \right\rceil}{10^{r_u^*} \left(\prod_{\bar{l}} W^{u,\bar{l}} \right) + n(n-1)10^{r_u^*} \left(\prod_{\bar{l}} W^{u,\bar{l}} \right) /u} \\ &\geq \frac{\alpha_1}{1 + n(n-1)/u} \\ &> \alpha^* \end{aligned}$$

where the final strict inequality holds for all u sufficiently large. It remains to show that $k_u^* \to \infty$ as $u \to \infty$. Clearly, $K_u \to \infty$ as $u \to \infty$. Note that $k_u^* \geq \alpha_1 \prod_{\bar{l}} W^{u,\bar{l}}$. Thus, it is sufficient to demonstrate that

$$\max_{1 \leq l \leq K_u} W^{u,l} \to \infty \text{ as } u \to \infty.$$

We prove this assertion by contradiction. Suppose there exists an integer W such that $W^{u,l} \leq W$ for all l, u. Then, since K_u tends to infinity with u, the set $\{\mathbf{T} : \chi(\mathbf{T}) \leq W\}$ must have infinite cardinality. To demonstrate that this is false, note that the set $\mathcal{S}$ is a finite set of integer matrices, which implies that $\max_{\mathbf{S} \in \mathcal{S}} \max_{i,j} d_{ij}(\mathbf{S}) < \infty$. Then clearly, by the definition of χ, we must have

$$\{\mathbf{T} \in \mathbb{Z}_+^{n \times n} : \chi(\mathbf{T}) \leq W\} \subseteq \{\mathbf{T} \in \mathbb{Z}_+^{n \times n} : T_{ij} \leq W \max_{\mathbf{S} \in \mathcal{S}} \max_{i,j} d_{ij}(\mathbf{S})\}.$$

Above, the set on the right clearly has finite cardinality, which provides the contradiction.

5.D Proof of Theorem 5.4.1

Our proof follows for both the single-hop and multi-hop versions of the Theorem. Consequently, we will not distinguish between these cases, except to clarify the manner in which RWA decompositions are obtained. Additionally, the function $W(\cdot)$ will simultaneously represent either $W^{\text{nc}}(\cdot)$ or $W^{\text{c}}(\cdot)$.

We begin by demonstrating that

$$\mathbf{\Lambda}_w^* \subseteq \text{conv}\left(w\mathbf{\Lambda}_1^* \cap \mathbf{\Lambda}_{\text{port}} \cap \mathbb{Z}_+^m\right). \tag{5.14}$$

Consider any corner point $\boldsymbol{\lambda}$ of the region $\mathbf{\Lambda}_w^*$. By the definition of the capacity region $\mathbf{\Lambda}_w^*$, $\boldsymbol{\lambda}$ can be expressed as a convex combination of w-wavelength service matrices. Namely, there exists an integer L, non-negative coefficients α_l, and service activation matrices $\mathbf{S}^l \in \mathcal{S}_w$, for $l = 1, \ldots, L$, such that $\sum_l \alpha_l = 1$ and $\boldsymbol{\lambda} = \sum_l \alpha_l \mathbf{d}(\mathbf{S}^l)$. Here, $\mathcal{S}_w$ is the set of available w-wavelength service activation matrices. In the service activation $\mathbf{S}^l$, if we consider each wavelength as a separate *single-wavelength service activation*, and associate a rate of $1/w$ with each wavelength, then we obtain a valid single-wavelength RWA decomposition for the matrix $(1/w)\mathbf{d}(\mathbf{S}^l)$. In other words, we have that $(1/w)\mathbf{d}(\mathbf{S}^l) \in \mathbf{\Lambda}_1^*$. To be clear, observe that in order to obtain single-hop single-wavelength service activations at each wavelength in this decomposition, each service matrix $\mathbf{S} \in \mathcal{S}_w$ must be a single-hop service matrix with no wavelength conversion. For the case of multi-hop single-wavelength service activations, the set of service matrices $\mathcal{S}_w$ can include multi-hop activations, as well as activations employing wavelength conversion. By the convexity of $\mathbf{\Lambda}_1^*$ as well as of $\mathbf{\Lambda}_w^*$, we must then have

$$\{(1/w)\boldsymbol{\lambda} : \boldsymbol{\lambda} \in \mathbf{\Lambda}_w^*\} \subseteq \mathbf{\Lambda}_1^*,$$

which implies

$$\mathbf{\Lambda}_w^* \subseteq w\mathbf{\Lambda}_1^*. \tag{5.15}$$

Additionally, since $\mathbf{\Lambda}_{\text{port}}$ is the convex hull of all wavelength-unconstrained service configurations, $\mathbf{\Lambda}_{\text{port}}$ must contain the w-wavelength service configurations. This implies

$$\mathbf{\Lambda}_w^* \subseteq \mathbf{\Lambda}_{\text{port}}. \tag{5.16}$$

Together, (5.15) and (5.16) provide that $\mathbf{\Lambda}_w^* \subseteq w\mathbf{\Lambda}_1^* \cap \mathbf{\Lambda}_{\text{port}}$. Since $\mathbf{\Lambda}_w^*$ must have integer corner points, these integer configurations must lie in the set $w\mathbf{\Lambda}_1^* \cap \mathbf{\Lambda}_{\text{port}}$. Thus, intersecting the set $w\mathbf{\Lambda}_1^* \cap \mathbf{\Lambda}_{\text{port}}$ with the integer lattice $\mathbb{Z}^m$ necessarily captures the corners of $\mathbf{\Lambda}_w$, from which (5.14) follows by the convexity of $\mathbf{\Lambda}_w^*$.

Note that the above proof does not assume the round-up property, from which we conclude that (5.14) must be true in general.

Next, assuming the round-up property holds, we prove that

$$\text{conv}\left(w\mathbf{\Lambda}_1^* \cap \mathbf{\Lambda}_{\text{port}} \cap \mathbb{Z}_+^m\right) \subseteq \mathbf{\Lambda}_w^*. \tag{5.17}$$

Our proof will demonstrate that no integer point can belong to $w\mathbf{\Lambda}_1^*$ without having a corresponding w-wavelength service configuration. Consider any integer matrix $\mathbf{T} \in \mathbb{Z}_+^m$.

By the round-up property, there exists $\varepsilon \in (0,1)$ such that

$$\sup_{k\in\mathbb{Z}_+} \frac{k}{W(k\mathbf{T})}\mathbf{T} = \frac{1}{\inf_{k\in\mathbb{Z}_+} \frac{W(k\mathbf{T})}{k}}\mathbf{T} \leq \frac{1}{W(\mathbf{T})-1+\varepsilon}\mathbf{T}. \tag{5.18}$$

By Theorem 5.3.1, (5.18) implies that

$$\sup\{\alpha : \alpha\mathrm{T} \subseteq \mathbf{\Lambda}_1^*\} \leq \frac{1}{W(\mathbf{T})-1+\varepsilon}. \tag{5.19}$$

Now, we seek the minimum integer scaling w^* such that $\mathbf{T}$ belongs to $w^*\mathbf{\Lambda}_1^*$,

$$w^* = \min_{w\in\mathbb{Z}_+,\, \mathbf{T}\in w\mathbf{\Lambda}_1^*} w.$$

Suppose $w^* \leq W(\mathbf{T})-1$. Then since $\mathbf{T} \in w^*\mathbf{\Lambda}_1^*$, we obtain

$$\frac{1}{W(\mathbf{T})-1}\mathbf{T} \in \mathbf{\Lambda}_1^*,$$

which contradicts (5.19). We conclude that $w^* = W(\mathbf{T})$. Thus, for any integer traffic $\mathbf{T}$, $\mathbf{T}$ is an element of $w\mathbf{\Lambda}_1^*$ only when $\mathbf{T}$ requires w or fewer wavelengths. Thus, every integer point in $w\mathbf{\Lambda}_1^* \cap \mathbf{\Lambda}_{\text{port}}$ has a valid service configuration requiring at most w wavelengths, implying (5.17) as desired.

Chapter 6

Greedy weighted matching for scheduling the input-queued switch

In this chapter, we study the throughput properties of a network control algorithm that is computationally less complex than that of Tassiulas and Ephremides. In the optical network setting of the previous two chapters, the computational and communication complexity associated with control information dissemination was regarded as a small overhead. As the number of network nodes increases however, this overhead naturally grows. While this overhead is small with respect to the time scale of reconfiguration decisions in the network, it is desirable to employ efficient algorithms to keep the queue information upon with reconfigurations are based as recent as possible. In this chapter, we consider the simple *bipartite* network graph structure, which is a typical model for input-queued switches, and study the throughput properties of a *maximal* weight scheduling algorithm.

6.1 Overview and summary of contributions

We consider the scheduling problem for the $n \times n$ input-queued switch. It is widely held that the $O(n^3)$ computational complexity of *maximum weight* matching is overly burdensome for implementation on a slot-by-slot basis in practical systems operating at high rates [80]. Many practitioners resort to suboptimal matching algorithms in conjunction with speedup to provide optimal throughput performance. In this paper, we consider greedy *maximal weight* matching as a suboptimal matching algorithm, and we make no use of speedup. We conduct numerical and analytical studies to demonstrate the attractive throughput and delay performance properties of greedy matching based scheduling.

The switch scheduling literature often takes advantage of the simple fact that a greedy weighted matching on a weighted bipartite graph provides a 2-approximation to the weight of the maximum weight matching, and thus that at least 50% throughput is achievable (see e.g. [73]). Consequently, it is simple to demonstrate that 100% throughput is achievable under greedy matching in conjunction with a speedup of two. Less can be found on the topic of greedy matchings with no speedup [82].

In this chapter, we pursue two important goals:

1. To develop numerical simulations that attest to the attractive throughput and delay properties of greedy weighted matching based schedulers; and

2. To prove the throughput optimality of greedy weighted matching based scheduling in the 2×2 input-queued switch.

6.2 Greedy maximal weight matching

A greedy maximal weighted matching on a complete weighted bipartite graph (S, T, E) with $|S| = |T| = n$ and weight function $w : E \to \mathbb{R}_+$ selects sequentially the maximum weighted edge in (S, T, E) while maintaining a matching. For edge $e \in E$ let $\sigma(e) \in S$ and $\tau(e) \in T$ denote the S and T vertices corresponding to edge e, respectively. Thus, a greedy maximal weighted matching $M \subset E$ under weight function w is given as follows. This particular algorithm dates back at least to Reingold and Tarjan [123], where it was studied in complete weighted graphs. In [73, 74], the algorithm is considered in complete weighted bipartite graphs, and is referred to as the CQ algorithm. Since the bipartite graph (S, T, E) is complete, the algorithm clearly terminates.

Algorithm 10 Greedy maximal weight matching algorithm

```
Start with an empty matching, $M = \{\}$
repeat
    Select $e^* \in \arg\max_{\{e \in E : \sigma(e) \neq \sigma(\bar{e}), \tau(e) \neq \tau(\bar{e}) \forall \bar{e} \in M\}} w(e)$
    $M \leftarrow M \cup e^*$
until $|M| = n$
return $M$
```

6.2.1 Network model and scheduling algorithm

We consider an input-queued switch employing virtual output queues (VOQs) for each source-destination pair. Each queue contains fixed-size cells awaiting transmission to a particular output port of the switch. We consider slotted time, with index t, and assume that one time slot is required for transmission of any cell across the switch fabric. For $t \geq 0$, we define by $Q(t)$ the queue occupancy matrix at time t, with $Q_{ij}(t)$ equal to the number of cells in the queue at input port i destined to output port j at time t. The cumulative arrival process is defined as $\mathbf{A}(t)$, with $A_{ij}(t)$ equal to the total number of cell arrivals to input port i for destination port j by time slot t. We assume that VOQ$_{ij}$ has arrivals at rate λ_{ij} for all i, j. The rate matrix $\boldsymbol{\lambda}$ gathers each of these rates together. The set of admissible arrival rate matrices $\boldsymbol{\Lambda}^*$ is the doubly substochastic region:

$$\boldsymbol{\Lambda}^* = \left\{ \boldsymbol{\lambda} \geq 0 : \sum_j \lambda_{ij} \leq 1 \forall i, \quad \sum_i \lambda_{ij} \leq 1 \forall j \right\}.$$

The greedy maximal weighted matching based scheduler is as follows. The algorithm is equivalent to that employed in [73, 74].

Algorithm 11 Greedy maximal weight matching scheduler for the $n \times n$ input-queued switch

```
for each time $t \geq 0$ do
    Obtain the complete weighted $n \times n$ bipartite graph $(S, T, E)$ with edge weight $w(e) = Q_{\sigma(e)\tau(e)}(t)$ for $e \in E$
    Obtain a maximal weight matching $M$ using Algorithm 10
    Configure the switch according to $M$
end for
```

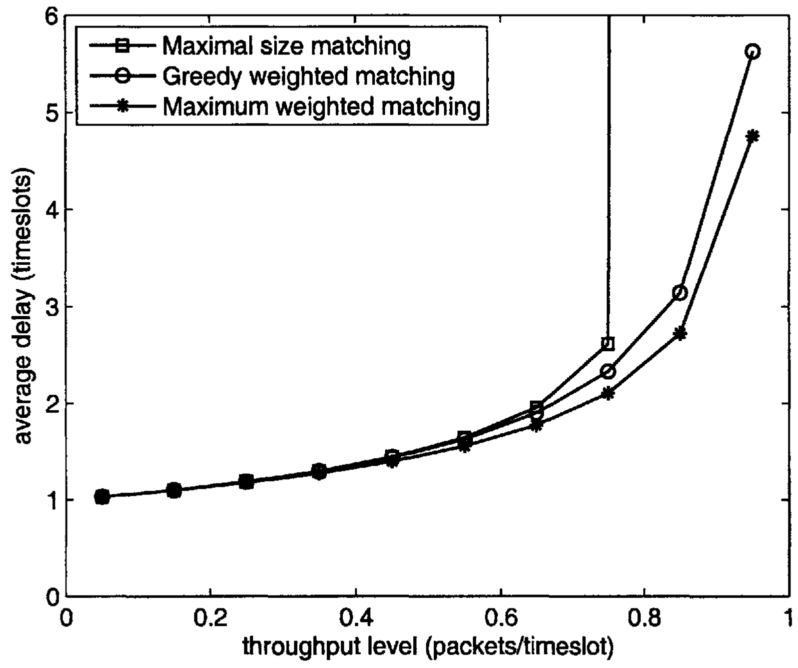
6.3 Numerical study

Here we report the results of our numerical simulations of greedy weighted scheduling. In addition to demonstrating the attractive throughput properties of the scheduler, we also observe delay performance quite similar to that achievable under maximum weight matching based scheduling.

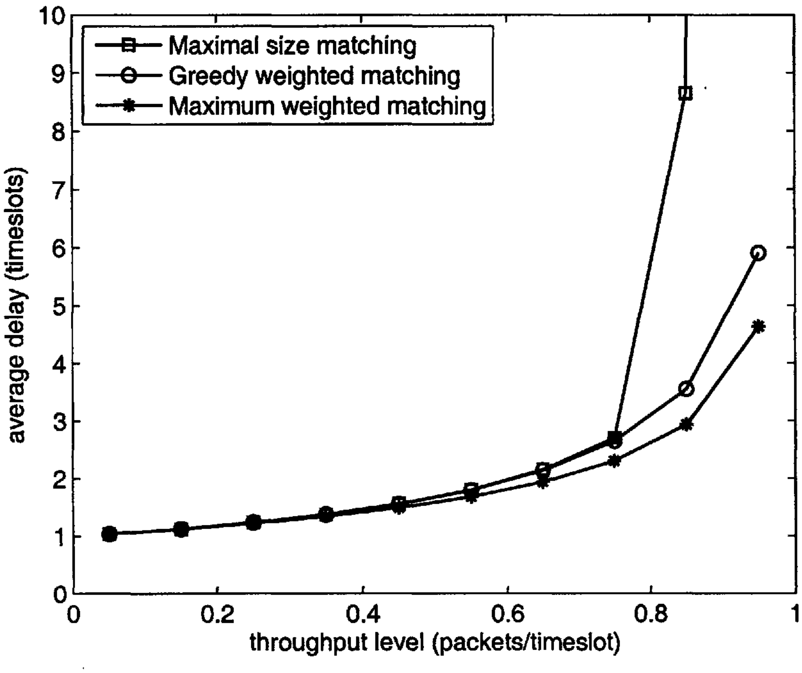
Our simulation scenario considers $n = 6, 16$, and a range of throughput levels for simulation. Each throughput level is given by the maximum row/column sum of the arrival rate matrix. At each throughput level, 50 arrival rate matrices are generated randomly and for each rate matrix, a sample path is simulated over 2.5×10^5 time slots, starting at initial VOQ occupancies of zero. The average queueing delay over each sample path is averaged over the 50 sample paths to generate an individual data point representing the average delay at that throughput level. We present the simulation results in Figure 6-1. The figures present average delay performance over a range of throughput levels for three scheduling algorithms: maximum weight matching, maximal (greedy) weighted matching, and maximal (size) matching. Briefly, a maximal size matching based scheduler greedily selects edges for which *any* nonzero queue backlog awaits service. Thus, maximal size matching can be likened to maximal weighted matching, where each edge weight is equal to the corresponding VOQ backlog, taken to the power zero.

Note above that maximum weighted matching and the greedy algorithm maintain a close level of delay over the entire range of throughput considered. Additionally, we observe that greedy scheduling never suffers instability over the range of throughput levels. This points to significantly improved throughput performance over the 50% level that can be trivially shown to be sufficient (though clearly not necessary) under any 2-approximation algorithm to maximum weighted matching. The maximal size matching algorithm in both figures shows a throughput loss somewhere in the range of the 0.75 to 0.85 throughput level. These simulations attest that when there is no speedup, maximal size matching demonstrates an observable throughput loss, while maximal weight matching does not.

Given these attractive delay and throughput performance properties of the simulated greedy weighted matching based scheduler, we next analytically pursue the maximum throughput properties of the switch under the greedy algorithm. We begin by considering the 2×2 switch.



(a) $n = 6$



(b) $n = 16$

Figure 6-1: Average delay performance over a range of throughput levels for maximal size matching, greedy weight matching, and maximum weight matching based scheduling.

6.4 Greedy matching achieves 100% throughput in the 2×2 input-queued switch

For the 2×2 switch, there are only two configurations that can be selected as greedy matchings. In matrix form, they are given by

$$\pi_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \pi_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

The main result of this section is provided next. It states that the greedy maximal weighted matching based scheduler stabilizes any arrival process with rate matrix belonging to the admissible region Λ^*.

Theorem 6.4.1 *For the* 2×2 *input-queued switch, greedy maximal weighted matching based scheduling achieves* 100% *throughput.*

Proof: The proof based on fluid limits is straightforward, and can be found in Appendix 6.A. A stronger stability argument for the case of Bernoulli arrivals is made in Appendix 6.B. ■

It is interesting to note that although the different proofs of Theorem 6.4.1 follow a similar approach, the rate stability argument based on fluid limits is significantly less cumbersome.

6.5 Beyond the 2×2 switch

In this section, we seek to understand the throughput properties of $n \times n$ switches, where $n \geq 3$, and of $n \times m$ switches.

6.5.1 The 3×3 switch

In [51], it was shown that the 6-ring (a cycle consisting of 6 edges) can potentially suffer from throughput loss under certain traffic processes, namely a deterministic fluid process with uniform load at each edge.[1] Here, we find that this 6-ring structure makes an appearance in our analysis of the $n \times n$ switch, for $n \geq 3$. In particular, consider the marked entries in the matrix in Figure 6-2(a), where entry (i, j) represents VOQ_{ij} of the switch. Figure 6-2(b) presents the network graph of edges that must be employed to service these marked VOQ's. Clearly, two VOQ's share a vertex if they have and input or output port in common. Note that the edges of the graph in Figure 6-2(b) form the graph C_6, the 6-ring.

The following theorem establishes that the 3×3 switch can only lose throughput on a low-dimensional set of arrival rates. In other words, we can assert that maximal matching based schedulers achieve the network capacity region Λ^*, up to a set of Lebesgue measure zero.

[1] We will study the work of [51] at length in Chapters 7 and 8.

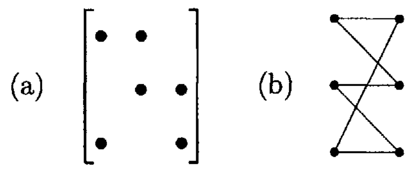


Figure 6-2: The appearance of a 6-ring in the $n \times n$ switch, $n \geq 3$. (a) Marked entries indicate VOQ's of interest. (b) The network graph of edges employed in servicing these VOQ's is a 6-ring.

Theorem 6.5.1 *The* 3×3 *switch is rate stable under greedy maximal weighted matching for all arrival processes having rates belonging to the region* $\mathbf{\Lambda}^* \setminus \mathbf{\Lambda}_3$, *where*

$$\begin{aligned}\mathbf{\Lambda}_3 = \mathbf{\Lambda}^* \cap \big(&\{\boldsymbol{\lambda} : \lambda_{11} - \lambda_{23} = \lambda_{22} - \lambda_{31} = \lambda_{33} - \lambda_{12}\} \cup \\ &\{\boldsymbol{\lambda} : \lambda_{11} - \lambda_{22} = \lambda_{23} - \lambda_{31} = \lambda_{32} - \lambda_{13}\} \cup \\ &\{\boldsymbol{\lambda} : \lambda_{12} - \lambda_{21} = \lambda_{23} - \lambda_{32} = \lambda_{31} - \lambda_{13}\} \cup \\ &\{\boldsymbol{\lambda} : \lambda_{12} - \lambda_{23} = \lambda_{21} - \lambda_{32} = \lambda_{33} - \lambda_{11}\} \cup \\ &\{\boldsymbol{\lambda} : \lambda_{13} - \lambda_{22} = \lambda_{21} - \lambda_{33} = \lambda_{32} - \lambda_{11}\} \cup \\ &\{\boldsymbol{\lambda} : \lambda_{13} - \lambda_{21} = \lambda_{22} - \lambda_{33} = \lambda_{31} - \lambda_{12}\}\big)\end{aligned}$$

The set $\mathbf{\Lambda}_3$ *has Lebesgue measure zero in* $\mathbb{R}_+^m$.

Proof: See Appendix 6.C. ■

6.5.2 Larger switches

Our result concerning the 3×3 switch is striking, and immediately calls into question the case of the $n \times n$ switch, for $n \geq 4$. In particular, we wish to explore whether for $n \geq 4$ the measure zero property continues to hold for the set of arrival rates that cannot be guaranteed stabilizable. Here we demonstrate that the $n_1 \times n_2$ input-queued switch, where $n_1, n_2 \geq 4$, potentially suffers from throughput loss over a *non-negligible* portion of the switch capacity region.

Consider the marked entries in the matrix in Figure 6-3(a). Figure 6-3(b) presents the network graph of edges that must be employed to service these marked VOQ's. Recall that two VOQ's share a vertex if they have an input or output port in common. Note that the edges of the graph in Figure 6-3(b) form the graph C_8, the 8-ring.

In Chapter 8, we demonstrate that C_8 fails Local Pooling. The implication of this result is that we *cannot* conclude that maximal weight matching based scheduling is stable when the set of maximum weighted network graph edges equates to C_8. In our input-queued switch context, this implies that whenever a configuration of maximum weighted edges equivalent to that depicted in Figure 6-3 arises, the network cannot be guaranteed stable.

In the 3×3 scenario above, the appearance of C_6 raised a similar concern. Fortunately though, the appearance of C_6 implied that the arrival rates must belong to a very small, indeed negligible, set within the switch capacity region. Thus, we could effectively dismiss

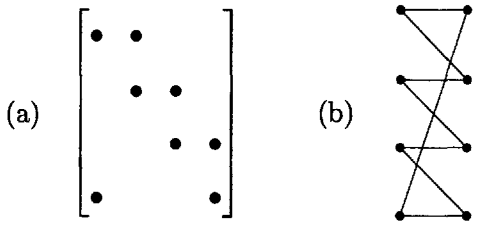


Figure 6-3: The appearance of an 8-ring in the $n_1 \times n_2$ switch, $n_1, n_2 \geq 4$. (a) Marked entries indicate VOQ's of interest. (b) The network graph of edges employed in servicing these VOQ's is an 8-ring.

all instances where C_6 could potentially lead to instability. To emphasize the point, we did not attempt to further analyze the stability properties of the switch in cases where C_6 arises, because such cases can only arise in a negligible subset of the capacity region.

It turns out that the appearance of C_8 does not have this attractive property. The following theorem demonstrates that C_8 can arise in a subset of the capacity region having non-zero measure. Consequently, we have no guarantee that the $n_1 \times n_2$ switch is rate stable under maximal weight matching based scheduling over the entire capacity region up to a set of measure zero. Indeed, the result says that the set of suspicious arrival rates has non-zero measure.

Theorem 6.5.2 *The set of arrival rates under which C_8 can arise as the network graph of simultaneously maximum weighted queues has non-zero Lebesgue measure in the $n_1 \times n_2$ switch capacity region, where $n_1, n_2 \geq 4$.*

Proof: See Appendix 6.D. The tools of this proof resemble the stability considerations that arise in the paper of Dimakis and Walrand [51]. ■

6.5.3 The $2 \times n$ switch

Here we briefly mention a result that follows from our study of *Local Pooling* in the next chapter. We have found earlier in this chapter that the 2×2 switch achieves 100% throughput under maximal weight scheduling. In Chapter 7, we determine several graphs for which Local Pooling is satisfied, and find as a corollary (see Corollary 7.4.2) that any $2 \times n$ switch, with $n \geq 1$, also achieves 100% throughput under maximal weight scheduling.

6.6 Conclusions

In this chapter, we have set out to consider an algorithm that lends itself to distributed implementation in the network setting. Specifically, instead of employing maxweight scheduling, we consider maximal weight schedulers, which can be implemented using localized control algorithms.

This chapter has focused on the input-queued switch. We began by presenting numerical results attesting to the excellent throughput properties of a maximal weight scheduler. For the 2×2 switch, we proved that maximal weight scheduling is throughput optimal. Finally, for the 3×3 switch, we demonstrated that although the network cannot be guaranteed to achieve 100% throughput, the set of arrival rates having suspicious stability properties is a set of measure zero within the switch capacity region.

6.6.1 Future directions

The result of Theorem 6.5.2 does not complete the story regarding the $n \times n$ switch, when $n \geq 4$. It only allows us to say that the set of suspicious arrival rates under which throughput loss may occur has non-zero measure. To actually assert a throughput loss over a non-negligible set of arrival rates, it must be shown that there exist arrival processes under which rate stability fails under maximal weight matching based scheduling. Furthermore, the set of arrival processes for which rate stability fails must have rates that constitute a set of non-zero measure in the switch capacity region.

Appendix

6.A Proof of Theorem 6.4.1 based on fluid limits

The proof begins with a characterization of the fluid limit functions $\bar{Q}_{ij}\,\forall i,j$, $\bar{A}_{ij}\,\forall i,j$, $\bar{D}_{ij}\,\forall i,j$, $\bar{F}_{\mathbf{S}}\,\forall \mathbf{S} \in \mathcal{S}^{\text{sh}}$ (recall that we are considering a single-hop algorithm), identically as in Appendix 2.B. It can be shown that the fluid model equations (2.17)-(2.24) hold.

Let $h : \mathbb{R}_+ \to \mathbb{R}_+$ be defined according to $h(t) = \max_{ij} \bar{Q}_{ij}(t)$. Consider a regular time $t \geq 0$ at which $h(t) > 0$. There are several cases of interest, which we will address in turn. Suppose first that $\max\{\bar{Q}_{11}(t), \bar{Q}_{22}(t)\} > \max\{\bar{Q}_{12}(t), \bar{Q}_{21}(t)\}$. From (2.20) we have that

$$\dot{\bar{Q}}_{11}(t) = \lambda_{11} - \dot{\bar{D}}_{11}(t), \quad \dot{\bar{Q}}_{22}(t) = \lambda_{22} - \dot{\bar{D}}_{22}(t).$$

Because of our assumption, we must have that $\dot{\bar{D}}_{11}(t) = 1$ if $\bar{Q}_{11}(t) \geq \bar{Q}_{22}(t)$ and $\dot{\bar{D}}_{22}(t) = 1$ if $\bar{Q}_{11}(t) \leq \bar{Q}_{22}(t)$. To see why this is true, one must consider the scaled functions that converge to the fluid limit functions, and note that in the locality of time t, the maximal weight scheduling algorithm allocates service exclusively to configuration π_1. Consequently, we obtain

$$\begin{aligned} \dot{h}(t) &\leq \max\{\lambda_{11}, \lambda_{22}\} - 1 \\ &\leq 0, \end{aligned}$$

where the second inequality follows by the assumption of a doubly substochastic arrival rate matrix $\boldsymbol{\lambda}$. By symmetry, we can conclude that $\dot{h}(t) \leq 0$ if $\max\{\bar{Q}_{11}(t), \bar{Q}_{22}(t)\} < \max\{\bar{Q}_{12}(t), \bar{Q}_{21}(t)\}$.

The only remaining case to consider is where $\max\{\bar{Q}_{11}(t), \bar{Q}_{22}(t)\} = \max\{\bar{Q}_{12}(t), \bar{Q}_{21}(t)\}$. This case yields several subcases, as follows.

1. $h(t) = Q_{11}(t) = Q_{12}(t) > \{Q_{21}(t), Q_{22}(t)\}$. Here, we must have $\dot{\bar{D}}_{11}(t) + \dot{\bar{D}}_{12}(t) = 1$. Further, because t is a regular time, we have that $\dot{h}(t) = \lambda_{11} - \dot{\bar{D}}_{11}(t) = \lambda_{12} - \dot{\bar{D}}_{12}(t)$. By algebraic manipulation, we obtain $\dot{h}(t) = \frac{1}{2}(\lambda_{11} + \lambda_{12} - 1) \leq 0$.

2. $h(t) = Q_{11}(t) = Q_{12}(t) = Q_{21}(t) > Q_{22}(t)$. Here, we must have $\dot{\bar{D}}_{11}(t) + \dot{\bar{D}}_{12}(t) = 1$. Also we must have $\dot{\bar{D}}_{12}(t) = \dot{\bar{D}}_{21}(t)$, which implies that this subcase cannot occur unless $\lambda_{12} = \lambda_{21}$. Similarly to the first case, algebraic manipulation provides $\dot{h}(t) \leq 0$.

3. $h(t) = Q_{11}(t) = Q_{12}(t) = Q_{21}(t) = Q_{22}(t)$. Here, we must have $\dot{\bar{D}}_{11}(t) + \dot{\bar{D}}_{12}(t) = 1$. Also we must have $\dot{\bar{D}}_{11}(t) = \dot{\bar{D}}_{22}(t)$ and $\dot{\bar{D}}_{12}(t) = \dot{\bar{D}}_{21}(t)$, which implies that this subcase cannot occur unless $\lambda_{11} = \lambda_{22}$ and $\lambda_{12} = \lambda_{21}$. Similarly to the first case, algebraic manipulation provides $\dot{h}(t) \leq 0$.

This set of subcases is complete, in that any other instance can be translated to one of the above subcases through a relabeling of the switch ports.

We have demonstrated that at any regular time at which $h(t) > 0$, then $\dot{h}(t) \leq 0$. Since $h(0) = 0$, we must then have that $h(t) = 0$ almost everywhere. Then, we must have that $\bar{\mathbf{Q}}(t) = 0$ almost everywhere and the fluid model is weakly stable. Consequently the queueing system under greedy weighted matching is rate stable.

6.B Proof of Theorem 6.4.1 for Bernoulli arrivals using a Lyapunov drift argument

The proof is carried out by demonstrating that the queueing system under the maximal weight matching algorithm is *weakly stable.* We adopt the following characterization for weak stability [87].

Definition 6.B.1 *The Markov Chain* $(\mathbf{Q}(t), t \in \mathbb{Z}_+)$ *is* weakly stable *if there exists a Lyapunov function* V *such that for any* $\epsilon > 0$*, there exists* $B > 0$ *such that*

$$\lim_{t \to \infty} P[V(\mathbf{Q}(t)) > B] < \epsilon.$$

We make use of the following Lyapunov function, $V : \mathbb{R}_+^{2\times 2} \to \mathbb{R}_+$. Let $\mathbf{Q} = (Q_{ij})$ be a 2×2 matrix. Then $V(\mathbf{Q}) = \max_{ij} Q_{ij}$.

For the entire proof, we assume without loss of generality that VOQ$_{11}$ has the maximum number of cells: $Q_{11} \geq Q_{ij}, \forall i, j = 1, 2$. Any other case can be trivially converted to this scenario by relabeling the input/output ports. We divide the proof into two key cases: the first case has π_1 strictly dominating π_2 under greedy matching, and the second case has π_1 equivalent to π_2 as greedy choices.

For the proof, we assume $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*$ is *strictly* doubly substochastic:

$$\sum_j \lambda_{ij} < 1, \forall i \quad \sum_i \lambda_{ij} < 1, \forall j.$$

6.B.1 Case 1: $Q_{11}(t) > \max\{Q_{12}(t), Q_{21}(t)\}$

In this section we consider the simple case where there is no ambiguity about which configuration is dominant. We assume without loss of generality that $Q_{11}(t)$ is *strictly larger* than $\max\{Q_{12}(t), Q_{21}(t)\}$.

Lemma 6.B.1 *When the queue occupancy matrix* $\mathbf{Q}(t)$ *satisfies* $Q_{11}(t) - 1 \geq Q_{12}(t)$*,* $Q_{11}(t) - 1 \geq Q_{21}(t)$*, and* $Q_{11}(t) \geq Q_{22}(t)$*,*

$$E\left[V(\mathbf{Q}(t+1)) - V(\mathbf{Q}(t)) | \mathbf{Q}(t)\right] \leq -(1 - \lambda_{11})(1 - \lambda_{12})(1 - \lambda_{21})(1 - \lambda_{22}).$$

Proof: Assume $V(\mathbf{Q}(t)) \geq 1$, with $Q_{11}(t) - 1 \geq Q_{12}(t)$, $Q_{11}(t) - 1 \geq Q_{21}(t)$, and $Q_{11}(t) \geq Q_{22}(t)$. Designating $a_{ij}(t)$ as the number of arrivals to VOQ$_{ij}$ at the beginning of time slot t, we have

$$\begin{aligned}
Q_{11}(t+1) &= Q_{11}(t) - 1 + a_{11}(t+1) \\
Q_{12}(t+1) &= Q_{12}(t) + a_{12}(t+1) \\
Q_{21}(t+1) &= Q_{21}(t) + a_{21}(t+1) \\
Q_{22}(t+1) &= \max\{Q_{22}(t) - 1, 0\} + a_{22}(t+1)
\end{aligned}$$

We are interested in finding an upper-bound for the expression $E\left[V(\mathbf{Q}(t+1)) - V(\mathbf{Q}(t))|\mathbf{Q}(t)\right]$. Since the quantity $V(\mathbf{Q}(t+1)) - V(\mathbf{Q}(t))$ can only take on the value 0 or -1 in this case (due to Bernoulli arrivals and the service restriction of one cell from each queue per time slot), we must lower bound the probability of the event $\{V(\mathbf{Q}(t+1)) - V(\mathbf{Q}(t)) = -1 \mid \mathbf{Q}(t)\}$. Clearly, this occurs when $Q_{11}(t) - 1 = Q_{12}(t) = Q_{21}(t) = Q_{22}(t) - 1$, since an arrival to any VOQ at the beginning of time slot $t+1$ results in $V(\mathbf{Q}(t+1)) = V(\mathbf{Q}(t))$. The following bounds are then evident, under the assumed Bernoulli-distributed arrivals.

$$P\left[V(\mathbf{Q}(t+1)) = V(\mathbf{Q}(t))|\mathbf{Q}(t)\right] \leq 1 - (1-\lambda_{11})(1-\lambda_{12})(1-\lambda_{21})(1-\lambda_{22}) \tag{6.1}$$

$$P\left[V(\mathbf{Q}(t+1)) = V(\mathbf{Q}(t)) - 1|\mathbf{Q}(t)\right] \geq (1-\lambda_{11})(1-\lambda_{12})(1-\lambda_{21})(1-\lambda_{22}). \tag{6.2}$$

Since (6.1) and (6.2) completely characterize the difference $V(\mathbf{Q}(t+1)) - V(\mathbf{Q}(t))$ in this scenario, we have

$$\begin{aligned}
E\left[V(\mathbf{Q}(t+1)) - V(\mathbf{Q}(t))|\mathbf{Q}(t)\right] &= 0 \cdot P[V(\mathbf{Q}(t+1)) - V(\mathbf{Q}(t)) = 0|\mathbf{Q}(t)] \\
&\quad - 1 \cdot P[V(\mathbf{Q}(t+1)) - V(\mathbf{Q}(t)) = -1|\mathbf{Q}(t)] \\
&\leq -(1-\lambda_{11})(1-\lambda_{12})(1-\lambda_{21})(1-\lambda_{22}).
\end{aligned}$$

■

The remainder of this section treats the case where the maximum element of $\{Q_{11}(t), Q_{22}(t)\}$ equals the maximum element of $\{Q_{12}(t), Q_{21}(t)\}$.

6.B.2 Drift analysis for a simple two queue system

In this section, we analyze a simple two queue system and characterize an important *drift* property of the Markov chain describing the system's queue evolution. We will subsequently (in Section 6.B.3) derive conditions on the 2×2 input-queued switch in order to take advantage of the drift properties of the simple two queue system.

Consider a queueing system consisting of two queues, with respective backlogs at time t equal to $Z_1(t), Z_2(t)$. These queues are subject to Bernoulli arrivals with possibly *time-varying rates*. Let $A_1(t), A_2(t)$ be the cumulative arrivals to queues Z_1, Z_2 respectively, up to and including time t. There is a single server that is only able to serve one of the two queues at a time, with one unit of service at a queue resulting in a reduction in the queue's backlog by one cell at the end of the time slot. The scheduling policy employed is *longest queue first* (LQF), where the queue having maximum backlog is serviced at each slot, with queue 1 chosen as the default queue for service in the event of equal queue backlogs. Suppose at time t, $Z_1(t) = Z_2(t) = a \geq 0$, and there exist $\bar{\tau}, k$, such that the expected number of arrivals to Z_1 and Z_2 are upper-bounded by r_1 and r_2, respectively, for each time slot in the range $t + \bar{\tau}, \ldots, t + \bar{\tau} + k$. We are interested in understanding the expected *drift* of this queueing system from time slot t through time slot $t + \bar{\tau} + k$, given by

$$d(n, \bar{\tau}, k, a) = E[\max\{Z_1(t + \bar{\tau} + k), Z_2(t + \bar{\tau} + k)\} - \max\{Z_1(t), Z_2(t)\} \mid \mathbf{Z}(t) = (a, a)].$$

Above, we denote $\mathbf{Z}(t) = (Z_1(t), Z_2(t))$.

The following lemma demonstrates that when $|Z_1(t) - Z_2(t)| \leq 2$, then after any number of time slots, prior to the arrivals, Z_1 and Z_2 are within 1 unit of each other. For convenience, we denote the queue backlogs prior to arrivals by

$$\hat{Z}_1(t) = Z_1(t) - (A_1(t) - A_1(t-1)) \qquad \hat{Z}_2(t) = Z_2(t) - (A_2(t) - A_2(t-1)).$$

Lemma 6.B.2 *When $|Z_1(t) - Z_2(t)| \leq 2$, then for $k \geq 1$, $|\hat{Z}_1(t + k) - \hat{Z}_2(t + k)| \leq 1$.*

Proof: Our proof is by induction. Assume $|Z_1(t) - Z_2(t)| \leq 2$. Without loss of generality, we assume that $\max\{Z_1(t), Z_2(t)\} = Z_1(t)$, from which we can assume without loss of generality that under LQF, queue 1 is selected for service at time slot t. Then, $|\hat{Z}_1(t+1) - \hat{Z}_2(t+1)| \leq 1$. For the inductive step, assume that $|\hat{Z}_1(t + k) - \hat{Z}_2(t + k)| \leq 1$. Thus it must be true that the arrivals at time slot $t{+}k$ are such that $|Z_1(t{+}k) - Z_2(t{+}k)| \leq 2$. Under LQF, the service over time slot k is applied to queue i, where $Z_i(t + k) \geq Z_j(t + k)$, $i \neq j$. This implies $|\hat{Z}_1(t + k + 1) - \hat{Z}_2(t + k + 1)| \leq 1$, which completes the induction. ■

The following lemma bounds the $(\bar{\tau} + k)$-slot drift in our two-queue system, when both queue occupancies are initially equal.

Lemma 6.B.3 *Consider the case $Z_1(t) = Z_2(t) = a$. Suppose there exist integers $\bar{\tau}, k \geq 0$ such that for each slot in the range $t + \bar{\tau} + 1, \ldots, t + \bar{\tau} + k$, the expected number of arrivals to queues $1, 2$ are upper-bounded by r_1, r_2, respectively. Then, when $a \geq \bar{\tau} + k$,*

$$d(t, \bar{\tau}, k, a) \leq \frac{3 + \bar{\tau}}{2} - \frac{r_1 + r_2}{2} - \frac{k(1 - r_1 - r_2)}{2} \tag{6.3}$$

Proof: Let $Z_1(t) = Z_2(t) = a \geq \bar{\tau} + k$. For time slots $t, \ldots, t + \bar{\tau}$, a maximum of $2\bar{\tau}$ arrivals can occur in total to both queues. Over $\bar{\tau}$ slots, $\bar{\tau}$ total cells must be serviced from the queues under LQF. By Lemma 6.B.2, $|Z_1(t{+}\bar{\tau}) - Z_2(t{+}\bar{\tau})| \leq 2$. Thus, the total number of cells in both queues at time $t{+}\bar{\tau}$ is at most $2a{+}\bar{\tau}$, which implies $\max\{Z_1(t{+}\bar{\tau}), Z_2(t{+}\bar{\tau})\} \leq a + \bar{\tau}/2 + 1$.

Subsequent to time $t + \bar{\tau}$, assume for each time slot that the expected number of arrivals to queues Z_1, Z_2 is upper-bounded by r_1, r_2, respectively. We consider the sum total number of arrivals to both queues from time slot $t + \bar{\tau} + 1$ through time $t + \bar{\tau} + k - 1$. Define

$$\rho(i, \bar{\tau}, k) \triangleq P[A_1(t + \bar{\tau} + k - 1) - A_1(t + \bar{\tau}) + A_2(t + \bar{\tau} + k - 1) - A_2(t + \bar{\tau}) = i \mid \mathbf{Z}(t)].$$

From Lemma 6.B.2, it is clear that after $k-1$ arrival opportunities and k service opportunities to the queues Z_1, Z_2, irrespective of the manner in which the arrivals and service occurred, the queue backlogs must be within 1 cell of one another. We can then conclude that if i total arrivals occurred to the queues over time slots $t+\bar{\tau}+1, \ldots, t+\bar{\tau}+k$, the queue backlogs (in no particular order) are upper-bounded by

$$a + \left\lceil \frac{\bar{\tau}+i-k}{2} \right\rceil, \quad a + \left\lfloor \frac{\bar{\tau}+i-k}{2} \right\rfloor. \tag{6.4}$$

To complete the characterization of the drift, we must account for the arrivals at time slot $t+\bar{\tau}+k$. A sufficient bound is to assign probability 1 to the occurrence of an additional arrival to the maximum-valued queue. The drift is then upper-bounded by

$$\begin{aligned} d(t,\bar{\tau},k,a) &\leq \sum_{i=0}^{2k-2} \left(1 + \frac{1}{2} + \frac{\bar{\tau}+i-k}{2}\right) \rho(i,\bar{\tau},k), && (6.5) \\ &= \frac{3+\bar{\tau}-k}{2} + \frac{1}{2} \sum_{i=0}^{2k-2} i\rho(i,\bar{\tau},k), \\ &= \frac{3+\bar{\tau}-k}{2} + \frac{1}{2} E[A_1(t+\bar{\tau}+k-1) - A_1(t+\bar{\tau}) + A_2(t+\bar{\tau}+k-1) - A_2(t+\bar{\tau})]. \\ & && (6.6) \end{aligned}$$

The limits of the sum in (6.5) account for up to $2(k-1)$ total arrivals to the queues over $k-1$ time slots. The 1 in (6.5) corresponds to the additional cell whose arrival occurs with probability 1 at time $t+\bar{\tau}+k$, and the 1/2 term in (6.5) provides a bound on the ceiling of (6.4). The expectation at right in (6.6) is effectively upper-bounded as follows,

$$\begin{aligned} & E[A_1(t+\bar{\tau}+k-1) - A_1(t+\bar{\tau}) + A_2(t+\bar{\tau}+k-1) - A_2(t+\bar{\tau})] \\ &= \sum_{i=0}^{k-2} (E[A_1(t+\bar{\tau}+i+1) - A_1(t+\bar{\tau}+i)] + E[A_2(t+\bar{\tau}+i+1) - A_2(t+\bar{\tau}+i)]), \\ &\leq (k-1)(r_1+r_2). && (6.7) \end{aligned}$$

Above, (6.7) follows because at each time slot, we have assumed that the expected number of Bernoulli arrivals at queues Z_1, Z_2 are upper-bounded by r_1, r_2 respectively. (6.3) follows immediately. ■

We have now established the necessary tools that will be used in our remaining development of the stability of the 2 × 2 switch.

6.B.3 Case 2: $Q_{11}(t) = \max\{Q_{12}(t), Q_{21}(t)\}$

In this section, we return our attention to the 2 × 2 switch, and we deal with the more interesting case of when both π_1 and π_2 are valid selections according to the greedy weighted matching algorithm. We assume without loss of generality that $\mathbf{Q}(t)$ has the form

$$\mathbf{Q}(t) = \begin{bmatrix} a & a \\ b & c \end{bmatrix}, \tag{6.8}$$

where $a \geq b \geq 0$ and $a \geq c \geq 0$.

We make use of the same notation as in the previous section, namely the queuing variables Z_1 and Z_2. We will demonstrate how these variables fit naturally in the analysis of Section 6.B.2. We study the probability of an increase in either $Z_1(t+t') = \max\{Q_{11}(t+t'), Q_{22}(t+t')\}$ or $Z_2(t+t') = \max\{Q_{12}(t+t'), Q_{21}(t+t')\}$ due to cell arrivals. Here, Z_1, Z_2 may be considered as a pair of induced queues corresponding to π_1, π_2, respectively. We shall study the conditions under which Z_1 and Z_2 satisfy the conditions of Lemma 6.B.3. Given $\mathbf{Q}(t)$ of the form (6.8), the expected number of arrivals at Z_1 at time $t+t'$ is given by

$$\begin{aligned}\mu_1(t+t') &= (\lambda_{11} + \lambda_{22} - \lambda_{11}\lambda_{22})P[Q_{11}(t+t') = Q_{22}(t+t')|\mathbf{Q}(t)] \\ &\quad + \lambda_{11}P[Q_{11}(t+t') > Q_{22}(t+t')|\mathbf{Q}(t)] + \lambda_{22}P[Q_{11}(t+t') < Q_{22}(t+t')|\mathbf{Q}(t)].\end{aligned}$$

Similarly, the expected number of arrivals at Z_2 at time $t+m$ is given by

$$\begin{aligned}\mu_2(t+t') &= (\lambda_{12} + \lambda_{21} - \lambda_{12}\lambda_{21})P[Q_{12}(t+t') = Q_{21}(t+t')|\mathbf{Q}(t)] \\ &\quad + \lambda_{12}P[Q_{12}(t+t') > Q_{21}(t+t')|\mathbf{Q}(t)] + \lambda_{21}P[Q_{12}(t+t') < Q_{21}(t+t')|\mathbf{Q}(t)].\end{aligned}$$

Define

$$\gamma = \frac{1}{4}\left(1 - \max\{\lambda_{11}, \lambda_{22}\} - \max\{\lambda_{12}, \lambda_{21}\}\right).$$

If we can obtain the bound

$$(\lambda_{11} + \lambda_{22} - \lambda_{11}\lambda_{22})P[Q_{11}(t+t') = Q_{22}(t+t')|\mathbf{Q}(t)] < \gamma, \tag{6.9}$$

then we can conclude

$$\begin{aligned}\mu_1(t+t') &< \frac{1}{4}\left(1 - \max\{\lambda_{11}, \lambda_{22}\} - \max\{\lambda_{12}, \lambda_{21}\}\right) + \max\{\lambda_{11}, \lambda_{22}\} \\ &= \frac{1}{4} + \frac{3}{4}\max\{\lambda_{11}, \lambda_{22}\} - \frac{1}{4}\max\{\lambda_{12}, \lambda_{21}\} \\ &\triangleq r_1.\end{aligned}$$

Similarly for $\mu_2(t+t')$, if

$$(\lambda_{12} + \lambda_{21} - \lambda_{12}\lambda_{21})P[Q_{12}(t+t') = Q_{21}(t+t')|\mathbf{Q}(t)] < \gamma, \tag{6.10}$$

then we can conclude

$$\begin{aligned}\mu_2(t+t') &< \frac{1}{4}\left(1-\max\{\lambda_{11},\lambda_{22}\}-\max\{\lambda_{12},\lambda_{21}\}\right)+\max\{\lambda_{12},\lambda_{21}\}\\ &= \frac{1}{4}-\frac{1}{4}\max\{\lambda_{11},\lambda_{22}\}+\frac{3}{4}\max\{\lambda_{12},\lambda_{21}\}\\ &\triangleq r_2.\end{aligned}$$

In this section we will establish the upper-bounds r_1, r_2, implying that the induced queues Z_1, Z_2 will fit into the drift analysis of Section 6.B.2.

Define the constant

$$\xi_0 = \left\lceil \frac{3-r_1-r_2}{1-r_1-r_2} \right\rceil + 1,$$

such that Lemma 6.B.3 can be applied to induced queues Z_1, Z_2, with values $\bar{\tau} = 0, k = \xi_0, a \geq 2\xi_0$ such that $d(n, 0, \xi_0, a) < 0$ when $r_1 + r_2 < 1$.

Lemma 6.B.4 *If $a-c > 2\xi_0$ and $a-b > 2\xi_0$, and $a \geq 2\xi_0+1$, then*

$$P[Q_{11}(t+t') = Q_{22}(t+t')|\mathbf{Q}(t)] = 0, \quad t' = 1,2,\ldots,\xi_0 \tag{6.11}$$

$$P[Q_{12}(t+t') = Q_{21}(t+t')|\mathbf{Q}(t)] = 0, \quad t' = 1,2,\ldots,\xi_0 \tag{6.12}$$

Proof: Over time slots $t,\ldots,t+\xi_0-1$, the greedy matching algorithm ensures Q_{11} decreases by at most ξ_0 cells. The backlog Q_{22} cannot increase by more than ξ_0 cells over time slots $t,\ldots,t+\xi_0$. Then we must have that $Q_{11}(t+t') \geq a-\xi_0$ and $Q_{22}(t+t') < a-\xi_0$ for $t' = 1,2,\ldots,t+\xi_0$, from which (6.11) follows. The proof for (6.12) follows identically and is omitted. ■

Lemma 6.B.4 ensures that (6.9) and (6.10) are satisfied over slots $t,\ldots,t+\xi_0$. Thus the drift of induced queues Z_1, Z_2 in any switch state satisfying the assumptions of Lemma 6.B.4 is negative after ξ_0 time slots.

Lemma 6.B.5 *If $a-c \leq 2\xi_0$, then there exist $\bar{\tau}_1, \xi_1$ such that if $a-b > 2(\bar{\tau}_1+\xi_1)$ and $a \geq 2\xi_0+2\xi_1+2\bar{\tau}_1$ then*

$$P[Q_{11}(t+t') = Q_{22}(t+t')|\mathbf{Q}(t)] < \frac{\gamma}{\lambda_{11}+\lambda_{22}-\lambda_{11}\lambda_{22}}, \quad t' = \bar{\tau}_1+1,\ldots,\bar{\tau}_1+\xi_1 \tag{6.13}$$

$$P[Q_{12}(t+t') = Q_{21}(t+t')|\mathbf{Q}(t)] = 0, \quad t' = 1,\ldots,\bar{\tau}_1+\xi_1 \tag{6.14}$$

Proof: For any t' such that over time slots $t,\ldots,t+t'$, neither VOQ$_{11}$ or VOQ$_{22}$ reaches zero occupancy, we have

$$Q_{11}(t+t') - Q_{22}(t+t') = a-c+\left(A_{11}(t+t')-A_{11}(t)\right)-\left(A_{22}(t+t')-A_{22}(t)\right). \tag{6.15}$$

The expression $(A_{11}(t+t')-A_{11}(t))-(A_{22}(t+t')-A_{22}(t))$ can be regarded as a summation of t' i.i.d. random variables, each of which take values from the set $\{-1,0,1\}$. Using

(6.15), we have for any t' such that over time slots $t, \ldots, t+t'$, neither VOQ_{11} or VOQ_{22} reaches zero occupancy that

$$P[Q_{11}(t+t') = Q_{22}(t+t')|\mathbf{Q}(t)] = P[\big(A_{11}(t+t') - A_{11}(t)\big) - \big(A_{22}(t+t') - A_{22}(t)\big) = c-a].$$

We throw out the case of $\lambda_{11} = \lambda_{22} = 0$, since in this case, $Q_{11}(t) = 0, Q_{22}(t) = 0, \forall n$ almost surely. For any other $\lambda_{11}, \lambda_{22}$ values, a simple normal approximation to the i.i.d. summation guarantees the existence of $\bar{\tau}_1^{a-c}$ such that for all $t' > \bar{\tau}_1^{a-c}$,

$$P[\big(A_{11}(t+t') - A_{11}(t)\big) - \big(A_{22}(t+t') - A_{22}(t)\big) = c-a] < \frac{\gamma}{\lambda_{11} + \lambda_{22} - \lambda_{11}\lambda_{22}}. \tag{6.16}$$

Taking the maximum over all $a-c$ values considered in this lemma, we obtain a common value $\bar{\tau}_1$ such that after time $t + \bar{\tau}_1$, (6.16) is satisfied: $\bar{\tau}_1 = \max_{(a-c)\in\{0,\ldots,2\xi_0\}} \bar{\tau}_1^{a-c}$. Note $\bar{\tau}_1$ is a constant derived only from the constants $\lambda_{11}, \lambda_{22}$.

Define the constant
$$\xi_1 = \left\lceil \frac{3 + \bar{\tau}_1 - r_1 - r_2}{1 - r_1 - r_2} \right\rceil + 1,$$

Suppose $a \geq 2\xi_0 + 2\xi_1 + 2\bar{\tau}_1$. Then since $a - c \leq 2\xi_0$, we have $c \geq 2(\bar{\tau}_1 + \xi_1)$. Thus, for the first $\bar{\tau}_1 + \xi_1$ services to configuration $\boldsymbol{\pi}_1$, both Q_{11} and Q_{22} are reduced by one cell at each service (since neither queue reaches zero occupancy). Since $c \geq 2(\bar{\tau}_1 + \xi_1)$ guarantees that $Q_{11}(t) > 0$ and $Q_{22}(t) > 0$ for time slots $t, \ldots, t + \bar{\tau}_1 + \xi_1$, we conclude that (6.13) is satisfied. Finally, assuming $a - b > 2(\bar{\tau}_1 + \xi_1)$, then following in a similar manner to the proof of Lemma 6.B.4, there is no sample path on which the queue backlogs Q_{12} and Q_{21} coincide over time slots $t, \ldots, t + \bar{\tau}_1 + \xi_1$, giving (6.14) as desired. ■

Lemma 6.B.5 provides that (6.9) and (6.10) are satisfied over slots $t+\bar{\tau}_1, \ldots, t+\bar{\tau}_1+\xi_1$. We have defined ξ_1 such that Lemma 6.B.3 can be applied to induced queues Z_1, Z_2, with values $\bar{\tau} = \bar{\tau}_1, k = \xi_1, a \geq 2\xi_0 + 2\xi_1 + 2\bar{\tau}_1$ such that $d(t, \bar{\tau}_1, \xi_1, a) < 0$ when $r_1 + r_2 < 1$. Thus the drift of induced queues Z_1, Z_2 in any switch state satisfying the assumptions of Lemma 6.B.5 is negative after $\xi_1 + \bar{\tau}_1$ time slots.

Corollary 6.B.1 *If $a - b \leq 2\xi_0$, then there exist $\bar{\tau}_2, \xi_2$ such that if $a - c > 2(\bar{\tau}_2 + \xi_2)$ and $a \geq 2\xi_0 + 2\xi_2 + 2\bar{\tau}_2$ then*

$$P[Q_{11}(t+t') = Q_{22}(t+t')|\mathbf{Q}(t)] = 0, \quad t' = 1, \ldots, \bar{\tau}_2 + \xi_2$$
$$P[Q_{12}(t+t') = Q_{21}(t+t')|\mathbf{Q}(t)] < \frac{\gamma}{\lambda_{11} + \lambda_{22} - \lambda_{11}\lambda_{22}}, \quad t' = \bar{\tau}_2 + 1, \ldots, \bar{\tau}_2 + \xi_2$$

Proof: The proof follows identically to the proof of Lemma 6.B.5. ■

Note in Corollary 6.B.1 that ξ_2 is the number of time slots required to ensure $d(t, \bar{\tau}_2, \xi_2, a) < 0$, when $a \geq 2\xi_0 + 2\bar{\tau}_2 + 2\xi_2$. Thus the drift of induced queues Z_1, Z_2 in any switch state satisfying the assumptions of Corollary 6.B.1 is negative after $\xi_2 + \bar{\tau}_2$ time slots.

Denote $\bar{\tau}_3 = \max\{\bar{\tau}_1, \bar{\tau}_2\}$ and $\xi_3 = \max\{\xi_1, \xi_2\}$.

Table 6.1: Required values for $a, \bar{\tau}, k$ for different possible $a-b, a-c$ values, such that when $\mathbf{Q}(t)$ is as in (6.8), then the drift of the induced queues Z_1, Z_2 is negative: $d(n, \bar{\tau}, k, a) < 0$.

$a-b$	$a-c$	$a \geq$	$\bar{\tau}$	k
$> 2\xi_0$	$> 2\xi_0$	$2\xi_0$	0	ξ_0
$> 2\bar{\tau}_1 + \xi_1$	$\leq 2\xi_0$	$2\xi_0 + 2\xi_1 + 2\bar{\tau}_1$	$\bar{\tau}_1$	ξ_1
$\leq 2\xi_0$	$> 2\bar{\tau}_2 + 2\xi_2$	$2\xi_0 + 2\xi_2 + 2\bar{\tau}_2$	$\bar{\tau}_2$	ξ_2
$\leq 2(\bar{\tau}_3 + \xi_3)$	$\leq 2(\bar{\tau}_3 + \xi_3)$	$3(\bar{\tau}_3 + \xi_3)$	$\bar{\tau}_3$	ξ_3

Lemma 6.B.6 *If $a - b \leq 2(\bar{\tau}_3 + \xi_3)$ and $a - c \leq 2(\bar{\tau}_3 + \xi_3)$ and $a \geq 3(\bar{\tau}_3 + \xi_3)$ then*

$$P[Q_{11}(t+t') = Q_{22}(t+t')|\mathbf{Q}(t)] < \frac{\gamma}{\lambda_{11} + \lambda_{22} - \lambda_{11}\lambda_{22}}, \quad t' = \bar{\tau}_3 + 1, \ldots, \bar{\tau}_3 + \xi_3 \quad (6.17)$$

$$P[Q_{12}(t+t') = Q_{21}(t+t')|\mathbf{Q}(t)] < \frac{\gamma}{\lambda_{11} + \lambda_{22} - \lambda_{11}\lambda_{22}}, \quad t' = \bar{\tau}_3 + 1, \ldots, \bar{\tau}_3 + \xi_3 \quad (6.18)$$

Proof: From Lemma 6.B.5 and Corollary 6.B.1, it is clear that $\bar{\tau}_3$ slots are sufficient to ensure convergence of the expected number of arrivals to induced queues Z_1, Z_2 to less than r_1, r_2, respectively. This entire discussion is valid so long as no queue reaches zero occupancy over time slots $t, \ldots, t + \bar{\tau}_3 + \xi_3 - 1$. Clearly, restricting $a \geq 3(\bar{\tau}_3 + \xi_3)$ and $b, c \geq \bar{\tau}_3 + \xi_3$ is sufficient to guarantee this condition. Thus (6.17) and (6.18) follow as desired. ■

Note in Lemma 6.B.6 that ξ_3 is the number of time slots required to ensure $d(t, \bar{\tau}_3, \xi_3, a) < 0$, when $a \geq 3\bar{\tau}_3 + 3\xi_3$. Thus the drift of induced queues Z_1, Z_2 in any switch state satisfying the assumptions of Lemma 6.B.6 is negative after $\xi_3 + \bar{\tau}_3$ time slots.

We have organized the results of this section in Table 6.1. The table shows the range of $a - b, a - c$ values accounted for by the results of this section. Since $a \geq b, a \geq c$, we conclude that all possible values have been considered. We now have all the tools necessary to prove Theorem 6.4.1.

6.B.4 Proof of Theorem 6.4.1

The proof is through a Lyapunov drift argument. We will determine a sequence of time slot indices $(\zeta_i, i \in \mathbb{Z}_+)$ such that $\zeta_0 = 0$, there exists $M < \infty$ such that $\zeta_{i+1} - \zeta_i < M$, and

$$E[V(\mathbf{Q}(\zeta_{i+1})) - V(\mathbf{Q}(\zeta_i))|\mathbf{Q}(\zeta_i)] < 0, \; V(\mathbf{Q}(\zeta_i)) > \nu \quad (6.19)$$

$$E[V(\mathbf{Q}(\zeta_{i+1}))|\mathbf{Q}(\zeta_i)] < \infty, \; V(\mathbf{Q}(\zeta_i)) \leq \nu \quad (6.20)$$

for all $i \geq 0$.

Consider any integer $i \geq 0$. At time ζ_i, suppose $V(\mathbf{Q}(\zeta_i)) \geq 2\xi_0 + 3(\bar{\tau}_3 + \xi_3) \triangleq \nu$. The system state $\mathbf{Q}(\zeta_i)$ is accounted for by one of Lemmas 6.B.1, 6.B.4, 6.B.5, 6.B.6, and

Table 6.2: Values of $\zeta_{i+1} - \zeta_i$ required under the conditions of the Lemmas and Corollary of Section 6.B.3

Case covered by	$\zeta_{i+1} - \zeta_i$
Lemma 6.B.1	1
Lemma 6.B.4	ξ_0
Lemma 6.B.5	$\bar{\tau}_1 + \xi_1$
Corollary 6.B.1	$\bar{\tau}_2 + \xi_2$
Lemma 6.B.6	$\bar{\tau}_3 + \xi_3$

Corollary 6.B.1. For Lemma 6.B.1 to be valid, we require $V(\mathbf{Q}(\zeta_i)) \geq 1$, which is satisfied by ν. For Lemmas 6.B.4, 6.B.5, 6.B.6 and Corollary 6.B.1, the values are listed as restrictions on the variable a in Table 6.1. Each of these are satisfied by ν. Recall the condition for achieving negative drift under the $\bar{\tau}, k$ values listed in Table 6.1 is that $r_1 + r_2 < 1$. Under this condition the values of $\bar{\tau}, k$ in Table 6.1 will yield negative drift, with the values for ζ_{i+1} listed in Table 6.2. This completes the characterization of (6.19).

The only remaining case is when $V(\mathbf{Q}(\zeta_i)) \leq \nu$. In this case, we use $\zeta_{i+1} = \zeta_i + 1$. Then it is clear under Bernoulli arrivals that $|V(\mathbf{Q}(\zeta_{i+1})) - V(\mathbf{Q}(\zeta_i))| \leq 1$, implying (6.20).

The above application of Lemma 6.B.1 requires only that $0 \leq \lambda_{ij} < 1, \forall i, j$. Further, the above application of Lemmas 6.B.3, 6.B.4, 6.B.5, 6.B.6, and Corollary 6.B.1 holds for any $r_1 + r_2 < 1$. The following lemma connects this requirement to the admissible region of rates.

Lemma 6.B.7 $r_1 + r_2 < 1$ *if and only if* $\boldsymbol{\lambda}$ *is strictly doubly substochastic.*

Proof: Suppose $r_1 + r_2 < 1$. This is equivalent to

$$\frac{1}{2} + \frac{1}{2}\max\{\lambda_{11}, \lambda_{22}\} + \frac{1}{2}\max\{\lambda_{12}, \lambda_{21}\} < 1. \tag{6.21}$$

Rearranging, we obtain $\max\{\lambda_{11}, \lambda_{22}\} + \max\{\lambda_{12}, \lambda_{21}\} < 1$, which implies $\boldsymbol{\lambda}$ is strictly doubly substochastic. Conversely, suppose that $\boldsymbol{\lambda}$ is strictly doubly substochastic. Then (6.21) is satisfied, and we conclude $r_1 + r_2 < 1$. ■

Thus, our proof of stability will follow for any strictly doubly substochastic $\boldsymbol{\lambda}$. In order to conclude that Foster's Criteria [13, Ch I, Prop. 5.3] are satisfied for positive recurrence of the embedded Markov Chain $(\mathbf{Q}(\zeta_i), i \in \mathbb{Z}_+)$, it is necessary (and trivial) to demonstrate that the embedded chain has a single irreducible class.

Since $(\mathbf{Q}(\zeta_i), i \in \mathbb{Z}_+)$ is irreducible, $\inf_{\mathbf{Q} \in \mathbb{R}_+^{2\times 2}} V(\mathbf{Q}) = 0$, and by (6.19), (6.20), [13, Ch I Prop. 5.3] implies $(\mathbf{Q}(\zeta_i), i \in \mathbb{Z}_+)$ is positive recurrent. As explained in [87], we may then conclude that for any $\epsilon > 0$, there exists finite $B_1 > 0$ such that

$$\lim_{i \to \infty} P[V(\mathbf{Q}(\zeta_i)) > B_1] < \epsilon. \tag{6.22}$$

Finally, we turn our attention to the weak stability of the queue backlog process, $(\mathbf{Q}(t), t \in \mathbb{Z}_+)$. Define $\kappa(t)$ as the maximum index j such that $\zeta_j \leq t$: $\kappa(t) = \max\{j : \zeta_j \leq t\}$. Also, define $M = \max\{1, \xi_0, \bar{\tau}_1 + \xi_1, \bar{\tau}_2 + \xi_2, \bar{\tau}_3 + \xi_3\}$. Clearly M is a finite constant providing an upper bound on $\zeta_{i+1} - \zeta_i$ for any $i \geq 0$. Then

$$\begin{aligned} V(\mathbf{Q}(t)) &= V(\mathbf{Q}(\zeta_{\kappa(t)})) + \big(V(\mathbf{Q}(t)) - V(\mathbf{Q}(\zeta_{\kappa(t)}))\big) \\ &\leq V(\mathbf{Q}(\zeta_{\kappa(t)})) + M. \end{aligned} \tag{6.23}$$

Above, (6.23) follows by the fact that the maximum queue backlog in the system can only increase by 1 cell at each slot, and that there are at most M slots between times $\kappa(t)$ and t. Then it immediately follows that for $B_2 > 0$, $P[V(\mathbf{Q}(t)) > B_2] \leq P[V(\mathbf{Q}(\zeta_{\kappa(t)})) + M > B_2]$. Then we have the following series of equations,

$$\begin{aligned} \lim_{t\to\infty} P[V(\mathbf{Q}(t)) > B_2] &\leq \lim_{t\to\infty} P[V(\mathbf{Q}(\zeta_{\kappa(t)})) + M > B_2] \\ &= \lim_{t\to\infty} P[V(\mathbf{Q}(\zeta_t)) > B_2 - M]. \end{aligned} \tag{6.24}$$

Above, (6.24) follows since $\kappa(t) \to \infty$ as $t \to \infty$. Using (6.22), we conclude for any $B_2 \geq B_1 + M$ that $\lim_{t\to\infty} P[V(\mathbf{Q}(t)) > B_2] < \epsilon$. Thus, we have the weak stability of the queue backlog process as desired, for all stricly doubly substochastic arrival rate matrices $\boldsymbol{\lambda}$. We conclude that greedy maximal matching achieves 100% throughput.

6.C Proof of Theorem 6.5.1

As in the proof of Theorem 6.4.1 (see Appendix 6.A), the proof begins with a characterization of the fluid limit functions $\bar{Q}_{ij}\,\forall i,j$, $\bar{A}_{ij}\,\forall i,j$, $\bar{D}_{ij}\,\forall i,j$, $\bar{F}_{\mathbf{S}}\,\forall \mathbf{S} \in \mathcal{S}^{\text{sh}}$ (recall that we are considering a single-hop algorithm), identically as in Appendix 2.B. It can be shown that the fluid model equations (2.17)-(2.24) hold.

Let $h : \mathbb{R}_+ \to \mathbb{R}_+$ be defined according to $h(t) = \max_{ij} \bar{Q}_{ij}(t)$. Consider a regular time $t \geq 0$ at which $h(t) > 0$. We must consider all possible queue configurations that realize the maximum value $h(t)$. The unique configurations (up to isomorphism) are represented in Table 6.3.

Cases 1, 3, 7. Note that every possible switch configuration that would be selected by a maximal scheduling algorithm must service VOQ_{11}. Consequently, we must have $\dot{\bar{D}}_{11}(t) = 1$. Since each of these cases has $h(t) = \max_{ij} \bar{Q}_{ij}(t) = \bar{Q}_{11}(t)$, and t is a regular time,

$$\begin{aligned} \dot{h}(t) &= \dot{\bar{Q}}_{11}(t) \\ &= \lambda_{11} - 1 \\ &\leq 0. \end{aligned}$$

The above inequality holds because $\boldsymbol{\lambda}$ is an admissible arrival rate matrix (equivalently, $\boldsymbol{\lambda}$ is doubly substochastic).

Cases 2, 5, 6, 9, 10–12, 16, 17. Note that every possible switch configuration that

Table 6.3: Possible VOQ configurations that realize the maximum value $h(t)$. Each of the 25 configurations is numbered. Additionally, the number of distinct configurations equivalent to the one depicted is provided for each configuration.

$$\underset{\text{1: 9 equiv.}}{\begin{bmatrix}\bullet\\ {}\\ {}\end{bmatrix}}\quad
\underset{\text{2: 18 equiv.}}{\begin{bmatrix}\bullet & \bullet\\ & \\ & \end{bmatrix}}\quad
\underset{\text{3: 18 equiv.}}{\begin{bmatrix}\bullet & \\ & \bullet\\ & \end{bmatrix}}\quad
\underset{\text{4: 6 equiv.}}{\begin{bmatrix}\bullet & \bullet & \bullet\\ & & \\ & & \end{bmatrix}}\quad
\underset{\text{5: 36 equiv.}}{\begin{bmatrix}\bullet & \bullet\\ \bullet & \\ & \end{bmatrix}}\quad
\underset{\text{6: 36 equiv.}}{\begin{bmatrix}\bullet & \bullet & \\ & & \bullet\\ & & \end{bmatrix}}\quad
\underset{\text{7: 6 equiv.}}{\begin{bmatrix}\bullet & & \\ & \bullet & \\ & & \bullet\end{bmatrix}}$$

$$\underset{\text{8: 36 equiv.}}{\begin{bmatrix}\bullet & \bullet & \bullet\\ \bullet & & \\ & & \end{bmatrix}}\quad
\underset{\text{9: 9 equiv.}}{\begin{bmatrix}\bullet & \bullet\\ \bullet & \bullet\\ & \end{bmatrix}}\quad
\underset{\text{10: 36 equiv.}}{\begin{bmatrix}\bullet & \bullet & \\ \bullet & & \\ & & \bullet\end{bmatrix}}\quad
\underset{\text{11: 36 equiv.}}{\begin{bmatrix}\bullet & \bullet & \\ & \bullet & \bullet\\ & & \end{bmatrix}}\quad
\underset{\text{12: 9 equiv.}}{\begin{bmatrix}\bullet & \bullet & \\ & & \bullet\\ & & \bullet\end{bmatrix}}\quad
\underset{\text{13: 36 equiv.}}{\begin{bmatrix}\bullet & \bullet & \bullet\\ \bullet & \bullet & \\ & & \end{bmatrix}}\quad
\underset{\text{14: 9 equiv.}}{\begin{bmatrix}\bullet & \bullet & \bullet\\ \bullet & & \\ \bullet & & \end{bmatrix}}$$

$$\underset{\text{15: 36 equiv.}}{\begin{bmatrix}\bullet & \bullet & \bullet\\ \bullet & & \\ & \bullet & \end{bmatrix}}\quad
\underset{\text{16: 36 equiv.}}{\begin{bmatrix}\bullet & \bullet & \\ & & \bullet\\ & \bullet & \bullet\end{bmatrix}}\quad
\underset{\text{17: 9 equiv.}}{\begin{bmatrix}\bullet & \bullet & \\ \bullet & \bullet & \\ & & \bullet\end{bmatrix}}\quad
\underset{\text{18: 6 equiv.}}{\begin{bmatrix}\bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet\\ & & \end{bmatrix}}\quad
\underset{\text{19: 36 equiv.}}{\begin{bmatrix}\bullet & \bullet & \bullet\\ \bullet & \bullet & \\ & & \bullet\end{bmatrix}}\quad
\underset{\text{20: 36 equiv.}}{\begin{bmatrix}\bullet & \bullet & \bullet\\ \bullet & \bullet & \\ \bullet & & \end{bmatrix}}\quad
\underset{\text{21: 6 equiv.}}{\begin{bmatrix}\bullet & \bullet & \\ & \bullet & \bullet\\ \bullet & & \bullet\end{bmatrix}}$$

$$\underset{\text{22: 18 equiv.}}{\begin{bmatrix}\bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet\\ \bullet & & \end{bmatrix}}\quad
\underset{\text{23: 18 equiv.}}{\begin{bmatrix}\bullet & \bullet & \bullet\\ \bullet & \bullet & \\ & \bullet & \bullet\end{bmatrix}}\quad
\underset{\text{24: 9 equiv.}}{\begin{bmatrix}\bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet\\ \bullet & \bullet & \end{bmatrix}}\quad
\underset{\text{25: 1 equiv.}}{\begin{bmatrix}\bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet\end{bmatrix}}$$

would be selected by a maximal scheduling algorithm in each of these cases must service either VOQ$_{11}$ or VOQ$_{12}$. Thus, $\dot{\bar{D}}_{11}(t) + \dot{\bar{D}}_{12}(t) = 1$. Since $h(t) = \bar{Q}_{11}(t) = \bar{Q}_{12}(t)$, and t is a regular time,

$$\begin{aligned}
2\dot{h}(t) &= \dot{\bar{Q}}_{11}(t) + \dot{\bar{Q}}_{12}(t) \\
&= \lambda_{11} + \lambda_{12} - \dot{\bar{D}}_{11}(t) - \dot{\bar{D}}_{12}(t) \\
&= \lambda_{11} + \lambda_{12} - 1 \\
&\leq 0
\end{aligned}$$

The above inequality holds because $\boldsymbol{\lambda}$ is doubly substochastic.

Cases 4, 8, 13–15, 18–20, 22–25. Note that every possible switch configuration that would be selected by a maximal scheduling algorithm in each of these cases must service either VOQ$_{11}$, VOQ$_{12}$, or VOQ$_{13}$. Thus, $\dot{\bar{D}}_{11}(t) + \dot{\bar{D}}_{12}(t) + \dot{\bar{D}}_{13}(t) = 1$. Since $h(t) =$

$\bar{Q}_{11}(t) = \bar{Q}_{12}(t) = \bar{Q}_{13}(t)$, and t is a regular time,

$$\begin{aligned}
3\dot{h}(t) &= \dot{\bar{Q}}_{11}(t) + \dot{\bar{Q}}_{12}(t) + \dot{\bar{Q}}_{13}(t) \\
&= \lambda_{11} + \lambda_{12} + \lambda_{13} - \dot{\bar{D}}_{11}(t) - \dot{\bar{D}}_{12}(t) - \dot{\bar{D}}_{13}(t) \\
&= \lambda_{11} + \lambda_{12} + \lambda_{13} - 1 \\
&\leq 0
\end{aligned}$$

The above inequality holds because $\boldsymbol{\lambda}$ is doubly substochastic.

Case 21. (This is the only remaining case.) Note that this case can be tied to the 6-ring network graph, as in Figure 6-2. The following are the switch configurations (equivalently, link activations) employed by a maximal scheduler when the following VOQ's have dominant backlogs: $\mathrm{VOQ}_{11}, \mathrm{VOQ}_{12}, \mathrm{VOQ}_{22}, \mathrm{VOQ}_{23}, \mathrm{VOQ}_{33}, \mathrm{VOQ}_{31}$.

$$\begin{gathered}
\pi^1 = \{(1,1),(2,2),(3,3)\}, \quad \pi^2 = \{(1,2),(2,3),(3,1)\}, \\
\pi^3 = \{(1,1),(2,3)\}, \quad \pi^4 = \{(1,2),(3,3)\}, \quad \pi^5 = \{(2,2),(3,1)\}
\end{gathered}$$

Denote by $\mathbf{S}^i$ the single-hop service configuration matrix corresponding to link activation set π^i for $i = 1, \ldots, 5$. Then $\sum_{i=1}^{5} \dot{\bar{F}}_{\mathbf{S}^i}(t) = 1$. Consequently, we must have

$$\begin{aligned}
&\dot{\bar{D}}_{11}(t) = \dot{\bar{F}}_{\mathbf{S}^1} + \dot{\bar{F}}_{\mathbf{S}^3}, \quad \dot{\bar{D}}_{12}(t) = \dot{\bar{F}}_{\mathbf{S}^2} + \dot{\bar{F}}_{\mathbf{S}^4}, \quad \dot{\bar{D}}_{22}(t) = \dot{\bar{F}}_{\mathbf{S}^1} + \dot{\bar{F}}_{\mathbf{S}^5}, \\
&\dot{\bar{D}}_{23}(t) = \dot{\bar{F}}_{\mathbf{S}^2} + \dot{\bar{F}}_{\mathbf{S}^3}, \quad \dot{\bar{D}}_{33}(t) = \dot{\bar{F}}_{\mathbf{S}^1} + \dot{\bar{F}}_{\mathbf{S}^4}, \quad \dot{\bar{D}}_{31}(t) = \dot{\bar{F}}_{\mathbf{S}^2} + \dot{\bar{F}}_{\mathbf{S}^5}.
\end{aligned} \tag{6.25}$$

Since t is a regular time, and using $h(t) = \max_{ij} \bar{Q}_{ij}(t)$,

$$\begin{aligned}
\dot{h}(t) &= \lambda_{11} - \dot{\bar{F}}_{\mathbf{S}^1} - \dot{\bar{F}}_{\mathbf{S}^3} = \lambda_{12} - \dot{\bar{F}}_{\mathbf{S}^2} - \dot{\bar{F}}_{\mathbf{S}^4} = \lambda_{22} - \dot{\bar{F}}_{\mathbf{S}^1} - \dot{\bar{F}}_{\mathbf{S}^5} \\
&= \lambda_{23} - \dot{\bar{F}}_{\mathbf{S}^2} - \dot{\bar{F}}_{\mathbf{S}^3} = \lambda_{33} - \dot{\bar{F}}_{\mathbf{S}^1} - \dot{\bar{F}}_{\mathbf{S}^4} = \lambda_{31} - \dot{\bar{F}}_{\mathbf{S}^2} - \dot{\bar{F}}_{\mathbf{S}^5}
\end{aligned} \tag{6.26}$$

Straightforward algebraic manipulation of the above equations provides

$$\lambda_{11} - \lambda_{23} = \lambda_{22} - \lambda_{31} = \lambda_{33} - \lambda_{12}. \tag{6.27}$$

We obtain similar equalities by following the same procedure for each of the 6 distinct patterns that are isomorphic to Case 21. This provides each of the sets that make up $\boldsymbol{\Lambda}_3$ Note that while the doubly substochastic region $\boldsymbol{\Lambda}^*$ is full-dimensional in $\mathbb{R}_+^m$, equation (6.27) corresponds to the intersection of three hyperplanes, which has at most dimension $m - 3$. Thus, the set of arrival rates at which the configuration depicted as Case 21 must be considered, is lower-dimensional and has Lebesgue measure zero.

We have considered every possible case of VOQ configurations that realize the maximum $h(t)$. The only case for which the derivative $\dot{h}(t)$ could not be proved to be upper bounded by zero, was Case 21. We have demonstrated that Case 21 need only be considered for arrival rates belonging to $\boldsymbol{\Lambda}_3$, which is a set of Lebesgue measure zero in $\mathbb{R}_+^m$.

6.D Proof of Theorem 6.5.2

As in the proof of Theorem 6.5.1, suppose the following VOQ's have dominant backlogs at the regular time $t \geq 0$: $\mathrm{VOQ}_{11}, \mathrm{VOQ}_{12}, \mathrm{VOQ}_{22}, \mathrm{VOQ}_{23}, \mathrm{VOQ}_{33}, \mathrm{VOQ}_{34}, \mathrm{VOQ}_{44}, \mathrm{VOQ}_{41}$. The following matrix then represents the possible configurations employed by a maximal weight scheduler when these VOQ's are dominant. In particular, each column represents a valid maximal link activation. The rows of the matrix are ordered according to the above sequence of dominant VOQ's.

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

We can infer from matrix $\mathbf{M}$ that there are 10 maximal configurations corresponding to this set of dominant VOQ's. Let the service activation matrices corresponding to these configurations be labeled $\mathbf{S}^1, \ldots, \mathbf{S}^{10}$. Then $\sum_{i=1}^{10} \dot{\bar{F}}_{\mathbf{S}^i} = 1$, and similar to (6.25), we obtain

$$\begin{bmatrix} \dot{\bar{D}}_{11}(t) \\ \vdots \\ \dot{\bar{D}}_{41}(t) \end{bmatrix} = \mathbf{M} \begin{bmatrix} \dot{\bar{F}}_{\mathbf{S}^1} \\ \vdots \\ \dot{\bar{F}}_{\mathbf{S}^{10}} \end{bmatrix}.$$

Above, the matrix on the left has entries appearing in order of the VOQ's listed at the beginning of this appendix. Since t is a regular time, we have the following analogue to (6.26).

$$\dot{h}(t)\mathbf{e} = \begin{bmatrix} \lambda_{11} \\ \vdots \\ \lambda_{41} \end{bmatrix} - \mathbf{M} \begin{bmatrix} \dot{\bar{F}}_{\mathbf{S}^1} \\ \vdots \\ \dot{\bar{F}}_{\mathbf{S}^{10}} \end{bmatrix}$$

Thus, the set of arrival rates under which this 8-ring configuration of VOQ's is dominant is

$$\begin{aligned} \mathbf{\Lambda}_4 = \{\boldsymbol{\lambda} \geq 0 : \boldsymbol{\lambda} = \mathbf{M}\boldsymbol{\nu} + c\mathbf{e}, \quad & \mathbf{e}^T\boldsymbol{\nu} = 1, \\ \boldsymbol{\lambda} \leq \mathbf{M}\boldsymbol{\mu}, \quad & \mathbf{e}^T\boldsymbol{\mu} = 1, \\ \boldsymbol{\mu} \geq 0, \boldsymbol{\nu} \geq 0, c > 0\} & \end{aligned} \tag{6.28}$$

Above, the constraints $\boldsymbol{\lambda} \leq \mathbf{M}\boldsymbol{\mu}$, $\mathbf{e}^T\boldsymbol{\mu} = 1$ ensure that the arrival rates belong to the capacity region. We also require the constraint $c > 0$, because this forces us to discard arrival rates for which $\dot{h}(t)$ cannot be positive. Such cases cannot consequently lead to difficulty in concluding stability of maximal weight scheduling.

Consider the following quantities:

$$\tilde{\boldsymbol{\mu}} = 0.5 \times (1,0,0,0,1,0,0,0,0,0), \quad \tilde{\boldsymbol{\nu}} = 0.125 \times (0,1,1,1,0,1,1,1,1,1)$$

Then, we observe that $\mathbf{M}\tilde{\boldsymbol{\mu}} = \mathbf{M}\tilde{\boldsymbol{\nu}} + 0.125\mathbf{e}$. These quantities imply that $0.5\mathbf{e} \in \Lambda_4$. Furthermore, the value 0.125 corresponds to c in (6.28). Noting that c is a proxy for $\dot{h}(t)$, this implies that we cannot guarantee that $\dot{h}(t)$ is nonpositive when $\boldsymbol{\lambda} = 0.5\mathbf{e}$.

Consider the first constraint in (6.28): $\boldsymbol{\lambda} = \mathbf{M}\boldsymbol{\nu} + c\mathbf{e}$. We can add the constraint $\mathbf{e}^T\boldsymbol{\nu} = 1$ by using the equation $\nu_1 = 1 - \sum_{i=2}^{10} \nu_i$. Consequently,

$$\boldsymbol{\lambda} = \mathbf{M} \begin{bmatrix} 1 - \sum_{i=2}^{10} \nu_i \\ \nu_2 \\ \vdots \\ \nu_{10} \end{bmatrix} + c\mathbf{e},$$

which, through algebraic manipulation, can be expressed as

$$\boldsymbol{\lambda} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ -1 & -1 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ -1 & 0 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \nu_2 \\ \vdots \\ \nu_{10} \\ c \end{bmatrix} \qquad (6.29)$$

Note that the 8×10 matrix in the above equation (which we denote by $\mathbf{M}_{\mathrm{d}}$) has full rank[2]. Consequently, if the values $\nu_2, \ldots, \nu_{10}, c$ are *unconstrained*, they can be chosen to realize any desired value of $\boldsymbol{\lambda}$. In other words, they can be chosen to reach a non-negligible set of arrival rates in $\mathbb{R}^8_+$. Clearly however, these values are constrained in (6.28). Nevertheless, we next show that Λ_4 is non-negligible. Consider the values $\nu_2, \ldots, \nu_{10} = 0.1$ and $c = 0.05$ in (6.29), which provide $\boldsymbol{\lambda} = 0.45\mathbf{e}$. Clearly $\boldsymbol{\lambda} < 0.5\mathbf{e} = \mathbf{M}\tilde{\boldsymbol{\mu}}$, and we have that $\boldsymbol{\lambda} \in \Lambda_4$.

[2] The *rank* of a matrix equals the maximum number of linearly independent columns of the matrix. An $m \times n$ matrix is said to have *full rank* when its rank equals $\min\{m, n\}$. [11]

Now consider $c = 0.05 + \varepsilon_1$ and $\nu_i = 0.1 + \varepsilon_i$ for $i = 2, \ldots, 10$. In (6.29), these values yield

$$\boldsymbol{\lambda} = 0.45\mathbf{e} + \mathbf{M}_{\mathrm{d}} \begin{bmatrix} \varepsilon_2 \\ \vdots \\ \varepsilon_{10} \\ \varepsilon_1 \end{bmatrix}$$

Note that there exists $\varepsilon^* > 0$ such that when $|\varepsilon_i| < \varepsilon^*$ for all i, the following properties are maintained:

$$\boldsymbol{\lambda} < 0.5\mathbf{e}, \quad c > 0, \quad \sum_{i=2}^{10} \nu_i \leq 1.$$

These are exactly the conditions required to ensure $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_4$. Since $\mathbf{M}_{\mathrm{d}}$ has full rank, ranging over the set of values $\varepsilon_1, \ldots, \varepsilon_{10}$ whose absolute values are each less than ε^* must span a subset of $\mathbb{R}_+^8$ having nonzero Lebesgue measure. Since the stability properties provide no additional constraints concerning arrival rates of VOQ's that are not dominant, the set of arrival rates under which rate stability cannot be guaranteed must have nonzero Lebesgue measure in $\mathbb{R}_+^{n_1 \times n_2}$.

Chapter 7

Enabling distributed throughput maximization in wireless mesh networks: A partitioning approach

The consideration of simple *maximal weight* scheduling algorithms in Chapter 6 focused on the case of bipartite network graphs. In this chapter, we seek to broaden our understanding of the impact of maximal weight scheduling to more general network structures. In particular, we wish to understand the network structures under which maximal weight scheduling algorithms maximize throughput. Since maximal weight scheduling is highly conducive to being implemented in a decentralized fashion, we focus our attention on wireless networks, where decentralized control is particularly important. For wireless mesh networks, we develop network partitioning algorithms to enable efficient decentralized scheduling algorithms to achieve maximum throughput.

7.1 Overview and contributions

Wireless Mesh Networks (WMNs) have recently emerged as a solution for providing last-mile Internet access [7]. Several such networks are already in use, including testbeds and commercial deployments. A WMN consists of mesh routers, that form the network backbone, and mesh clients. Mesh routers are rarely mobile and usually do not have power constraints. The mesh routers are usually equipped with multiple wireless interfaces operating in orthogonal channels. Therefore, a major challenge in the design and operation of such networks is to allocate channels and schedule transmissions to efficiently share the common spectrum among the mesh routers. Several recent works focused on *multi-radio multi-channel* WMNs (e.g. [1,8,77,122]). Specifically, [8,122] study the issues of channel allocation, scheduling, and routing in WMNs, assuming that the traffic statistics are given. In this chapter, we study the issues of channel allocation and scheduling but unlike most previous works, we *do not* assume that the traffic statistics are known. Alternatively, we assume a *stochastic arrival process* and present a novel approach that enables throughput maximization by distributed scheduling algorithms.

Joint scheduling and routing in a slotted multihop wireless network with a stochastic packet arrival process was considered in the seminal paper by Tassiulas and Ephremides [150]. In that paper they presented the first *centralized* policy that is guaranteed to stabilize the network (i.e. provide 100% throughput) whenever the arrival rates are within the stability region. The results of [150] have been extended to various settings of wireless networks and input-queued switches (e.g. [6,97,111], and references therein). However, optimal algorithms based on [150] require repeatedly solving a *global optimization problem*, taking into account the queue backlog information for every link in the network. Obtaining a centralized solution to such a problem in a wireless network does not seem to be feasible, due to the communication overhead associated with continuously collecting the queue backlog information, and due to the limited processing capability available to the nodes. On the other hand, distributed algorithms usually provide only approximate solutions, resulting in significantly reduced throughput.

In this chapter, we show that *the multi-radio and multi-channel capabilities of WMNs provide an opportunity for simple deterministic distributed algorithms to obtain 100% throughput.* Mesh routers are usually equipped with multiple radios (transceivers) and can transmit and receive on multiple channels simultaneously [1, 8, 77]. Hence, channels have to be allocated to the links and the transmissions on each link have to be scheduled to avoid collisions. By allocating different channels to different links, several non-interfering subnetworks can be constructed. We study which subnetwork topologies enable simple distributed scheduling algorithms to achieve 100% throughput. Based on these results, we develop network partitioning algorithms that decompose the network into such subnetworks.

Although in *arbitrary topologies* the worst case performance of simple distributed maximal scheduling algorithms can be very low, there are some topologies in which they *can achieve 100% throughput.* This observation is based on a recent theoretical work by Dimakis and Walrand [51] in which they study the performance of the Longest Queue First (LQF) scheduling algorithm in a graph of interfering queues[1]. The LQF algorithm is a greedy maximal weight scheduling algorithm that selects the set of served queues greedily according to the queue lengths. We note that unlike a *maximum* weight (i.e. optimal) solution a *maximal* weight solution can be easily obtained in a distributed manner. Dimakis and Walrand [51] present sufficient conditions for a maximal weight algorithm to provide 100% throughput. These conditions are referred to as *Local Pooling* (LoP) and are related to the properties of all maximal independent sets in the conflict graph.

In this chapter we conduct the first thorough study of the implications of the LoP conditions on the network performance. We start by presenting a motivating example demonstrating that channel allocation algorithms that take into account LoP can enable distributed throughput maximization while increasing the overall capacity. We then conduct an extensive numerical study of the satisfaction of LoP by conflict graphs of up to 7 nodes. We show that *out of 1,252 graphs, only 14 do not satisfy LoP.* It is an indication of the strength of maximal weight scheduling for achieving 100% throughput regardless of the network topology, aside from a few "bad" topologies. Due to computational limitations,

[1] A graph of interfering queues can be constructed from the network graph according to the interference constraints and is usually referred to as an interference or conflict graph [72].

exhaustively verifying the satisfaction of LoP in graphs with more than 7 nodes seems infeasible. In order to be able to utilize larger graphs, we study what general properties of conflict graphs assist or hinder the LoP conditions. For example, we show that cliques (complete graphs) that are connected to each other in different manners satisfy LoP. On the other hand, we show that all n node ring graphs (with $n \geq 8$) do not satisfy LoP.

These observations provide several building blocks for partitioning a graph into subgraphs satisfying LoP. In order to demonstrate this capability and for the ease of presentation, we focus on scheduling under primary interference constraints[2] (studied in [36, 39, 104, 136, 160, 164, 169]). For example, we show that a tree network graph, when subject to the primary interference constraints, yields an interference graph which satisfies LoP. Hence, *in such a tree, maximal weight algorithms achieve 100% throughput.* We also study bipartite network graphs that provide insights regarding the number of required subgraphs. For instance, we show that in any $K_{2,n}$ bipartite graph (i.e. a $2 \times n$ input-queued switch) maximal weight matching algorithms achieve 100% throughput.

Building upon our observations, we design channel allocation algorithms. Similarly to [8] and to the static channel assignment in [77], we assume that a channel is assigned to a radio interface for an extended period of time. Under this assumption, using the minimum number of channels requires a partitioning of the network into the minimum number of subnetworks satisfying LoP. The general LoP conditions are extremely challenging to incorporate into a channel allocation algorithm. Fortunately, our study provides some useful building blocks. Since tree network graphs satisfy LoP, a possible approach (which we pursue) is to partition the network into non-overlapping forests, such that each edge will be part of a single forest and each forest will use a different channel. This problem is closely related to the *matroid intersection* and *matroid partitioning* problems.

Given k channels, the problem of partitioning the graph into k forests such that the number of edges included in the forests is maximized is referred to as the k-forest problem [59]. A simple approach is to obtain an *approximate* solution by a Breadth First Search (BFS) algorithm. Alternatively, since the k-forest problem is actually a specific case of a Matroid Cardinality Intersection problem, an *optimal* solution can be found by the Matroid Cardinality Intersection (MCI) algorithm of [85] (having polynomial complexity). We show that the MCI algorithm can be adapted to take into account the scenario in which different nodes have different numbers of radios. Using either the BFS algorithm or the MCI algorithm enables a simple distributed scheduling algorithm to achieve the capacity region (i.e. achieve 100% throughput). Yet, the capacity region itself may not be the best possible. This results from the *undesirable property* that the sizes (number of edges) of the forests are unbalanced. Therefore, and since the capacity of the largest forest may be significantly lower than the capacity of the smallest forest, the network capacity may be affected.

We present three algorithms that aim to expand the capacity region, while maintaining the LoP conditions in all the subnetworks. The main objective is to balance the number of edges across channels and to reduce the node degrees in each channel. Two of these novel capacity expansion algorithms make use of augmenting paths (in the spirit of the MCI

[2] The approach can be extended to more realistic interference constraints and to joint routing and scheduling.

algorithm of [85]) to balance the node degree across channels. Thus, they can be viewed as *balanced* Matroid Cardinality Intersection algorithms. We evaluate the performance of the algorithms via simulation. We show that the MCI algorithm significantly outperforms the BFS algorithms. We also compare the performance of the capacity expansion algorithms and the MCI algorithm and show that a large capacity improvement can be gained by using these algorithms. We conclude by exploring the tradeoffs between the capacities and the algorithms' complexities.

The main contributions of this chapter are two-fold. First, we conduct a rigorous study of the properties of network graphs satisfying Local Pooling. The second contribution is the development of network partitioning (i.e. channel allocation) algorithms that generate subnetworks with large capacity regions, while enabling distributed throughput maximization in each of the subnetworks.

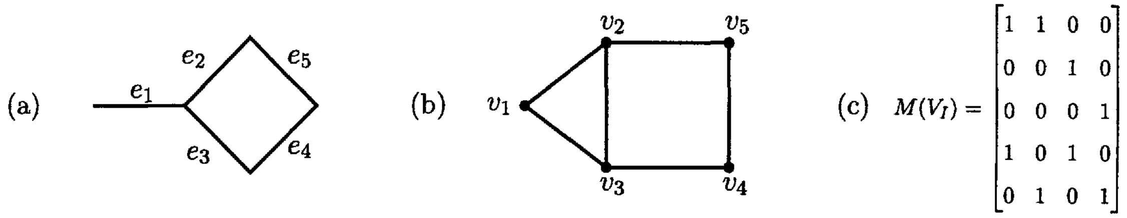
To the best of our knowledge, this is the first attempt to study the algorithmic implications of Local Pooling. This work is not only different from previous works on distributed stability, due to the focus on partitioning mesh networks, but also different from previous works on optimizing mesh networks that mostly rely on traffic statistics.

7.2 Model

We consider the backbone of a Wireless Mesh Network modeled by an *undirected network graph* $G_N = (V, E_N)$, where $V = \{1, \ldots, n\}$ is the set of nodes (mesh routers) and $E_N \subseteq \{(i,j) : i,j \in V\}$ is the set of bi-directional links, with $m \triangleq |E_N|$. Depending on the context, we denote a link either by (i,j) or by e_k. Note that unlike the scenario studied for switching and optical network scenarios in previous chapters, G_N need not be a complete graph.

Different wireless technologies pose different constraints on the set of transmissions that can take place simultaneously. For example, under *primary interference constraints*, the set of possible transmissions is the set of all possible matchings on G_N. More generally, in many cases an *interference graph* (also known as a conflict graph) $G_I = (V_I, E_I)$ can be defined based on the network graph G_N [72]. We assign $V_I \triangleq E_N$. Thus, each edge e_i in the network graph is represented by a vertex v_i of the interference graph and an edge (v_i, v_j) in the interference graph indicates a conflict between network graph links e_i and e_j (i.e. transmissions on e_i and e_j cannot take place simultaneously). In graph theoretic terminology, the interference graph resulting from primary interference constraints is called a *line graph* [69]. For example, Figure 7-1 illustrates a network graph and the corresponding interference graph under primary interference constraints (i.e. the line graph corresponding to the network graph). We note that the model can be easily generalized to capture network graphs with directional links. In such a case, link (i,j) may interfere with different links than those link (j,i) interferes with. Accordingly, the interference graph will include a node for each directional link.

We consider the application of Local Pooling to multi-radio multi-channel WMNs. Following the model of [8], we assume that each node v is equipped with $R(v)$ interfaces (radios). There are k available orthogonal channels and it is assumed that each of the $R(v)$



$$M(V_I) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Figure 7-1: (a) A network graph G_N, (b) the corresponding interference graph G_I under the primary interference constraints, and (c) the matrix $M(V_I)$ of maximal independent sets in G_I.

interfaces operates on a different channel. Similarly to [8] and to the static model of [77], we consider a static channel allocation model in which a channel is allocated to each interface for an extended period of time. Such an approach enables the use of commodity 802.11 radios [8]. We note that the extension of the model for a dynamic channel allocation is a subject for further research. We assume that transmissions in different channels cannot collide. Therefore, once the different channels are allocated, k disjoint interference graphs are generated.

For simplicity of presentation, we consider single-hop bi-directional traffic. Under this assumption, the joint routing and scheduling problem reduces to a scheduling problem. This is why G_N can be treated as an undirected graph. Consequently, under the general model of Chapter 2, we restrict the available service activations to the single-hop service activation set, $\mathcal{S}^{\text{sh}}$. Naturally, this implies that the network capacity region of arrival rates is the single-hop region Λ^*_{sh}. As mentioned above, the model can be extended to more general scenarios. In this wireless setting, which remains the focus of the remainder of this thesis, Π_N denotes the set of all feasible link activations in the network graph G_N, where $\pi = (\pi_{ij}, (i,j) \in E_N) \in \Pi_N$ is a $(0,1)$ column vector representing a possible link activation. Under primary interference constraints, Π_N includes all possible matchings, while in general, it corresponds to all independent sets in the interference graph G_I. Following the notation of [51], we denote by $\mathbf{M}(V_I)$ the matrix that includes all the *maximal* independent sets in G_I (i.e. all the maximal elements of Π_N). For example, Figure 7-1(c) shows the matrix $\mathbf{M}(V_I)$ for the interference graph G_I in Figure 7-1(b).

Given the above model, Algorithm 1 (the algorithm of Tassiulas and Ephremides [150]) is throughput optimal. However, the algorithm must find the *maximum weight independent set* in G_I at each time slot. Namely, it has to solve an NP-Complete problem in every time slot. In the context of switch scheduling and primary interference constraints, this algorithm has to schedule the edges of the *Maximum Weight Matching* at each time slot, where the edge weights are the queue sizes. The maximum weight matching in any graph can be found in $O(n^3)$ computation time, using a centralized algorithm [85]. However in wireless networks, implementing a centralized algorithm is not feasible and distributed algorithms (e.g. [71]) can obtain only an approximate solution, resulting in a fractional throughput. Hence, even under very simple transmission constraints, it is difficult to obtain 100% throughput in a distributed manner. This motivates us to develop channel allocation methods that will enable simple distributed scheduling algorithms to obtain 100% throughput. Therefore, we provide a general definition of the *Channel Allocation Problem* below. In Section 7.5 we will develop algorithms for *specific* versions of this problem.

Definition 7.2.1 (Static Channel Allocation Problem) *Given a network graph G_N, k channels, and $R(v)$ radios at each node $v \in V$, assign channels to links (i,j) $\forall (i,j) \in E_N$ such that at most $R(v)$ channels are used by links adjacent to v and* simple *(e.g. greedy) distributed algorithms are stable in each subnetwork operating in a different channel.*

Observe that the Static Channel Allocation Problem *binds* each network link to a single (or possibly multiple) channels, and maintains that allocation for an extended period of time. This is in contrast to a network that employs *Dynamic Channel Allocation*, where the channel(s) assigned to each link can vary dynamically with time, possibly in response to traffic variations. Dynamic channel allocation can be enabled at the expense of additional negotiation between network elements [77]. Under static channel allocation, a network node must only decide which link(s) it will send packets across, since the channel associated with each link is fixed.

Since the configurations available to a network employing dynamic channel allocation subsume those of a network employing static channel allocation, one would expect that the throughput achievable under dynamic channel allocation is always equal or greater than that under static channel allocation. Under *maximum* weight scheduling, this must indeed be the case by Theorem 2.3.1, since the set of service activation matrices $\mathcal{S}_{\text{static}} \subseteq \mathcal{S}_{\text{dynamic}}$. However, as we will show in Section 7.3.2, there exist specific instances where a distributed *maximal* weight scheduler suffers throughput loss under dynamic channel allocation next to a properly configured network employing static channel allocation. For general mesh networks, our studies of random networks in Section 7.6 demonstrate that dynamic channel allocation does indeed achieve equal or better throughput than static allocation in most instances.

We stress that direct comparison of throughput performance between a network employing static or dynamic channel allocation cannot be considered fair, since a network employing dynamic channel allocation is in essence significantly *more capable* than a network employing static allocation. Nevertheless, the relative performance of static versus dynamic channel allocation is of interest, and we attempt to quantify it in our numerical studies.

7.2.1 Extensions of the network model

The focus of the wireless chapters of this thesis will be exclusively on wireless networks in which a well-defined interference (or conflict) graph exists. Recent studies, notably [105, 144], suggest that a more appropriate model would have simultaneous communication constrained by signal-to-interference-plus-noise-ratio (SINR). In an SINR-constrained network, a link activation π can be used for communication if the SINR at each receiving node exceeds a certain threshold. Consequently, there is a well-defined link activation set Π_N and service activation set $\mathcal{S}$ under an SINR-constrained system, which implies that such a model remains within the framework of Tassiulas and Ephremides [150] (see [110, 111] for early applications of this connection). However, since there is no interference graph in this setting, we cannot apply scheduling algorithms for finding maximal weight independent sets of an interference graph. Nevertheless, the Local Pooling principle can be extended to an SINR-constrained system. We discuss this possibility in the conclusions at the end of the chapter.

Another extension of the wireless network model is to links having nonzero probability of transmission failure. This is also a feature of the Tassiulas and Ephremides framework [150], and can easily be incorporated into our model.

7.3 Local Pooling conditions

7.3.1 Definitions

In this section we restate the definition and implications of Local Pooling (LoP) presented in [51]. We also present and demonstrate a somewhat simpler set of definitions. Recall that $\mathbf{M}(V_I)$ is the collection of maximal independent vertex sets on G_I, organized as a matrix (an example appears in Figure 7-1). Denote by conv($\mathbf{M}$) the convex hull of the columns of matrix $\mathbf{M}$. We now provide the definition of LoP from [51][3].

Definition 7.3.1 (Local Pooling - LoP [51]) *The set of nodes (queues) $V \subseteq V_I$ satisfies local pooling, if there exists a nonzero vector $\boldsymbol{\alpha} \in \mathbb{R}_+^{|V|}$ such that $\boldsymbol{\alpha}^T \boldsymbol{\phi}$ is a positive constant for all $\boldsymbol{\phi} \in \text{conv}(\mathbf{M}(V))$. Local pooling is satisfied, if every $V \subseteq V_I$ satisfies local pooling.*

In this chapter, we separate the definition of Local Pooling to two different definitions and present a somewhat simpler definition for the satisfaction of LoP by a set of nodes. We show that although this definition does not take into account the convex hull of $\mathbf{M}$, it is equivalent to the definition in [51]. Recall that $\mathbf{e}$ represents a vector having each entry equal to unity. We deliberately avoid specifying its size, because it will be obvious by the context of its use.

Definition 7.3.2 (Subgraph Local Pooling - SLoP) *An inter-ference graph G_I satisfies Subgraph Local Pooling, if there exists $\boldsymbol{\alpha} \in \mathbb{R}_+^m$ and $c > 0$ such that $\boldsymbol{\alpha}^T \mathbf{M}(V_I) = c\mathbf{e}^T$.*

Lemma 7.3.1 *The definition of Subgraph Local Pooling and the satisfaction of Local Pooling by a set of nodes (Definition 7.3.1) are equivalent.*

Proof: Suppose the set of nodes $V \subseteq V_I$ satisfies local pooling as defined in Definition 7.3.1. Then, there exists $c > 0$ and $\boldsymbol{\alpha} \in \mathbb{R}_+^{|V|}$ such that $\boldsymbol{\alpha}^T \boldsymbol{\phi} = c$ for all $\boldsymbol{\phi} \in \text{conv}(\mathbf{M}(V))$. Clearly each column of $\mathbf{M}(V)$ belongs to conv($\mathbf{M}(V)$), which gives $\boldsymbol{\alpha}^T \mathbf{M}(V) = c\mathbf{e}^T$. Thus the subgraph of G_I over nodeset V satisfies SLoP. Conversely, suppose that the subgraph of G_I over nodeset V satisfies SLoP. Then there exist $c > 0$ and $\boldsymbol{\alpha} \in \mathbb{R}_+^{|V|}$ such that $\boldsymbol{\alpha}^T \mathbf{M}(V) = c\mathbf{e}^T$. Now consider $\boldsymbol{\phi} \in \text{conv}(\mathbf{M}(V))$, which must equal by definition $\mathbf{M}(V)\boldsymbol{\beta}$ for $\boldsymbol{\beta} \in \mathbb{R}_+^{|\mathbf{M}(V)|}$ with $\mathbf{e}^T \boldsymbol{\beta} = \sum_j \beta_j = 1$, $\beta_j \geq 0$, $\forall j$ and $|\mathbf{M}(V)|$ equal to the number of columns in $\mathbf{M}(V)$. Then, we have $\boldsymbol{\alpha}^T \boldsymbol{\phi} = \boldsymbol{\alpha} \mathbf{M}(V) \boldsymbol{\beta} = c\mathbf{e}^T \boldsymbol{\beta} = c$. Note that this value is

[3] This statement of the LoP conditions can be weakened if certain restrictions are made on the arrival processes [51].

constant regardless of the choice of ϕ. Thus, the set of nodes V satisfies local pooling as defined in Definition 7.3.1. ■

We can now define the notion of Overall Local Pooling which requires that Subgraph Local Pooling (SLoP) will be satisfied in any subgraph of a given interference graph induced by selecting a *subset of the nodes.*

Definition 7.3.3 (Overall Local Pooling - OLoP) *Interference graph G_I satisfies Overall Local Pooling if each induced subgraph over the nodes $V \subseteq V_I$ satisfies SLoP.*

We continue with the example of the interference graph G_I and the corresponding matrix $\mathbf{M}(V_I)$ depicted in Figure 7-1. We can see that G_I satisfies SLoP since for $\boldsymbol{\alpha} = (1,1,1,1,1)$, $\boldsymbol{\alpha}^T\mathbf{M}(V_I) = 2\mathbf{e}^T$. Similarly, the subgraph composed of the vertex set $\{2,3,4\}$ satisfies SLoP, since for $\boldsymbol{\alpha} = (1,1,0)$, $\boldsymbol{\alpha}^T\mathbf{M}(\{2,3,4\}) = \mathbf{e}^T$. It can be shown that all subgraphs of G_I satisfy SLoP, and therefore, G_I satisfies OLoP.

We can now describe the stability of the system when the service in each time slot is scheduled according to the Longest Queue First (LQF) algorithm. This algorithm is an iterative greedy algorithm that selects the node of G_I with the longest queue, and removes it and its neighbors from the interference graph. This process is repeated successively until no nodes remain in the graph. When two queues have the same length a tie-breaking rule has to be applied. The set of selected nodes is a maximal independent set in the interference graph. Hence, since the nodes are selected according to their weights, we will refer to the LQF algorithm as the Maximal Weight Independent Set algorithm. Such a greedy algorithm can be easily implemented in a distributed manner. In [51] the following theorem is proved:

Theorem 7.3.1 (Dimakis and Walrand, 2006 [51]) *If interference graph G_I satisfies the OLoP conditions, a Maximal Weight Independent Set scheduling algorithm achieves 100% throughput.*

To conclude, the satisfaction of OLoP by an interference graph is a *sufficient* condition for distributed maximal weight algorithm to be throughput optimal (i.e. in that case, there is no need to obtain an optimal solution to (2.10) in each slot).

7.3.2 Channel allocation example

The following simple example demonstrates the application of the LoP conditions, presented above, to a channel allocation (network partitioning) problem. We consider the 6-node ring network graph, depicted on the left in Figure 7-2. Under the primary interference constraints, this graph has a corresponding 6-node ring interference graph representation, which is illustrated on the right in Figure 7-2. Under primary interference constraints, the maximal weight independent set in the interference graph is equivalent to the maximal weight matching in the network graph. A maximal weight matching can be obtained in a distributed manner by the greedy algorithm of Hoepman [71].

If a single radio is located at each node of the 6-node ring illustrated in Figure 7-2(a), then no two adjacent edges can be simultaneously active. The capacity region $\boldsymbol{\Lambda}^*$ is then

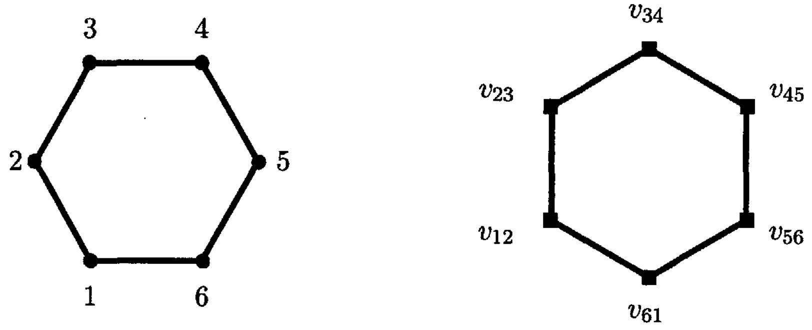


Figure 7-2: A 6-node ring network graph and its interference graph.

characterized by the following inequalities:

$$\lambda_{12} + \lambda_{23} \leq b,\ \lambda_{23} + \lambda_{34} \leq b,\ \lambda_{34} + \lambda_{45} \leq b,$$
$$\lambda_{45} + \lambda_{56} \leq b,\ \lambda_{56} + \lambda_{61} \leq b,\ \lambda_{61} + \lambda_{12} \leq b, \tag{7.1}$$

where $b = 1$. This capacity region can be achieved by a centralized algorithm that finds a maximum weight matching (i.e. obtains the optimal solution to (2.10)) in each time slot.

It was shown in [51] that in the 6-node ring, OLoP does not hold, and that in general a *maximal* weight matching algorithm does not achieve 100% throughput in the 6-node ring[4]. According to [90], a *maximal* weight matching algorithm can only guarantee stability for arrival rates that are 50% of the rates in the region above (Λ^*). Hence, the guaranteed distributedly achievable region is given by (7.1) with $b = 0.5$.

If we allow two channels to be used simultaneously, and provide two transceivers to each node, then in every time slot a node can transmit two packets on the selected link (similarly to a speedup of two, defined in [49]). Thus, the guaranteed achievable region (using maximal weight matching) is again given by (7.1) with $b = 1$.

Alternatively, links $(1, 2), (2, 3)$, and $(3, 4)$ can use one channel, while the remaining links use the other channel. The interference graph on each channel is now a tree (e.g. the line connecting v_{12}, v_{23}, and v_{34}). Since [51] shows that the maximal weight independent set algorithm is throughput optimal in *tree* interference graphs, the *distributedly achievable* stability region is now given by

$$\lambda_{12} + \lambda_{23} \leq 1,\ \lambda_{23} + \lambda_{34} \leq 1,$$
$$\lambda_{45} + \lambda_{56} \leq 1,\ \lambda_{56} + \lambda_{61} \leq 1. \tag{7.2}$$

This provides a strict performance improvement over the region achievable by using two channels (speedup of two) in the interference graph represented in 7-2(b). Yet, it is clear that this channel allocation is not the best possible: the allocation in which links $(1, 2), (3, 4)$, and $(5, 6)$ use one channel, while the remaining links use the other channel can provide each network link with a stable rate of one unit per time slot (i.e. $\lambda_{ij} \leq 1\ \forall (i, j) \in E_N$).

[4] In [51], it was shown that under *restricted* arrival processes (subject to a variance constraint and a large deviation bound), a maximal weight matching algorithm is stable in the 6-node ring. In this work the arrival processes are not restricted in this way.

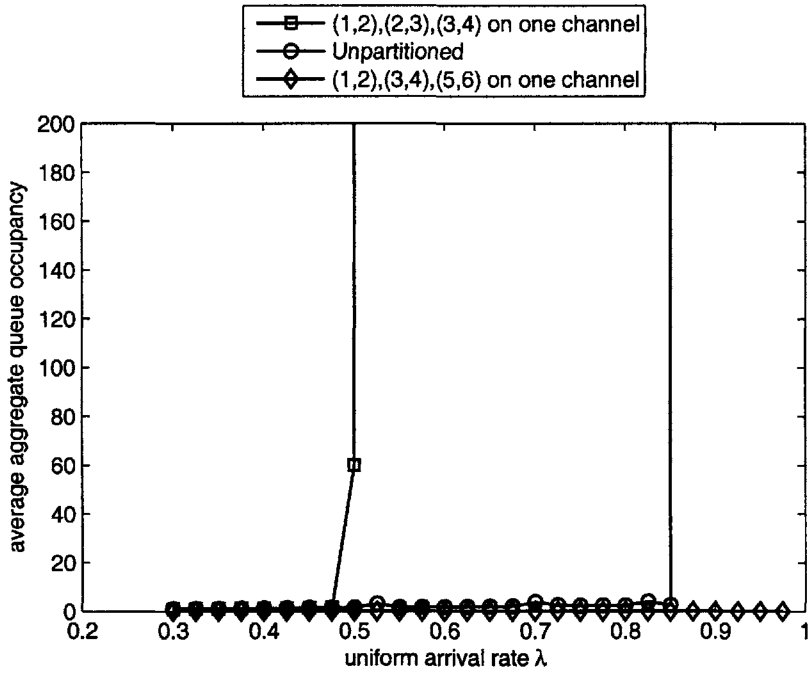


Figure 7-3: Average aggregate queue backlog as a function of uniform arrival rate λ, for different partitioning strategies, when each strategy is used in conjunction with maximal weight scheduling.

To supplement the above theoretical discussion, we have conducted numerical simulations of the performance of the 6-ring under the different partitioning strategies. Our first simulation subjected the network to deterministic fluid arrivals, with each edge having a total load of λ packets per slot. In Figure 7-3, the average aggregate queue backlog is plotted as a function of λ. Observe that the first proposed partitioning scheme (links $(1,2),(2,3),(3,4)$ on one channel, and the remaining links on the other channel) becomes unstable at $\lambda = 0.5$. For the case of the unpartitioned network, we observe that the network achieves approximately 85% throughput, which is better than the 50% lower bound we have quoted above. The improved partitioning scheme ($(1,2),(3,4),(5,6)$ on one channel, and the remaining links on the other) does not destabilize below $\lambda = 1$, and in fact maintains essentially empty queues at all times. Figure 7-3 clearly demonstrates that the unpartitioned network suffers throughput loss next to the well-partitioned network.

We have additionally considered the same network, subject to Poisson arrivals. Although the result of [51] implies that the unpartitioned network will not suffer throughput loss under this arrival process, we observe that the system does suffer a significant degradation in delay performance relative to the well-partitioned network. In Figure 7-4, we plot the aggregate queue backlog in the network as a function of time, for the unpartitioned and well-partitioned networks, when each edge has a load of $\lambda = 0.98$ packets per slot. The unpartitioned network suffers significant variations in aggregate backlog, while the partitioned network maintains a relatively steady level of backlog. This suggests that decoupling

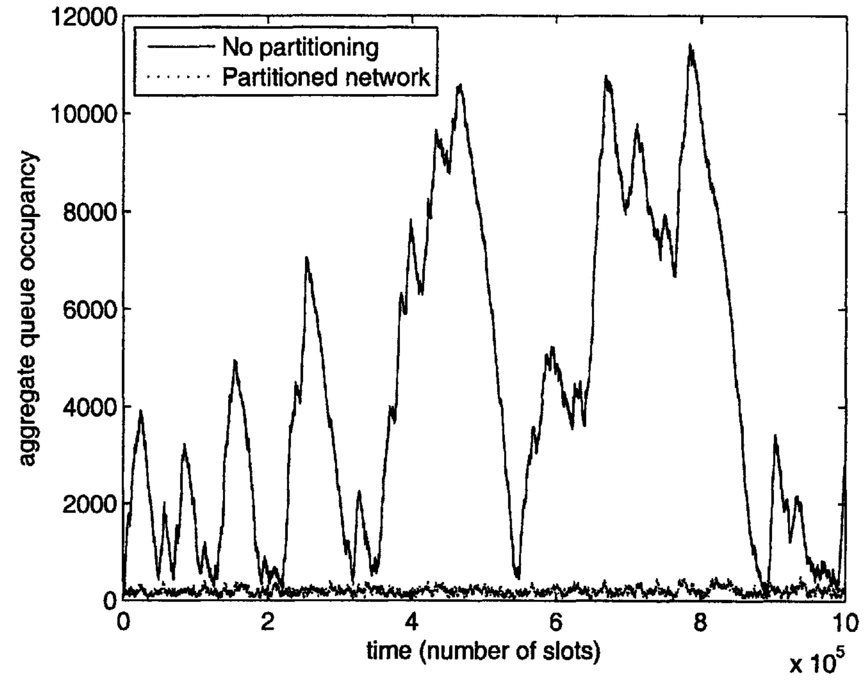


Figure 7-4: Sample paths of aggregate queue backlog for the unpartitioned and well-partitioned 6-ring under Poisson arrivals.

neighboring edges through partitioning can lead to beneficial delay performance properties. Figure 7-5 shows the average aggregate backlog as a function of uniform arrival rate λ for the partitioned and unpartitioned networks.

For a general network operating under primary interference constraints with a speedup of two (similar to allocating two channels to each link), a greedy maximal weight algorithm (implementable in a distributed manner) can achieve the network stability region Λ^* [90]. Our example above shows for a particular network scenario that when two channels are allocated such that each component satisfies OLoP, the stability region (that can be *achieved* by a distributed algorithm) is *strictly larger* than the original stability region Λ^*. The following lemma shows that such a strict performance improvement can be obtained in any network with primary interference constraints that can be partitioned into two non-trivial components satisfying OLoP.

Lemma 7.3.2 *Under primary interference constraints, if a network G_N can be partitioned into two subnetworks G_N^1, G_N^2 satisfying OLoP, the distributedly achievable joint stability region of G_N^1 and G_N^2 is strictly larger than the stability region of G_N (achievable distributedly by a speedup of two).*

Proof: See Appendix 7.A. ■

The above example demonstrates that careful channel allocation taking into account topologies that satisfy OLoP can provide provable and significant improvements over arbitrary channel allocation. Moreover, it shows that partitioning into different OLoP-satisfying

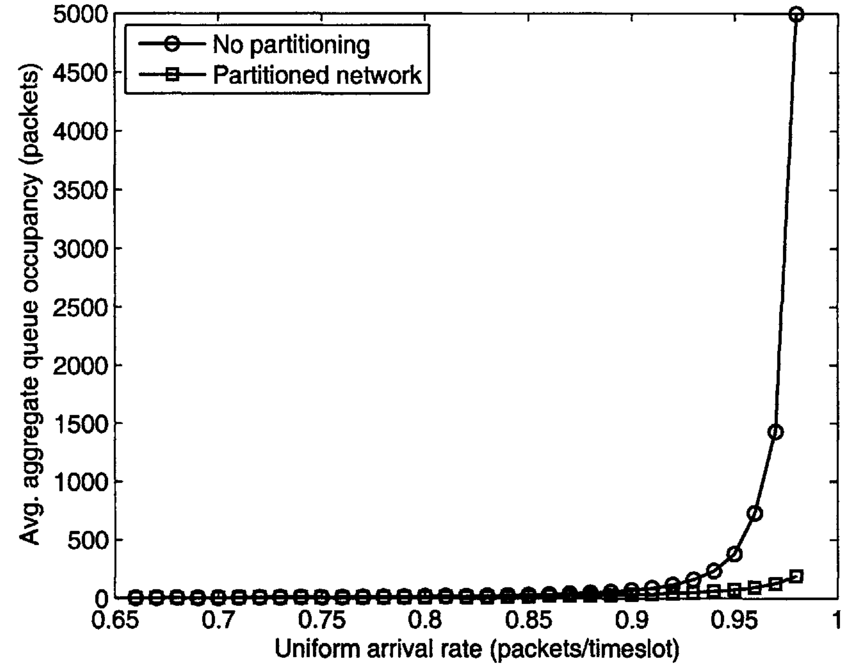


Figure 7-5: Average aggregate queue backlog as a function of uniform arrival rate λ, for the unpartitioned and well-partitioned 6-ring under Poisson arrivals.

components can result in different capacity regions. Thus, it provides the motivation to study the characteristics of network topologies satisfying OLoP and to design channel allocation algorithms that exploit such characteristics.

7.4 A study of Local Pooling

7.4.1 Exhaustive numerical search

We performed a numerical study in which we searched over all interference graphs of up to 7 nodes. We employed Mathematica to identify all simple graphs, and Matlab to determine the maximal configurations (i.e. to obtain the matrices $\mathbf{M}(V_I)$) and to verify the satisfaction of the OLoP conditions for each interference graph. The OLoP conditions are based on the SLoP conditions that were verified using the following linear program presented in [51].

$$
\begin{aligned}
c^* = \max_{c,\boldsymbol{\mu},\boldsymbol{\nu}} & \; c \\
\text{s.t. } & \mathbf{M}(V_I)\boldsymbol{\mu} \geq \mathbf{M}(V_I)\boldsymbol{\nu} + c\mathbf{e} \\
& \mathbf{e}^T\boldsymbol{\mu} = 1 \\
& \mathbf{e}^T\boldsymbol{\nu} = 1 \\
& \boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}_+^m \\
& c \in \mathbb{R}
\end{aligned}
$$

It has been shown in [51, Prop. 1] that the graph G_I satisfies SLoP if and only if $c^* = 0$.

In order to simplify the presentation of the numerical results, we first show that the OLoP conditions are satisfied by the disjoint union of two graphs (not sharing any vertices in common) satisfying the OLoP conditions. This allowed us to restrict our search to connected simple graphs.

Proposition 7.4.1 *A graph $G_I = G_I^1 \cup G_I^2$ (disjoint union) satisfies OLoP, if and only if G_I^1 and G_I^2 satisfy OLoP.*

Proof: Suppose G_I satisfies OLoP. Consider all induced subgraphs restricted to the vertices of G_I^1. Then, any such induced subgraph satisfies the SLoP conditions by our assumption that G_I satisfies OLoP. Thus, G_I^1 satisfies OLoP. The same reasoning provides that G_I^2 satisfies OLoP.

Suppose that G_I^1 and G_I^2 satisfy OLoP. Then, any induced subgraph of G_I can be split into disjoint induced subgraphs on G_I^1 and G_I^2. For the induced graph on G_I^1, our assumption provides that there exists nonzero $\boldsymbol{\alpha}_1 \geq 0$ that multiplies any maximal independent vector on the induced subgraph to yield a constant c_1. Similarly, there exists $\boldsymbol{\alpha}_2$ and c_2 for the induced subgraph on G_I^2. Every maximal independent set of the induced subgraph of G_I must be the disjoint union of a maximal independent set of the induced subgraph on G_I^1 and a maximal independent set of the induced subgraph on G_I^2. Thus, the augmented vector $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$ must yield a constant value of $c_1 + c_2$ for all maximal independent sets of the induced subgraph on G_I. ■

We note that in the following section we will present several additional theoretical results regarding LoP in general graphs. A specific case of one of the results that will be presented there (Lemma 7.4.1) is that graphs that have a node with degree 1 satisfy SLoP. This allowed us to restrict our search to graphs that do not have vertices of degree 1, thereby significantly reducing the computation time. We first considered all connected interference graphs having up to 5 vertices that do not have vertices of degree 1. There are 15 such graphs. We obtained the following numerical result.

Numerical Result 7.4.1 *All connected simple graphs of up to* 5 *nodes that do not have vertices of degree* 1 *satisfy SLoP.*

This immediately implies that all graphs having up to 5 vertices (there are 52 such graphs) satisfy OLoP. Next, we considered graphs of 6 vertices (there are 61 such connected graphs without degree 1) and obtained the following result.

Numerical Result 7.4.2 *All graphs of* 6 *vertices* except *the* 6*-node ring satisfy SLoP.*

Numerical Results 7.4.1 and 7.4.2 together imply that all graphs of up to 6 vertices except the 6-node ring satisfy OLoP.

Finally, we considered all graphs of 7 vertices. We first removed from consideration all such graphs having a 6-ring as an induced subgraph, since due to the failure of SLoP in a 6-ring, OLoP fails in these graphs by definition. There are 12 such graphs, and their general form is depicted in Figure 7-6(a). Among the remaining graphs of 7 vertices, we can then guarantee that there are no induced subgraphs, having 6 vertices or fewer, that fail the SLoP conditions.

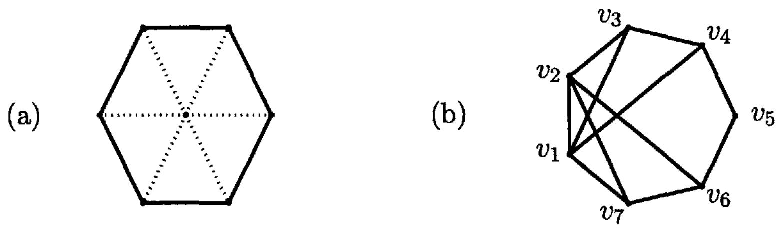


Figure 7-6: 7-node graphs that fail OLoP: (a) configurations where the induced graph over the outer 6 nodes is a 6-ring (the dotted lines indicate edges that can exist), and (b) the only 7-node graph that has no induced 6-ring subgraph and fails SLoP.

Numerical Result 7.4.3 *There is one graph of* 7 *vertices which does not have an induced* 6*-ring on any subset of* 6 *nodes that fails the SLoP conditions. This graph is depicted in Figure 7-6(b).*

To conclude, almost all 1,252 graphs of up to 7 nodes satisfy OLoP (specifically, 14 fail OLoP). All attempts at numerical evaluations for graphs of greater than 7 vertices suffered computational difficulty. Therefore, in the following section we focus on generating large graphs satisfying OLoP from small components.

7.4.2 Constructive approach

Our first observation is about connecting a graph and a clique (complete graph).

Lemma 7.4.1 *If G_I satisfies OLoP, then the graph G_I^*, which consists of G_I sharing a single vertex with clique $K_l, l \geq 2$, satisfies OLoP.*

Proof: Assume that G_I satisfies OLoP. Denote by v the vertex of G_I that is shared with clique K_l. We need only consider the induced subgraphs of G_I^* containing a vertex $v^* \neq v$ belonging to the clique K_l, since all other induced subgraphs are subgraphs of G_I and satisfy SLoP by our initial assumption. Clearly, the maximal independent sets of any such induced subgraph (whose vertex set is designated by V) either include vertex v or v^*, but never both vertices. Consequently, the vector $\boldsymbol{\alpha}$ having all zero entries except at the indices corresponding to vertices of K_l, where the entries are set to 1, yields $\boldsymbol{\alpha}^T\mathrm{M}(V) = \mathbf{e}^T$. Thus, such a subgraph satisfies SLoP. This holds for all induced subgraphs of G_I^* that include v^*, and we conclude that G_I^* satisfies OLoP. ■

From the proof of Lemma 7.4.1 it can be seen that a graph that has a node with degree 1 (such a graph can be viewed as a graph G_I sharing a node with K_2) satisfies SLoP. Recall that we have used this result in Section 7.4.1 to reduce the number of graphs in our numerical search. Moreover, the observation in [51] that any interference graph that is a tree (or forest) satisfies OLoP can be immediately obtained using Lemma 7.4.1. We note that in Section 7.4.3 we will show that even under the simple primary interference constraints, the only interference graph that can be a tree is a line. Therefore, we now study more complicated interference graphs.

Lemma 7.4.2 *Every complete graph satisfies OLoP.*

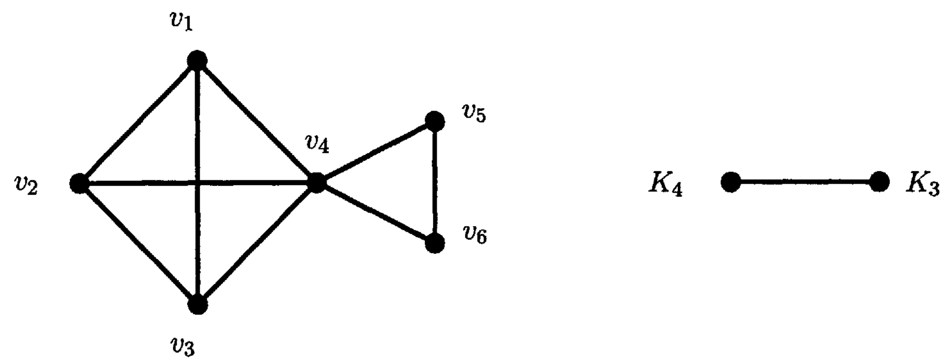


Figure 7-7: An interference graph composed of two cliques and the corresponding *tree of cliques* graph.

Proof: Consider the complete graph $G_I = K_l$. Then clearly any subset of the nodes of G_I, labeled V, also generates a complete induced subgraph. Each maximal independent set of a complete graph can only contain one vertex, from which we conclude that $\mathbf{M}(V)$ is the identity matrix of size $|V|$. Thus, we can use $\boldsymbol{\alpha} = \mathbf{e}$, which yields $\boldsymbol{\alpha}^T\mathbf{M}(V) = \mathbf{e}^T$ for any V, from which we conclude that every induced subgraph satisfies SLoP, and consequently that G_I satisfies OLoP. ■

We define a *tree of cliques* as follows (an example is provided in Figure 7-7) and derive the following Theorem.

Definition 7.4.1 *A* tree of cliques *is composed of cliques connected to each other in a tree structure. Its nodes can be equated to cliques and its edges imply a shared vertex between two adjacent cliques. No vertex can be shared by more than two adjacent cliques.*

Theorem 7.4.1 *A tree of cliques satisfies OLoP.*

Proof: Consider any clique G_I^1 on the tree. By Lemma 7.4.2 this clique satisfies OLoP. Then, consider any clique adjacent to G_I^1 in the tree of cliques, and denote the graph of the two combined cliques G_I^2. Since G_I^1 and the adjacent clique share only a single vertex, we can apply Lemma 7.4.1 to conclude that G_I^2 satisfies OLoP. By iteratively adding successive cliques to the overall graph under consideration, we see that each resulting graph must satisfy OLoP by Lemma 7.4.1. Thus, the overall tree of cliques must satisfy OLoP. ■

The next theorem considers cliques connected by disjoint edges, where no two connecting edges share any vertices in common. Consequently, at most $\min\{l_1, l_2\}$ edges can connect K_{l_1} and K_{l_2} while maintaining an overall simple graph. The proof considers four possible subgraph configurations and demonstrates SLoP for each type. The main idea is that each clique usually contributes a single vertex to every maximal independent set of each subgraph.

Theorem 7.4.2 *If two cliques are connected by any number of disjoint edges, the combined graph satisfies OLoP.*

Proof: See Appendix 7.B. ■

We now consider a generalized structure of the one defined in Definition 7.4.1, which we term "tree-of-blocks". Here, we generalize the types of structures that can correspond to each vertex of a tree. We have already shown that a clique is one such structure. We next show that two cliques connected by any number of disjoint edges is another such structure. As before, we require that two "blocks" can only share at most one vertex in common. The proof of the following theorem is along similar lines as the proof of Theorem 7.4.2.

Theorem 7.4.3 *A "tree-of-blocks", where each block is either a clique $K_l, l \geq 2$ or a pair of cliques K_{l_1}, K_{l_2}, $l_1, l_2 \geq 1$, connected by any number of disjoint edges, satisfies OLoP.*

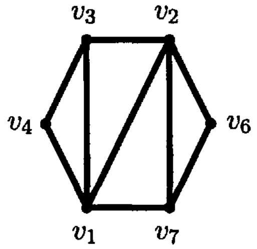
Proof: See Appendix 7.C. ■

7.4.3 Primary interference constraints

As mentioned above, the primary interference constraints yield an interference graph G_I which is the line graph of the network graph G_N. In this section, we study the restrictions imposed on such interference graphs. We begin by considering the only 7-node graph, which does not have an induced 6-ring, that failed SLoP (depicted in Figure 7-6(b)).

Proposition 7.4.2 *Under primary interference constraints, the interference graph presented in Figure 7-6(b) cannot correspond to any valid network graph.*

Proof: According to [69] a graph is a line graph, if and only if it does not contain any one of 9 specific induced subgraphs. In particular, the following graph is one of the 9 subgraphs, with vertices of Figure 7-6(b) labeled appropriately to show the correspondence.



■

We conclude that *only* the 6-ring leads to failure of the OLoP conditions in any network graph having 7 edges or fewer. By similar arguments, we can show that other interference graphs cannot exist under primary interference constraints. For example, we can show that there is no network graph whose interference graph (line graph) is a tree having a node degree greater or equal to 3. Any such tree has as an induced subgraph the complete bipartite graph $K_{1,3}$ (also known as the "claw"). According to [69], the existence of such an induced subgraph precludes the possibility that this interference graph is the line graph of any network graph.

Although there is no interference graph that is a tree, a network graph that is a tree can of course exist. It can be shown that the interference graph of such a network graph is

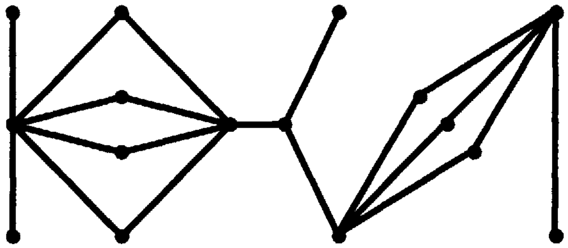


Figure 7-8: Example of a network graph whose interference graph satisfies OLoP.

always a tree of cliques, defined in Definition 7.4.1. The following corollary is an immediate result of Theorem 7.4.1. According to this corollary, *maximal weight matching algorithms are stable (provide 100% throughput) in trees.* To the best of our knowledge, this corollary provides the first non-trivial network structure in which simple distributed algorithms are stable. The channel allocation algorithms that will be presented in Section 7.5 are based on this observation.

Corollary 7.4.1 *Under primary interference constraints, the interference graph of a tree network graph satisfies OLoP.*

Based on the results presented in Section 7.4.2, we can construct other non-trivial networks in which maximal weight matching algorithms are stable. For example, Theorem 7.4.3 implies that the network described in Figure 7-8 satisfies OLoP, and thus is stable under distributed scheduling. Developing network partitioning algorithms that efficiently take advantage of such topologies is a subject for further research.

We have obtained additional results that concern bipartite graphs. Although mesh networks are usually not bipartite, bipartite graphs provide insight regarding the performance of our partitioning algorithms. Since input-queued switches are bipartite graphs with primary interference constraints, an additional byproduct is insight regarding switches. The following corollary generalizes the result of Chapter 6 (presented in [28]) regarding the 2×2 input-queued switch.

Corollary 7.4.2 *A maximal weight matching algorithm achieves* 100% *throughput in a* $K_{2,l}$ *bipartite graph (i.e. in a* $2 \times l$ *input-queued switch).*

Proof: A $K_{2,l}$ bipartite network graph is depicted on the left in Figure 7-9. Its interference graph can then easily be shown to be two cliques of size l (K_l), connected by l disjoint edges, as depicted on the right in Figure 7-9. The result is then directly derived from Theorem 7.4.2. ■

It follows that a $K_{4,l}$ bipartite graph can be partitioned into two subgraphs, each of whose interference graphs satisfies OLoP. In Section 7.5.2, we will use this observation to evaluate the performance of our channel allocation algorithms.

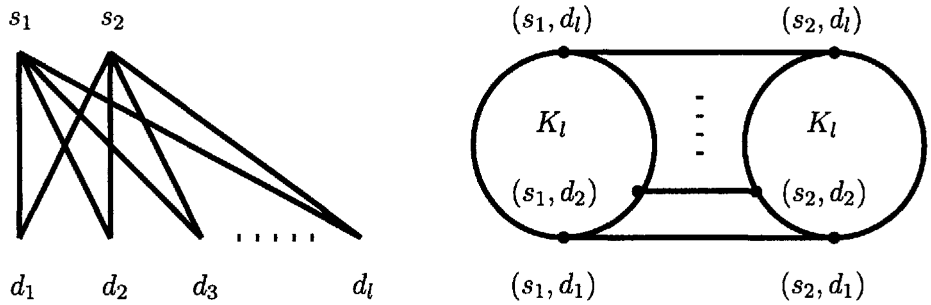


Figure 7-9: A network graph for a $K_{2,l}$ bipartite graph ($2 \times l$ input-queued switch) and the corresponding interference graph.

7.5 Channel allocation

The Channel Allocation Problem, introduced in Definition 7.2.1, seeks to assign a channel to every link such that each partition (operating in a different channel) can achieve 100% throughput by a distributed maximal weight scheduling algorithm. In this section our objective is to develop channel allocation algorithms that: (i) provide a large stability region and (ii) allow simple distributed algorithms to achieve 100% throughput in this region. As in Section 7.4.3, in order to demonstrate the presented concept, we assume that primary interference constraints hold.

In terms of LoP conditions, we seek to partition the network edges into channels such that the interference graph in each channel satisfies OLoP. The OLoP requirement is extremely challenging to incorporate into an optimization algorithm that generates a channel allocation, because it seeks the SLoP property for every subgraph on each channel. However, Corollary 7.4.1 shows that network graphs that are trees satisfy OLoP. Thus, it is sufficient to partition the edges of the network graph into channels such that each channel's network graph is a forest. This is the basis for our channel allocation algorithms.

Our channel allocation problem is equivalent to a coloring problem on the network graph. Namely, we seek to color the network edges such that edges of a single color do not compose a cycle (i.e. each color composes a forest). The minimum number of colors is known as the graph arboricity and can be found by an $O(m^2)$ algorithm [59].

Initially, we assume that all nodes have the same number of radios and that this number is equal to the number of channels (i.e. $R(v) = k \ \forall v \in V$).[5] When the number of available colors (channels) k is fixed, the k-forest problem [59,85] seeks to find the maximum number of edges of the graph that can be colored using only k colors without closing a single color cycle. This problem can be formulated as a *matroid*[6] *partitioning* or a *matroid intersection* problem. In order to enable the development of capacity expansion algorithms, we focus on the matroid intersection formulation. Under this formulation, the k-forest problem makes use of two matroids: the *graphic matroid* and the *partition matroid*. In our setting, we define these matroids by considering the graph $G_N^k = (V^k, \mathcal{E})$, equal to k disjoint *copies* of

[5] We will show below that this assumption can be relaxed.

[6] A matroid is a combinatorial structure $\mathcal{M} = (\mathcal{E}, \mathcal{I})$ in which $\mathcal{E}$ is a finite set of elements, and $\mathcal{I}$ is a collection of subsets of $\mathcal{E}$ satisfying (i) $\emptyset \in \mathcal{I}$, and if $I \in \mathcal{I}$, then all proper subsets of I belong to $\mathcal{I}$, and (ii) if $I_1, I_2 \in \mathcal{I}$ with $|I_2| = |I_1| + 1$, then there exists $e \in I_2$ such that $I_1 \cup \{e\} \in \mathcal{I}$.

the network graph G_N. The graphic matroid $\mathcal{M}_1 = (\mathcal{E},\mathcal{I}_1)$ assigns to $\mathcal{I}_1$ all possible forests in G_N^k. The partition matroid $\mathcal{M}_2 = (\mathcal{E},\mathcal{I}_2)$ partitions $\mathcal{E}$ into $m \triangleq |E_N|$ sets, where the i-th set, $\mathcal{E}_i$, contains all k copies of edge i. The collection $\mathcal{I}_2$ contains all sets of edges that have no more than a single element in any set of the partitions: $I \in \mathcal{I}_2$ implies $|I \cap \mathcal{E}_i| \leq 1$ for $i = 1,\ldots,m$. By associating with each copy of G_N in G_N^k a *unique color*, it can be seen that the sets belonging to $\mathcal{I}_1 \cap \mathcal{I}_2$ can be equated to colorings, where each subgraph of a particular color is a forest. This directly corresponds to a valid channel allocation, where each channel's network graph is a forest. The k-forest problem is to find for a given k the largest set of edges belonging to the matroid intersection of the graphic and partition matroids.

7.5.1 Partitioning algorithms

Our first algorithm for the k-forest problem is the suboptimal Breadth-First Search (BFS) algorithm. Such an algorithm was used in [120] as a heuristic solution to this problem. Its major advantage is its low complexity of $O(k(m+n))$. Yet, in Section 7.6 we will show that there is a large gap between the BFS solution and the optimal solution.

Therefore, we selected an optimal algorithm as a basis for developing our capacity expansion algorithms. The optimal solution to the k-forest problem can be found in polynomial time [59,85] by several algorithms. One of these algorithms is the *Matroid Cardinality Intersection* (MCI) algorithm of Lawler [85]. We present the MCI algorithm below, specialized to the k-forest problem of interest here. Given a valid coloring $I \in \mathcal{I}_1 \cap \mathcal{I}_2$, the MCI algorithm searches for an *augmenting path*, consisting of an alternating sequence of edges not in I and edges in I, such that when the edges of the path belonging to I are removed from I and those not belonging to I are added, the resulting coloring (channel allocation) belongs to $\mathcal{I}_1 \cap \mathcal{I}_2$ and its cardinality has increased by 1 (for more details see [85]). The complexity of the MCI algorithm is $O(km^2n' + k^2mn(n')^2)$, where $n' = \min\{n, m/k\}$. In the description of the following algorithms, we refer to two copies of the same edge on different colors in G_N^k as *parallel edges*.

Our channel allocation framework admits the practical situation where each node v is equipped with $R(v)$ radios (interfaces). Namely, different nodes have a different number of radios. In the formulation of the matroid intersection problem, we define the graph G_N^k as the disjoint union of k *identical* copies of the network G_N. This corresponds to the case, where each node is equipped with exactly k radios. Essentially, rather than generating k copies of each network graph edge, each network link should only have an edge represented in the i-th copy of the network graph G_N when there is a radio for that link available for use of the i-th channel.[7] Without loss of generality we refer to any graph defined in this manner as $G_N^k = (V^k, \mathcal{E})$. The matroid intersection properties, the MCI algorithm, and the algorithms described in Section 7.5.2 can then be applied to G_N^k.

Once the channel allocation is performed, at each time slot, one can use the distributed approximation algorithm of [71] that finds the maximal weight (greedy) solution, thereby providing 100% throughput. The (local) computational complexity of this algorithm is $O(1)$,

[7] When different nodes have a different number of radios, the specific allocation of the links to the different copies may affect the capacity region. An efficient allocation algorithm is a subject for further research.

Algorithm 12 Matroid cardinality intersection (MCI) [85]

```
Let the initial edge set be $I_{\mathrm{mci}} = \emptyset$
repeat
  Remove all labels associated with every edge
  Label '+' on every edge $e$ such that $I_{\mathrm{mci}} \cup \{e\} \in \mathcal{I}_1$
  while $e =$ [edge with oldest unscanned label] $\neq \emptyset$ do
    if $e$ is labeled '+' and $I_{\mathrm{mci}} \cup \{e\} \in \mathcal{I}_2$ then
      (augmenting path has been found)
      break the while loop
    else if $e$ is labeled '+' then
      ($I_{\mathrm{mci}}$ has an edge parallel to $e$)
      Label '-' on the edge in $I_{\mathrm{mci}}$ that is parallel to $e$ (if the edge is unlabeled)
    else
      ($e$ is labeled '-')
      Label '+' on each unlabeled edge in the unique cycle in $(V^k, I_{\mathrm{mci}} \cup \{e\})$
    end if
  end while
  if $e \neq \emptyset$ ($\exists$ augmenting path) then
    Trace the alternating path of '+' and '-' labels that lead to the '+' label at $e$ by
    assigning the edges labeled '+' to $I_1$ and those labeled '-' to $I_2$
    $I_{\mathrm{mci}} \leftarrow (I_{\mathrm{mci}} \setminus I_2) \cup I_1$
  end if
until $e = \emptyset$
```

which is low relative to the $O(n^3)$ complexity of a centralized optimal algorithm required to solve (2.10) [85]. In addition, the centralized algorithm has to collect queue backlog information from all nodes at each time slot (for an extended comparison see [104]).

In the realistic situation where the number of channels k is fixed and *insufficient* to partition all the network edges into k forests, we apply the MCI algorithm (or BFS) to generate an initial allocation that is a k-forest, and assign the unallocated network edges to the k-th channel. Thus, the first $k-1$ channels are guaranteed to satisfy OLoP, while the k-th channel operates at a worst-case 50% throughput.

A (theoretical) optimal solution will partition the graph into the minimum number of OLoP satisfying components, whereas our algorithms partition into forests. In order to evaluate the performance of our algorithms, we consider complete bipartite graphs. It can be shown that two channels are necessary and sufficient to guarantee the satisfaction of OLoP in $K_{3,3}$. Applying MCI, we find that the arboricity of $K_{3,3}$ is 2 and conclude that MCI achieves the minimum number of channels to guarantee OLoP. This and similar results point to the strong performance of the MCI algorithm in partitioning the network into a small number of channels satisfying OLoP. Yet, the following lemma provides a lower bound on the performance in general. Define $\kappa^*(G_N)$ as the minimum number of channels necessary to partition the edges of a network graph G_N such that the interference graph of each partitioned subgraph satisfies OLoP.

Lemma 7.5.1 *For $\varepsilon > 0$ there is no approximation algorithm that partitions a network* *graph G_N into $\kappa(G_N)$ forests, where*

$$\kappa(G_N) \leq (1.5 - \varepsilon)\kappa^*(G_N), \forall G_N.$$

Proof: Consider a $K_{4,4}$ bipartite network graph. It can be partitioned into two $K_{2,4}$ network graphs. According to Corollary 7.4.2, under primary interference constraints, an interference graph of $K_{2,4}$ satisfies OLoP. Therefore, 2 channels are sufficient to guarantee the satisfaction of OLoP in $K_{4,4}$. Namely, $\kappa^*(K_{4,4}) = 2$. Since $K_{4,4}$ has 8 nodes, any forest in such a graph can have at most 7 edges. Since $K_{4,4}$ has 16 edges, its arboricity must be at least 3 (i.e. $\kappa(K_{4,4}) = 3$). Hence, there exists a graph G_N for which $\kappa(G_N) = 1.5\kappa^*(G_N)$. ■

7.5.2 Capacity expansion algorithms

An important undesirable feature of the MCI and BFS algorithms is that each successive channel has a *maximal* number of network edges assigned to it, given the assignment to the previous channels. We wish to balance the trees in order to expand the capacity, thereby expanding the achievable throughput.

We present three algorithms for improving the network capacity properties. Since the admissible region restricts the summed throughput of all edges incident on the same vertex in the network graph to 1, it is desirable to minimize the maximum vertex degree over the network graphs on each channel. The first algorithm is called R-Greedy, and it operates by greedily selecting edges incident on vertices of maximum degree and seeking any channel that they can be reallocated to, such that the new allocation belongs to $\mathcal{I}_1 \cap \mathcal{I}_2$ and the allocation has an improved maximum degree. We note that $e = (v_i, v_j)$ implies that $v_i \in e$ and $v_j \in e$. The algorithm makes use of the function $\mathrm{TF}_1(I)$, which returns a negative value when the maximum degree or number of vertices at maximum degree under allocation I improves upon that of a reference allocation, I_0.

$$\mathrm{TF}_1(I) = \Delta_I^* - \Delta_{I_0}^* + 1_{\{\Delta_I^* = \Delta_{I_0}^*\}} \left(\sum_v 1_{\{\Delta_I(v) = \Delta_I^*\}} - \sum_v 1_{\{\Delta_{I_0}(v) = \Delta_{I_0}^*\}} \right).$$

Above, $\Delta_I(v)$ denotes the degree of vertex v in graph (V^k, I), Δ_I^* indicates the maximum vertex degree in graph (V^k, I), and $1_{\{\cdot\}}$ is the indicator function. The complexity of the R-Greedy algorithm is $O(dnmkn')$, where d is the maximum vertex degree in G_N.

Algorithm 13 Greedy Reallocation (R-Greedy)

```
begin with any edge set $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ (this could be the output of BFS or MCI)
repeat
  $I_0 \leftarrow I$
  if $\exists e_1 \in I$, $e_2 \notin I$ such that $\exists v \in e_1$, $\Delta_I(v) = \Delta_I^*$, $\mathrm{TF}_1((I \setminus \{e_1\}) \cup \{e_2\}) < 0$ then
    $I \leftarrow (I \setminus \{e_1\}) \cup \{e_2\}$
  end if
until $I$ equals $I_0$
```

Our second and third capacity expansion algorithms search for capacity improvements by directly attempting to balance the vertex degrees over all channels. They make use of augmenting paths in the spirit of the MCI algorithm to find new locations for edges that are incident on heavily-loaded vertices. The *maximum degree reallocation* algorithm (R-MAXD) seeks to minimize the maximum degree over vertices in all channels. It proceeds by disabling edges incident on maximum degree vertices and searching for augmenting paths that do not use such edges. The algorithm uses the function TF_1 for evaluating channel allocations, and the function $\mathrm{ESF}_1^0(I)$ for selecting candidate edges to disable. $\mathrm{ESF}_1^0(I)$ returns all edges incident on vertices having maximum degree in graph (V^k, I),

$$\mathrm{ESF}_1^0(I) = \{e \in I : v \in e,\ \Delta_I(v) = \Delta_I^*\}.$$

The *average degree reallocation* algorithm (R-AVGD) seeks to reduce *any* vertex degree in the graph so long as the reduction does not lead to higher vertex degrees or more vertices of maximum degree elsewhere in the graph. R-AVGD employs the performance evaluation function TF_2,

$$\mathrm{TF}_2(I) = \sum_{i=1}^{\Delta_I^*} 2^i \mathrm{sign}\left(\sum_v 1_{\{\Delta_I(v)=i\}} - 1_{\{\Delta_{I_0}(v)=i\}}\right).$$

Above, the function $\mathrm{sign}(x) = -1$ if $x < 0$, $\mathrm{sign}(x) = 1$ if $x > 0$, and $\mathrm{sign}(0) = 0$. The function $\mathrm{TF}_2(I)$ returns a negative value when the first entry at which the degree sequence[8] of (V^k, I) differs from that of (V^k, I_0) is lower in the sequence of (V^k, I) than that in (V^k, I_0). This function encourages trading higher degree vertices for more vertices of lower degree. R-AVGD also makes use of the function $\mathrm{ESF}_2^v(I)$, which returns all edges incident on vertex v in I,

$$\mathrm{ESF}_2^v(I) = \{e \in I : v \in e\}.$$

We simultaneously present both algorithms as Algorithm 14, making use of the parameter PARAM_i, with $\mathrm{PARAM}_1 = \{0\}$, and $\mathrm{PARAM}_2 = V^k$.

Algorithm 14 Maximum Degree/Average Degree Reallocation algorithms (R-MAXD [$i = 1$]/R-AVGD [$i = 2$])

```
begin with any edge set $I \in \mathcal{I}_1 \cap \mathcal{I}_2$
repeat
  $I_0 \leftarrow I$
  for $v \in \mathrm{PARAM}_i$ do
    $I \leftarrow \arg\min_{\tilde{I}}\{\mathrm{TF}_i(\tilde{I}) :$
    $\tilde{I} = \text{CE-MCI}(I, \{e\}, \mathrm{ESF}_i^v, \mathrm{TF}_i, 1),\ e \in \mathrm{ESF}_i^v(I)\}$
  end for
until $I$ equals $I_0$
```

R-MAXD and R-AVGD employ the recursive procedure CE-MCI that successively disables edges until an improved augmenting path is found, or all possible configurations are exhausted. CE-MCI takes as input the initial channel allocation I, the set of edges E_0 to

[8] The degree sequence of a graph G is a *nondecreasing* sequence of the vertex degrees of G.

Algorithm 15 CE-MCI(I_0,E_0,ESF,TF,Depth)

```
$\mathcal{I} = \{I_0 \setminus E_0\}$
while $\exists I \in \mathcal{I}$ with $|I| < m$ do
  $\mathcal{I} \leftarrow \mathcal{I} \setminus \{I\}$
  remove labels from all edges; assign $I_+ = I_- \leftarrow \emptyset$
  label '+' on every edge $e$ such that $I \cup \{e\} \in \mathcal{I}_1$ and $e \cap E_0 = \emptyset$
  while $e$ = [edge with oldest unscanned label] $\neq \emptyset$ do
    if $e$ is labeled '+' and $I \cup \{e\} \in \mathcal{I}_2$ then
      trace the alternating path of '+' and '-' labels that lead to the '+' label at $e$ by
      assigning edges labeled '+' to $I_+$ and those labeled '-' to $I_-$
      $\mathcal{I} \leftarrow \mathcal{I} \cup \{(I \setminus I_-) \cup I_+\}$
    else if $e$ is labeled '+' then
      label '-' on the edge in $I$ that is parallel to $e$ (if the edge is unlabeled)
    else
      label '+' on each unlabeled edge in the unique cycle in $(V^k, I \cup \{e\})$
    end if
  end while
end while
$\mathcal{I} \leftarrow \mathcal{I} \cup \{I_0\}$; $I_{\text{rmci}} \leftarrow \arg\min_{I \in \mathcal{I}} \text{TF}(I)$
if $\text{TF}(I_{\text{rmci}}) = \text{TF}(I_0)$ then
  (failed to generate an improved augmenting path)
  if Depth < D_MAX then
    $I_{\text{rmci}} \leftarrow \arg\min_I \{\text{TF}(I)$ :
    $I = $ CE-MCI($I_0$,$E_0 \cup \{e\}$,ESF,TF,Depth+1),
    $e \in \text{ESF}(I_0 \setminus E_0)\}$
  else
    $I_{\text{rmci}} \leftarrow I_0$
  end if
end if
return $I_{\text{rmci}}$
```

exclude when it attempts to search for augmenting paths, the functions ESF and TF, and an integer to track the depth of the recursion. The maximum depth of the recursion can be set using the constant D_MAX. While the MCI algorithm modifies the channel allocation at each iteration upon the discovery of its first augmenting path, CE-MCI labels over the entire graph and *selects the best augmenting path available* between all such paths found, in terms of the function TF.

The complexity of the algorithms is a function of the complexity of the MCI algorithm, which we denote by $c(\text{MCI})$. The complexity of R-MaxD is $O(dnm^{\text{D\_MAX}}c(\text{MCI}))$ and of R-AvgD is $O(d^{\text{D\_MAX}}nmc(\text{MCI}))$. As long as the search depth D_MAX is low, the complexity is reasonable. In the following section, we will see that significant capacity improvement is achieved for D_MAX = 2.

7.6 Performance evaluation

The partitioning and capacity expansion algorithms presented in Section 7.5 were implemented in Matlab and tested on numerous randomly generated networks. In this section we briefly describe the numerical results obtained for a number of representative cases. All presented results have been obtained for randomly generated instances in which the nodes are uniformly distributed in a plane of size $1000m \times 1000m$, with a link existing between two nodes if the distance between them is at most $250m$. We intentionally present results regarding relatively dense networks, since in very sparse networks the partitioning solution is often trivial and does not shed light on the tradeoffs involved in capacity expansion. As in the previous sections, we assumed that primary interference constraints hold. The presented results were obtained assuming that the number of radios equals the number of channels and is the same for all nodes (i.e. $R(v) = k\ \forall v$). As described in Section 7.5.1, this assumption can be easily relaxed.

7.6.1 Partitioning algorithms

Figure 7-10 compares the average number of channels (k) required by the BFS and the MCI algorithms. The results are presented as a function of the number of nodes in the network (n), where for each value of n, the average was obtained over 100 different random instances. Over all cases tested, the BFS algorithm required on average 32% more channels than the optimal MCI algorithm. Such a performance gap was observed throughout our numerical studies. Consequently, it seems that despite the higher computational complexity, using a matroid intersection algorithm is beneficial. This is one of the reasons the MCI algorithm was chosen as the basis for our capacity expansion algorithms.

Figure 7-10 also presents an *upper bound* on the edge chromatic number, which is the minimum number of colors (channels) such that an edge coloring exists having no two equally colored edges incident on the same vertex. According to Vizing's Theorem, the edge chromatic number is bounded above by $\Delta^* + 1$, where Δ^* is the maximum vertex degree in the network [69]. The large gap between the optimal solution and the edge chromatic number upper bound arises because under edge coloring, all edges can be active simultaneously, while MCI creates trees on which transmissions still have to be scheduled. Hence, by using edge coloring, the capacity region is enlarged to $\lambda_{ij} \leq 1 \forall (i,j) \in E_N$. In many network instances, such a large capacity expansion requires numerous channels.

7.6.2 Capacity expansion algorithms

We now demonstrate the operation of the different capacity expansion algorithms on a specific randomly generated network with 20 nodes. Figure 7-11 illustrates an example of the channel allocations performed by the different algorithms in a network in which the required number of channels is 4. The figure presents the network and then, for each algorithm, the 4 forests. Figure 7-11(a) presents the solution obtained by the MCI algorithm. It can be seen that the leftmost forest is relatively dense, while the rightmost tree is sparse (it includes only a single edge). The capacity is not efficiently allocated in this solution, since most of the nodes do not use the fourth channel, while the first channel has to be shared

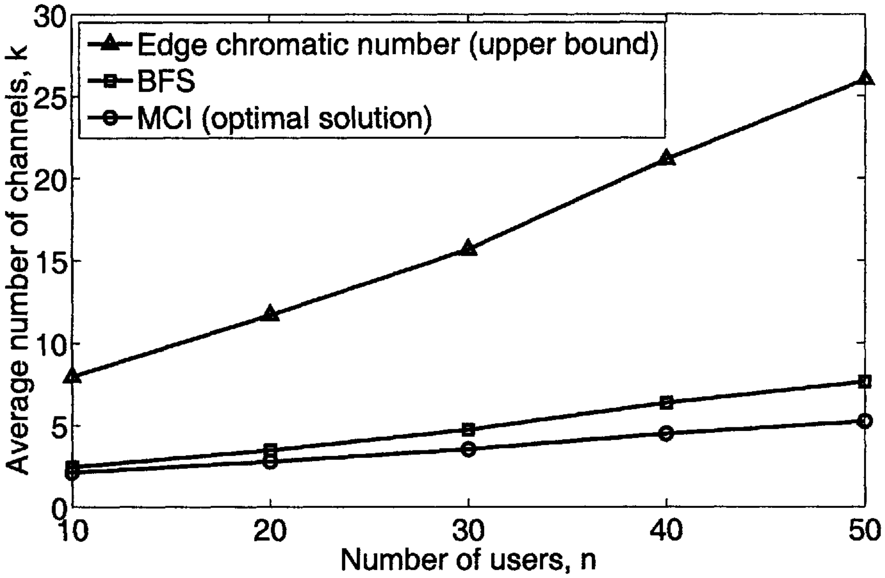


Figure 7-10: Average number of channels in the optimal solution, the number required by the BFS algorithm, and the upper bound.

by many links. Figure 7-11(b) presents the allocation performed by algorithm R-Greedy, using the MCI solution as input. It can be seen that several edges have now been migrated to the fourth (rightmost) channel. Figure 7-11(c) presents the allocation performed by algorithm R-MaxD, using the R-Greedy solution as input. The R-Greedy solution had two vertices of degree three, and R-MaxD manages to manipulate the allocation such that only a single vertex has degree three. Finally, the solution from R-MaxD is used as input in R-AvgD to obtain the channel allocation of Figure 7-11(d). Though the maximum vertex degree remains at three, lower degree vertices have had their degrees improved, with many more edges in this allocation entirely disconnected.

The example above demonstrates the operation of the capacity expansion algorithms. We now quantitatively evaluate their performance. Given a specific channel allocation it is not straightforward to represent the capacity region. This results from the fact that it is a polytope in $\mathbb{R}_{+}^{m}$. Yet, in order to obtain some insight, we make the following simplifying assumption regarding the capacity allocation that takes place once the channels are assigned to the links. We assume that some degree of fairness exists, and therefore, if possible, all edges connected to a node receive an equal share of the node capacity. This is sometimes impossible, due to a capacity limit resulting from the other node connected to an edge. Consequently, under this assumption the throughput on an edge (i, j) operating in channel k will be at least $(\max(\Delta_{i,k}, \Delta_{j,k}))^{-1}$, where $\Delta_{i,k}$ is the number of edges adjacent to node i that use channel k.

Accordingly, the first performance measure is *Average Capacity*, which is the average over all edges $(i, j) \in E_N$ of the above value. The second performance measure is the *Worst-Case Capacity*, which is the lowest capacity allocated to a link in the network. This

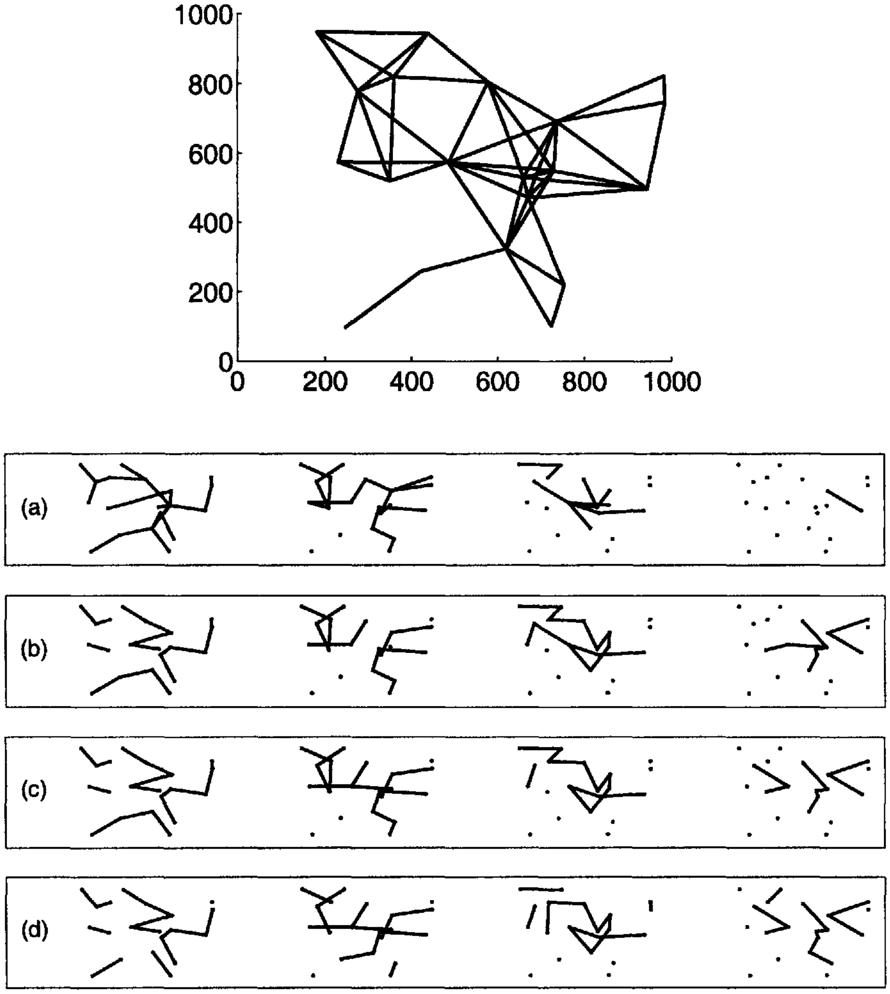


Figure 7-11: Channel assignments by (a) MCI (b) R-Greedy (c) R-MaxD, and (d) R-AvgD.

is inversely proportional to the maximum node degree over all nodes and all channels. Using the above notation, it is equal to $(\max_{i,k} \Delta_{i,k})^{-1}$.

Figure 7-12 illustrates these performance metrics for random networks with different numbers of nodes (n). For each value of n, the results were averaged over 50 different random network instances. It can be seen that both for the worst case and the average case, R-GREEDY provides significant throughput improvement over the MCI algorithm (average improvement of 29% and 40% in the average and worst-case capacity, respectively). This is notable, since the complexity of the greedy capacity expansion algorithm is small relative to that of MCI. When using the R-MAXD and R-AVGD, we employed a maximum search depth of D_MAX = 2. This implies that the complexities of R-MAXD and R-AVGD are respectively $O(dnm^2)$ and $O(d^2nm)$ times the complexity of MCI. Despite the higher complexities, the value of these algorithms is evident from their ability to significantly improve the performance metrics. Relative to the MCI solution, R-MAXD achieves average improvements of 36% and 56% in the average and worst-case capacities, respectively, while R-AVGD achieves 45% and 56%, respectively.[9] There is an evident tradeoff between complexity and performance. Since the channel allocation problem is solved in a different time scale from the scheduling problem, it seems beneficial to use R-MAXD or R-AVGD.

In realistic situations the number of channels and radios is bounded. Figure 7-13 depicts the average capacity metric versus the number of available channels (k) for a network with 20 nodes. For each value of k, the results were averaged over 50 different random network instances. Given a fixed k, the MCI, R-GREEDY, R-MAXD, and R-AVGD algorithms were enlisted to obtain and expand the capacity of k-forests. In instances where there were edges that could not be included in a valid k-forest, these edges were added to the last generated forest (at channel k). As explained in Section 7.5.1, the first $k-1$ channels are guaranteed to satisfy OLoP, while the k-th channel operates at a worst-case 50% throughput. If there was a cycle in the k-th channel, we assumed that the edges in the k-th channel achieve only 50% throughput when calculating the average capacity. Algorithms R-GREEDY, R-MAXD and R-AVGD provide significant improvement over the MCI algorithm alone.

[9] Note that the plots of the worst-case capacity for R-AVGD and R-MAXD overlap.

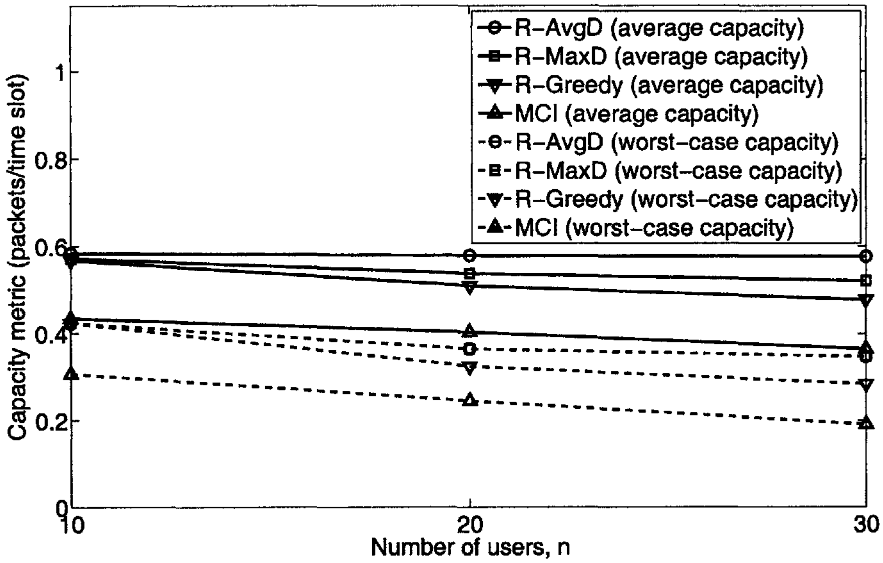


Figure 7-12: Average and worst-case capacities.

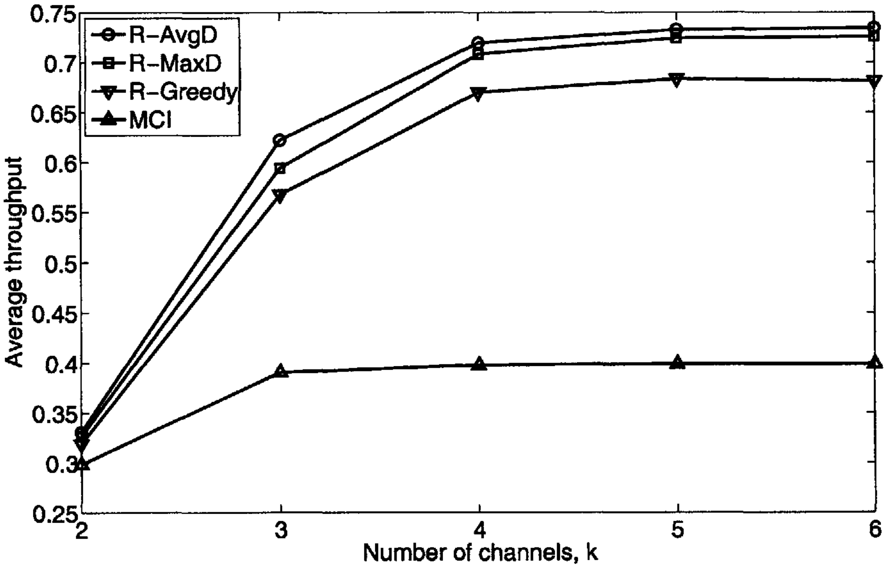


Figure 7-13: Average capacities given a fixed number of channels k.

7.6.3 Comparison with other static channel allocation algorithms

Thus far, our simulation studies have provided absolute measures of the performance of our proposed channel allocation algorithms. In this section, we compare the static channel allocation generated by our algorithm next that of [77]. We find that our rebalanced channel allocations are typically superior, in terms of maximum achievable throughput, than those generated by the algorithm of [77].

In [77], the authors consider the joint problem of determining multi-hop routes and channel allocation in wireless mesh networks. The authors solve the routing problem using a linear program, and subsequently use the implied link loads to determine an effective channel allocation (the algorithm can be found in [77, Figure 5]). Essentially, each (link,channel) combination is provided with a weight, and the algorithm successively determines the minimum weighted link and assigns a channel to that link. This algorithm has one ambiguity: it does not provide a tie-breaking condition for allocating a channel to a link, when multiple channels have the same weight. In our numerical studies, we find that the choice of tie-breaking condition has an effect on throughput performance, so we distinguish two versions of the algorithm: 1) ties are broken by selecting the channel with lowest index (*KN scheme*); 2) ties are broken by randomly selecting amongst equally weighted channels (*KN scheme with random tie-break*).

Note that the static channel allocation algorithm of [77] is intended to be used in conjunction with a time-division multiplexing (TDM) scheduler. This is because the traffic is assumed to be known and deterministic, in which case a fixed TDM schedule can be implemented for servicing the link loads. Recall that our scheduling objective is to service packets that arrive *stochastically*. Thus, for our simulations we cannot assume that the link loads are known in advance. Consequently, our simulations employ maximal weight scheduling, in order to dynamically adjust service rates based on traffic variations, with no assumptions made regarding the rate vector $\boldsymbol{\lambda}$. However, $\boldsymbol{\lambda}$ is explicitly considered an input to the channel allocation algorithm of [77]. In order to compare the performance of the channel allocation of [77] with our proposed allocation, we provide the algorithm of [77] with the true value of the long-term arrival rate vector $\boldsymbol{\lambda}$, and employ maximal weight scheduling for servicing packets that arrive to the network stochastically.

In order to measure throughput performance, we will consider uniform arrivals: $\lambda_{ij} = \lambda$ for all i, j. We will refer to the maximum value of λ in which the queues in the network remain stable (i.e. do not grow without bound) as the *maximum achievable throughput* of the network.

For our simulations, we consider $k = 3$ channels available at each edge, with 3 radios at each node, and require that at most one channel can be allocated to any edge. We present results relating to four channel allocation methods: 1) Our proposed static allocation, where we apply the MCI algorithm, the greedy rebalancing heuristic, the maximum degree rebalancing algorithm, and the average degree rebalancing algorithm in sequence, followed by assigning any unallocated edges to the lowest channel index; 2) the KN static channel allocation; 3) the KN static channel allocation with random tie-break; and 4) dynamic channel allocation, where links are not bound to channels, and (link,channel) combinations are activated at each slot based on maximal weight scheduling. Note that the dynamic

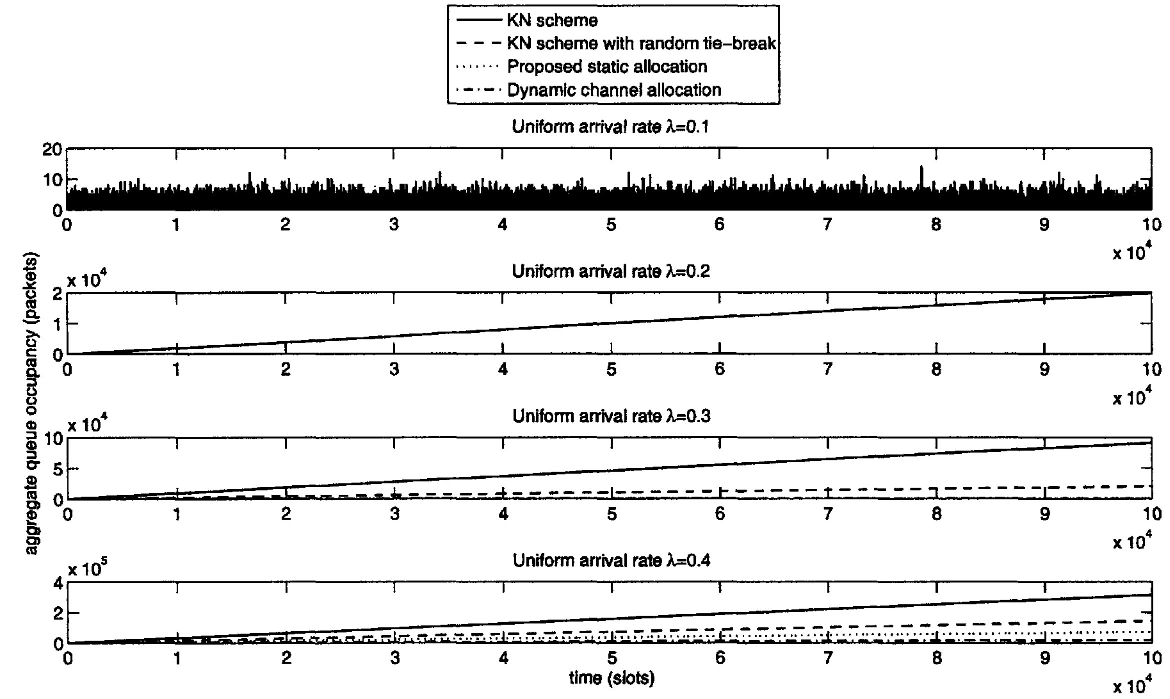


Figure 7-14: Simulated trajectories of four schedulers under Poisson-distributed arrivals, with uniform arrival rates $\lambda = 0.1, 0.2, 0.3, 0.4$ packets/time slot.

channel allocation method (point 4 above) has the advantage of being allowed to modify its channel allocation at each time slot. Consequently, one might expect that its performance is superior to any static allocation scheme.[10] Nevertheless, the throughput gap between static and dynamic channel allocations is of interest, since it clarifies the trade-off of performance against scheduler complexity.

Figure 7-14 shows several simulated trajectories for the various channel allocation methods described above. In each case, the system is subject to Poisson arrivals, and maximal weight scheduling is always employed. As in previous simulations, this study is based on a single placement of 25 users in a $1km \times 1km$ field. Observe that the lowest throughput performance is incurred by the KN scheme, where the queue backlog grows without bound at the uniform arrival rate $\lambda = 0.2$ packets per slot. The next lowest throughput performance is incurred by the KN scheme with random tie-break (unstable at $\lambda = 0.3$ packets per slot), followed by our proposed method and the dynamic channel allocation algorithm (both unstable at $\lambda = 0.4$ packets per slot). Observe that at $\lambda = 0.4$ packets per slot, none of the four channel allocation schemes enables stability of maximal weight scheduling. Figure 7-15 plots the average aggregate queue occupancy versus the uniform arrival rate λ

[10] This is not always the case under *maximal* weight scheduling: recall the example of Section 7.3.2 where static channel allocation was shown to be superior to dynamic allocation in the 6-ring.

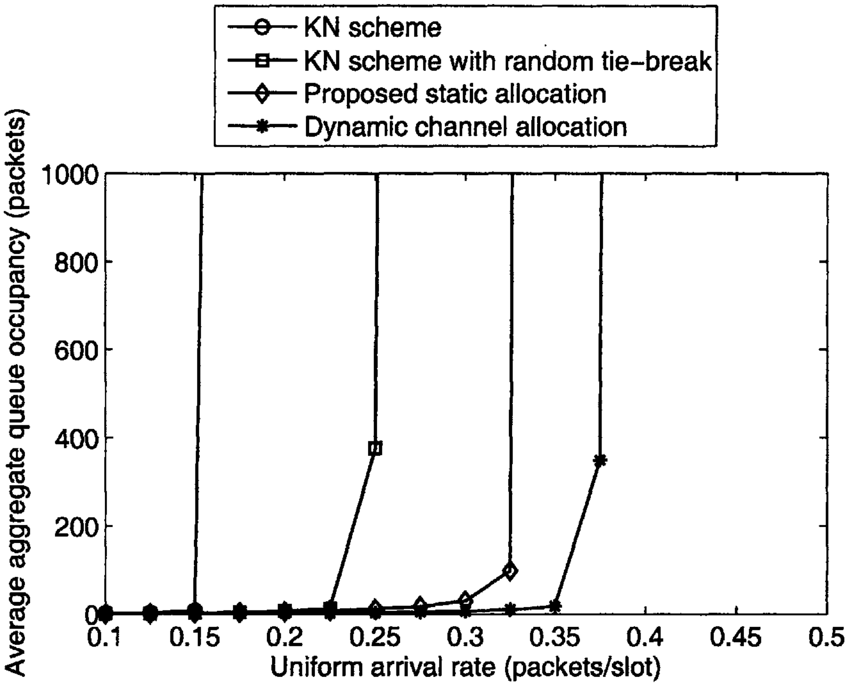


Figure 7-15: Average aggregate queue occupancy versus average uniform arrival rate. Each point is generated from a sample path of duration 250, 000 time slots.

experienced in the same network under the various channel allocations. This plot clarifies the relative throughput performance of the four channel allocation algorithms: The maximum throughput achievable under KN, KN with random tie-break, the proposed scheme, and the dynamic channel allocation are respectively: 0.15, 0.25, 0.325, and 0.375 packets per slot.

We considered 25 randomly generated mesh networks, each subject to uniform arrival rates. Figure 7-16 presents the maximum throughput performance of the different channel allocation algorithms in each of these networks. It can be seen that our channel allocation algorithm usually outperforms the other static channel allocations, with only two instances (network indices 15 and 17) in which one of the KN schemes achieves higher throughput. Overall, our channel allocation outperforms the best KN scheme by an average of 25%. Additionally, randomly breaking ties in the KN scheme usually leads to improved throughput performance over the simple KN scheme, with an average throughput performance improvement of 15%. Finally as expected, dynamic channel allocation always outperforms static allocation, with an average throughput performance improvement of 33% over the best static allocation.

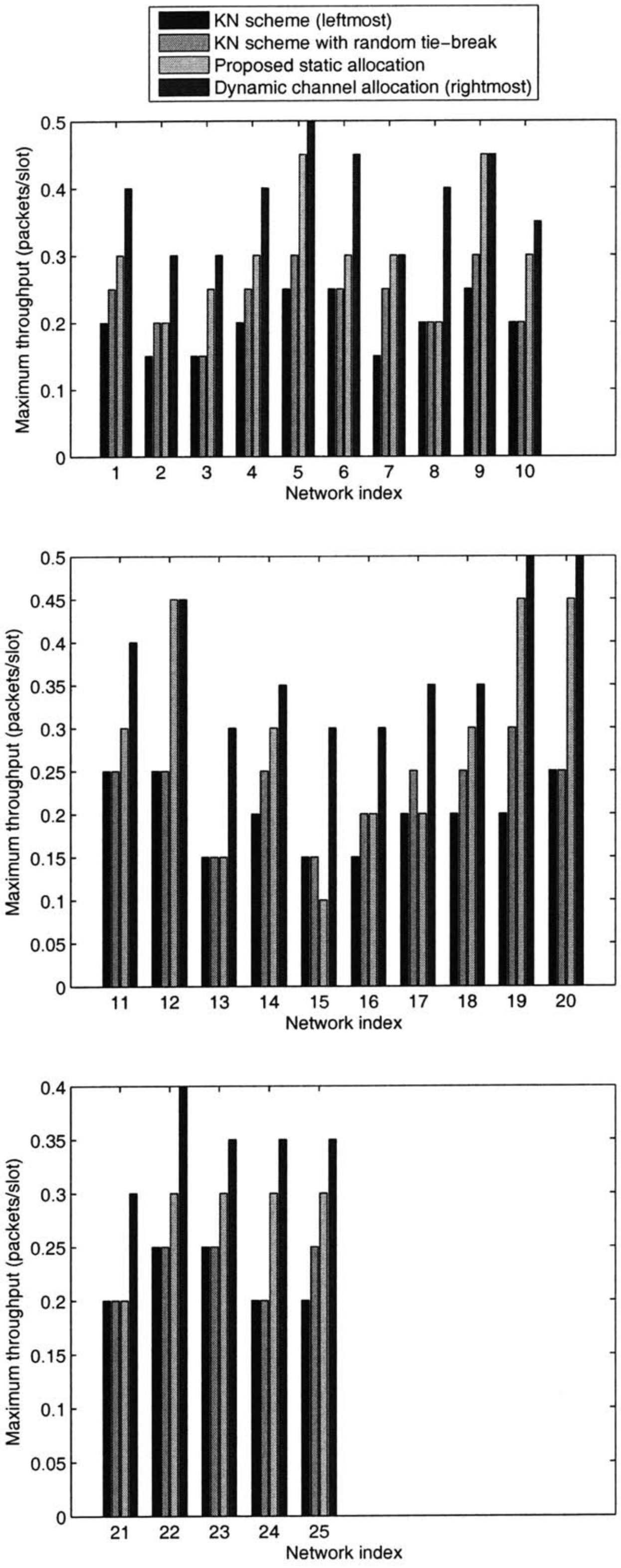


Figure 7-16: Maximum throughput for various channel allocation schemes.

7.7 Conclusions

In this chapter we have applied techniques stemming from stability theory and matroid theory to obtain novel results regarding the design of Wireless Mesh Networks. The application of these theories allows us to develop algorithms for partitioning a mesh network into a number of high capacity subnetworks such that in each of the subnetworks simple distributed algorithms can obtain 100% throughput.

We have performed a study of the implications of Local Pooling on network design and shown that although the notion of Local Pooling is rather abstract, its implications are quite powerful. Based on some of our observations, we developed matroid intersection algorithms for efficient network partitioning. In Section 7.6 we have shown that these algorithms perform very well in terms of capacity. We note that the scope of this work spans more than multi-radio multi-channel WMNs. It seems to be relevant to any wireless network with stochastic arrivals in which transmissions can be differentiated in the time domain (i.e. scheduling) as well as in other domains (frequency, code, etc.).

This chapter primarily provides a *theoretical contribution* that lays the foundation for developing *practical algorithms*. Hence, there are still many problems to deal with. For example, a future research direction is to allow dynamic channel allocation. This will require to tailor the channel allocation algorithms for online and perhaps distributed operation. In addition, Lemma 7.5.1 indicates that partitioning into trees may be suboptimal. Therefore, we would like to develop matroid intersection algorithms that will partition into other components similar to the ones identified in Section 7.4. In general, we would like to develop algorithms that partition the network to the minimum number of OLoP-satisfying components. It seems that this may be done by utilizing connections between the maximal independent sets in the interference graph and the characteristics of the graphic and partition matroids.

In Section 7.2.1, we mentioned SINR-based interference models as a useful (and perhaps more realistic) alternative to the graph-based models we consider in this chapter, as well as through the remainder of the thesis. Although an SINR-based interference model does not admit the use of an interference or conflict graph in obtaining maximal weight link activations, one could propose distributed scheduling techniques that arrive at valid SINR-constrained link activations. Given such a scheduling algorithm, it is not difficult to adapt the Local Pooling results of [51] into this setting. In particular, when we consider maximal weight scheduling in a network having a well-defined interference graph, the essential reasoning is that among the maximum weighted vertices of the *interference* graph (equivalently edges of the *network* graph), a *maximal* independent set must always be selected. Consequently when the set of vertices $V \subseteq V_I$ have dominant weights in the interference graph, the scheduler must exclusively select independent sets in the interference graph *that are maximal over the vertex set* V. For this reason, the maximal matrix $\mathbf{M}(V)$ appears in the Local Pooling definitions. In order to extend the Local Pooling analysis into other scheduling settings, such as an SINR-constrained network, we must simply understand what are the different *regimes* of network operation, and what are the possible link activations available to the scheduler in each regime. In the model containing an interference graph, the *regimes* consist of all possible sets of network edges (or vertices of the interference graph) that can simultaneously have maximum weight, and the possible link activations are contained in the matrix $\mathrm{M}(V)$. Thus, for scheduling in an SINR-constrained network, or given a different scheduling algorithm, a Local Pooling analysis can still be conducted. It is only necessary to identify the different service regimes, and the possible link activations corresponding to each regime.

Appendix

7.A Proof of Lemma 7.3.2

Based on Edmonds' Theorem [53] and the analysis of [68] and [169], in such networks the arrival rates should satisfy the following constraints:

$$\sum_{(i,j)\in E_N} \lambda_{ij} \leq 1 \quad \forall i \in V \tag{7.3}$$

$$\sum_{(i,j)\in L(U)} \lambda_{ij} \leq \lfloor |U|/2 \rfloor \quad \forall U \subseteq V,\ |U|\,\text{odd} \tag{7.4}$$

$$\lambda_{ij} \geq 0 \quad \forall (i,j) \in E_N. \tag{7.5}$$

where $L(U) \subseteq E_N$ is the collection of links connecting nodes in U. Using a maximal weight matching algorithm along with a two speedup (using two frequencies on each link) achieves the region defined in (7.3)-(7.5). Alternatively, assume that the network can be partitioned into two subnetworks with non-overlapping edges (denoted by $G_N^1 = (V^1, E_N^1)$ and $G_N^2 = (V^2, E_N^2)$) such that their interference graphs satisfy OLoP. In that case, the distributedly achievable stability region is defined by the following constraints that should hold for $k = 1, 2$:

$$\sum_{(i,j)\in E_N^k} \lambda_{ij} \leq 1 \quad \forall i \in V^k$$

$$\sum_{(i,j)\in L(U)} \lambda_{ij} \leq \lfloor |U|/2 \rfloor \quad \forall U \subseteq V^k,\ |U|\,\text{odd}$$

$$\lambda_{ij} \geq 0 \quad \forall (i,j) \in E_N^k.$$

This region is larger than the region in (7.3)-(7.5).

7.B Proof of Theorem 7.4.2

Designate the two cliques $G_I^1 = (V_I^1, E_I^1)$ and $G_I^2 = (V_I^2, E_I^2)$, where $V_I^1 \cap V_I^2 = \emptyset$ and $E_I^1 \cap E_I^2 = \emptyset$. Further, let E_d be the set of disjoint edges connecting G_I^1 and G_I^2. We then have $G_I = (V_I, E_I)$, where $V_I = V_I^1 \cup V_I^2$ and $E_I = E_I^1 \cup E_I^2 \cup E_d$. Consider the induced subgraph over the vertex set $V \subseteq V_I$. If $V \cap V_I^1 = \emptyset$ or $V \cap V_I^2 = \emptyset$, then Lemma 7.4.2 implies that V satisfies SLoP. If $|V \cap V_I^1| = 1$ and there exists $v \in V_I^2$ such that $(V \cap V_I^1, v) \in E_d$, then Lemma 7.4.1 ensures that SLoP is satisfied for V. If $|V \cap V_I^1| = 1$ and there is no $v \in V_I^2$ such that $\{V \cap V_I^1, v\} \in E_d$, then the induced subgraph over V consists of the disjoint union of two cliques, which satisfies SLoP by Lemma 7.4.2 and Proposition 7.4.1. The same reasoning applies when $|V \cap V_I^2| = 1$. Finally, when $|V \cap V_I^1| > 1$ and $|V \cap V_I^2| > 1$, we claim that every maximal independent set of the induced subgraph of vertices V in G_I contains two vertices. Denote by $\bar{G_I^1}$ the induced subgraph over G_I^1 and $\bar{G_I^2}$ that over G_I^2. Since both $\bar{G_I^1}$ and $\bar{G_I^2}$ are cliques, no more than two vertices can belong to any independent set, one in each clique. Suppose a maximal independent set contains one vertex, v, without loss of generality this vertex belongs to $\bar{G}_I^1$. By definition of the set E_d, v can only share an edge with a single vertex of $\bar{G}_I^2$. Then, if no vertex of $\bar{G}_I^2$ can be added to the independent set, $\bar{G}_I^2$ must be K_1, since otherwise any vertex of $\bar{G}_I^2$ not incident on v could be added. This is a contradiction. Consequently SLoP must be satisfied on such a subgraph. Thus, we have that SLoP is satisfied on any subgraph of G_I, which implies that OLoP is satisfied.

7.C Proof of Theorem 7.4.3

Note that any connected subgraph of a tree of blocks is tree of blocks or a forest of blocks. Thus, we only need to consider satisfaction of the SLoP properties of any tree of blocks, which will provide the satisfaction of OLoP for any tree of blocks. If the tree of blocks $G = (V, E)$ has any clique $K_l, l \geq 2$ associated with a leaf of the tree, then one vertex of this clique must belong to every maximal independent set of the tree of blocks. Consequently setting $\alpha_i = 1$ for any vertex corresponding to this clique and $\alpha_i = 0$ otherwise provides $\boldsymbol{\alpha}^T M(V) = \mathbf{e}^T$ and we conclude that SLoP is satisfied.

It remains to consider the case where every leaf of the tree of blocks corresponds to two cliques connected by any number of disjoint edges. Consider any such block and in particular we focus on the clique that has no other blocks sharing a vertex with it. Then it is clear that the proof of Theorem 7.4.2 applies to this clique, in that there must exist a vertex of this clique in every maximal independent set of vertices in G. Thus, SLoP must be satisfied for this configuration.

Since SLoP is satisfied for any tree of blocks, and each subgraph of a tree of blocks is a forest of blocks, we conclude that OLoP is satisfied for any tree of blocks.

Chapter 8

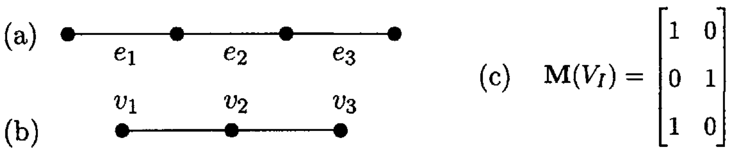
Distributed throughput maximization in wireless networks: Topology and interference considerations

In Chapter 7, we found several simple graph classes in which distributed *maximal* weight schedulers achieve 100% throughput. We proceeded to use these results to develop channel allocation algorithms that provide attractive throughput properties under primary interference. In this chapter, we deepen our understanding of graphs that satisfy Local Pooling (LoP). Furthermore, we consider more general interference conditions than simple primary interference.

8.1 Overview and summary of contributions

Identifying specific network topologies that satisfy LoP enables the design of algorithms that either partition a wireless network into subnetworks with such topologies (e.g. via channel allocation) or add artificial interference constraints that create such topologies. Hence, in Chapter 7, a few interference graphs satisfying LoP were identified and it was proved that under primary interference constraints, tree network graphs yield interference graphs that satisfy LoP. Although some knowledge about LoP has been acquired, [51] provides abstract conditions, while Chapter 7 focuses on primary interference constraints. Despite the fact that these constraints may hold for specific technologies, they are not realistic in most practical settings. Therefore, in order to allow the development of algorithms that take advantage of LoP, in this chapter we focus on identifying topologies of interference and network graphs that satisfy the LoP conditions, and studying the effect of *multihop* interference on these topologies.

We first use the LoP conditions to identify several new classes of LoP-satisfying graphs. It is shown that within the class of perfect graphs, chordal graphs, chordal bipratite graphs, cographs, and a subgroup of co-comparability graphs all satisfy LoP. These observations



$$\mathrm{M}(V_I) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Figure 8-1: (a) Undirected network graph G_N, (b) the corresponding interference graph G_I under primary interfernce, and (c) the matrix of maximal link activations.

increase the number of graphs that are known to satisfy LoP by a few orders of magnitude. We emphasize that despite the fact that in these graph classes a distributed maximal weight independent set algorithm usually *does not* achieve the optimal (maximum weight) solution, it achieves 100% throughput. We also show that all odd rings with at least 9 nodes and all even rings with at least 6 nodes do not satisfy LoP. Using the latter observation, we show that all bipartite graphs that are not chordal bipartite do not satisfy LoP.

We use the acquired knowledge about graph classes that satisfy and fail LoP to study the effect of increased interference on LoP. We focus on a generalization of the primary (1-hop) and secondary (2-hop) interference models to a k-hop interference model [142], where k is termed the interference degree. We show that in many cases, as k increases, it is more likely that the LoP conditions hold, and thereby, it is more likely that simple distributed algorithms achieve 100% throughput. Moreover, for every network topology, there is an interference threshold k^*, above which the corresponding interference graphs satisfy LoP. At first glance, it seems that since it is known that the worst case performance deteriorates as the interference degree increases [36, 91, 161], the results are counter-intuitive. Yet, the actual meaning of the results is that in many topologies, as k increases, the resulting interference graph is such that distributed maximal weight scheduling achieves the maximum throughput instead of the worst case throughput.

To summarize, this chapter focuses on identifying properties of network topologies satisfying the Local Pooling conditions. The main contributions are two-fold. First, we identify several graph classes that satisfy Local Pooling. Second, we show that due to Local Pooling, as the interference degree increases, it is more likely that simple distributed algorithms achieve 100% throughput. The obtained results can serve as a basis for the development of Local Pooling based algorithms.

8.2 Network model

We maintain the same network model as employed in Chapter 7 (see Section 7.2). To reiterate the undirected network graph and its interference properties, Figure 8-1 depicts an undirected graph and presents the corresponding interference graph and matrix of maximal link activations.

8.3 Interference graphs satisfying local pooling

The OLoP properties of graphs are only beginning to be understood. In Chapter 7, small graphs were studied by exhaustive search. Additionally, structural properties were used

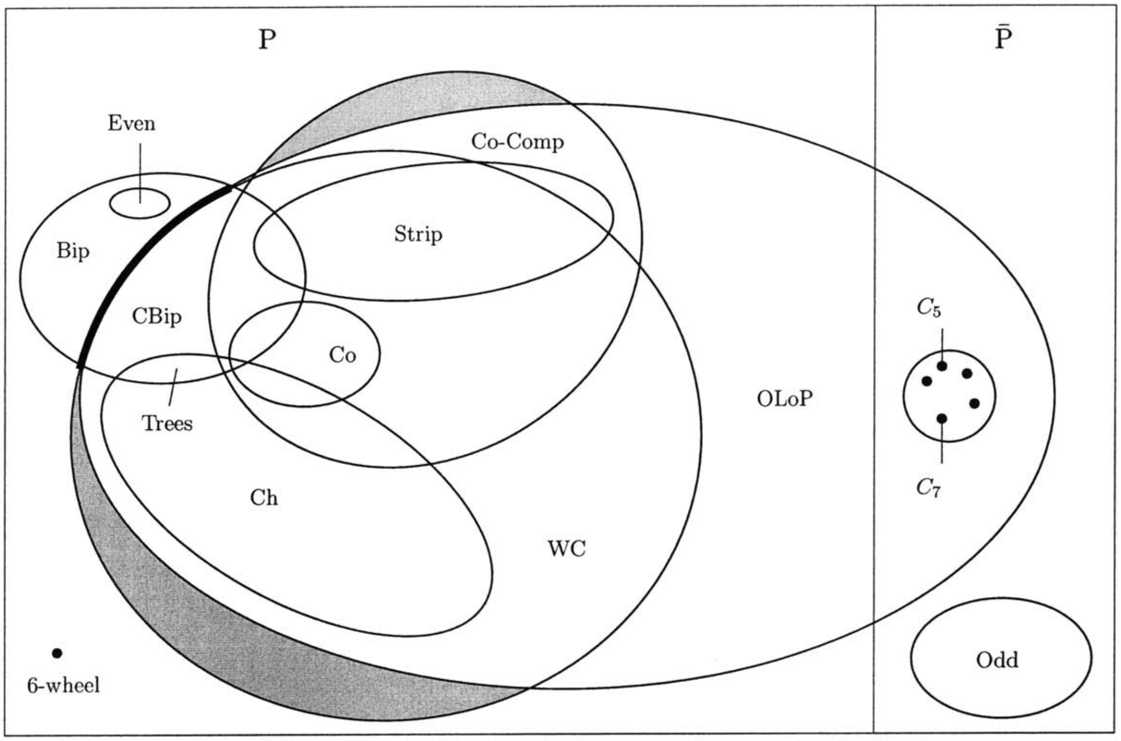


Figure 8-2: The relations between the OLoP-Satisfying class and other graph classes: P - perfect, P̄ - non-perfect, WC - weakly chordal, Ch - chordal, CBip - chordal bipartite, Bip - bipartite, Co - cograph, Co-Comp - co-comparability, Strip - strip-of-cliques, Even - cycles C_n with n even and $n \geq 6$, Odd - graphs with induced C_n with n odd and $n \geq 9$.

in [51] and Chapter 7 to show that the following interference graphs satisfy OLoP: trees, forests, *clique trees*, where each pair of cliques shares at most a single vertex, and a *pair-of-cliques* connected by disjoint edges.

Definition 8.3.1 (OLoP-Satisfying) *The collection of graphs for which OLoP is satisfied is called the* OLoP-Satisfying *class.*

In order to better understand the effect of interference on LoP, we use structural properties to identify various graph classes that satisfy OLoP. We identify known graph classes that are included within the OLoP-Satisfying class or intersect with it. It turns out that all the graph classes we identify using structural properties are subclasses of the class of perfect graphs. On the other hand, some of the graphs identified by the exhaustive search of Chapter 7 are not perfect graphs. Hence, in the following discussion we differentiate between perfect and non-perfect graphs. Our investigation leads to the taxonomy of graph classes depicted in Figure 8-2, showing the relationship of the OLoP-Satisfying class to the graph classes considered here.

We will make use of the following graph properties and definitions. For graph $G = (V, E)$, the *induced subgraph* over vertex set $V' \subseteq V$ is the graph $G' = (V', E')$, where E' is the set of edges in E whose endpoints are in V'. The complement $\overline{G} = (V, \overline{E})$ of graph

$G = (V, E)$ is defined by

$$\overline{E} = \{(u, v) : u, v \in V,\ u \neq v \text{ and } (u, v) \notin E\}.$$

A *chord* of a cycle (path) is an edge between two vertices of the cycle (path) that is not an edge of the cycle (path). A cycle (path) is *chordless*, if it contains no chords. We denote by C_n and P_n a chordless cycle and a chordless path, respectively, of length n. We denote by K_n a clique (complete graph) of n nodes. The set of neighbors of node v is denoted by $N(v)$.

8.3.1 Perfect graphs

A graph is *perfect*, if for each induced subgraph the size of the largest clique equals the chromatic number[1]. Several classical graph classes such as bipartite graphs, chordal graphs, comparability graphs, and their complements are perfect [25]. Here, we will identify a number of important classes of perfect graphs that are also subclasses of the OLoP-Satisfying class. We will show that all of the graphs identified in [29], [51] are *simple* special cases in these classes. The following graph classes are of particular interest.

Definition 8.3.2 (Chordal [25]) *A graph G is* chordal *if each cycle in G of at least* 4 *nodes has at least one chord.*

Definition 8.3.3 (Weakly Chordal [25]) *A graph G is* weakly chordal *if G and $\overline{G}$ contain no induced chordless cycle C_n, $n \geq 5$.*

Definition 8.3.4 (Chordal Bipartite [25]) *A bipartite graph B is* chordal bipartite *if each cycle in B of length at least* 6 *has a chord.*

Definition 8.3.5 (Cograph [25]) *A graph is a* cograph *if it does not contain the path graph P_4 (depicted in Figure 8-1(a)) as an induced subgraph.*

Notice that the chordal bipartite class is the intersection of the weakly chordal and bipartite classes. The following series of lemmas concern the OLoP properties of several large graph classes.

Lemma 8.3.1 *Every chordal graph satisfies OLoP.*

Proof: See Appendix 8.A. ∎

Lemma 8.3.2 *Every chordal bipartite graph satisfies OLoP.*

Proof: See Appendix 8.B. ∎

Lemma 8.3.3 *Every cograph satisfies OLoP.*

[1] Recall that the chromatic number is the smallest number of colors needed to color the vertices of a graph so that no two adjacent vertices share the same color.

Proof: See Appendix 8.C. ■

Lemma 8.3.4 *Every even cycle C_n with $n \geq 6$ fails SLoP.*

Proof: See Appendix 8.D. ■

Corollary 8.3.1 *Every bipartite graph that is not chordal bipartite does not belong to the OLoP-Satisfying class.*

Proof: By definition, if a bipartite graph is not weakly chordal (i.e. not chordal bipartite), it includes an even cycle C_n with at least 6 vertices. This cycle is an induced subgraph that, according to Lemma 8.3.4, fails SLoP. Hence, OLoP fails in bipartite graphs that are not weakly chordal. ■

Figure 8-2 illustrates the inclusion of the chordal, chordal bipartite, and cograph classes within the OLoP-Satisfying class. The class of chordal graphs has a few notable subclasses (i.e. classes of special graphs that are known to be chordal), including the strongly chordal, split, interval, threshold, and tree classes (additional subclasses are documented in [25]). Lemma 8.3.1 implies that all these subclasses satisfy OLoP. Therefore, the observation of [51] that trees satisfy OLoP immediately follows, as does the observation of Chapter 7 that every clique tree satisfies OLoP, since clique trees are chordal. Lemma 8.3.2 implies that all subclasses of chordal bipartite graphs satisfy OLoP, including the convex and bipartite $\cap$ distance-heriditary classes.

The contribution of Corollary 8.3.1 is its characterization of a *sharp* boundary separating the chordal bipartite graphs (OLoP-satisfying) from the bipartite graphs that are not chordal bipartite (not OLoP-satisfying). This boundary is depicted as a thick line in Figure 8-2. This result follows directly from the failure of the OLoP conditions in even cycles C_n with $n \geq 6$. Hence, any graph class that includes the bipartite graphs as a subclass cannot be fully included within the OLoP-Satisfying class. This allows us to exclude many of the major classes of perfect graphs (e.g. preperfect, strongly perfect, quasi-parity, and bip* [25]) as subclasses of the OLoP-Satisfying class.

We note that there exist other specific perfect graphs that are not bipartite and fail OLoP. For example, according to the exhaustive search of [29], a graph known as the 6-wheel [69] fails OLoP. In Figure 8-2, this graph appears outside the OLoP-Satisfying class.

Two major classes that have not been excluded as subclasses of the OLoP-Satisfying class are the weakly chordal graphs and the co-comparability graphs, defined next.

Definition 8.3.6 (Co-comparability [67]) *A graph is a co-comparability graph if it is the intersection graph of a set of curves*[2] *between two parallel lines in the plane, where every curve has one endpoint on each of the lines.*

In Figure 8-2 we have shaded portions of the weakly chordal and co-comparability classes to indicate the uncertainty of their inclusion relations with OLoP-Satisfying. Determining the nature of these shaded regions (whether or not they exist) remains an open problem.

[2] The intersection graph of a set of curves is a graph $G = (V, E)$, where V is in one-to-one correspondence with the curves, and there exists an edge $(u, v) \in E$ if and only if the curves corresponding to u and v intersect [94].

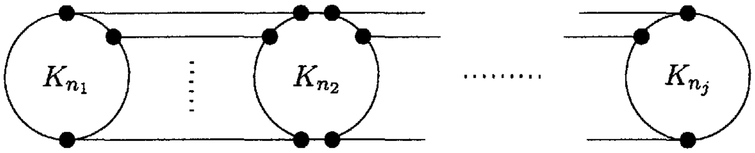


Figure 8-3: The structure of a strip-of-cliques.

We now present a subclass of the co-comparability class that we call a *strip-of-cliques.* A graph is in this class, if it is composed of an ordered set of cliques $1, \dots, j$, where two adjacent cliques i, $i+1$ are connected by any number of disjoint edges, and cliques that are not adjacent are not connected directly. Figure 8-3 illustrates such a graph. Notice that the pair-of-cliques presented in [29] is a specific case of a strip-of-cliques. The following lemmas show that a strip-of-cliques graph satisfies OLoP and that any such graph is a co-comparability graph.

Lemma 8.3.5 *Every strip-of-cliques graph satisfies OLoP.*

Proof: See Appendix 8.E. ■

Lemma 8.3.6 *Every strip-of-cliques graph is a co-comparability graph.*

Proof: See Appendix 8.F. ■

Figure 8-2 depicts the strip-of-cliques class partially overlapping several graph classes. For example, C_4 is chordal bipartite, and can also be viewed as two K_2's connected by parallel links. As another example, consider the graph composed of two K_3's connected by 2 disjoint links. This graph is clearly not bipartite, but is weakly chordal. Finally, consider two K_3's connected by 3 parallel links. This graph is the complement of C_6, denoted $\overline{C_6}$, and consequently not weakly chordal.

Finally, we note that the strip-of-cliques class can be generalized to a larger OLoP-Satisfying class by connecting cliques in a tree structure such that pairs of cliques are connected by any number of disjoint edges, and the intersection graph of the cliques has no cycle. Proving that such a structure satisfies OLoP can be done using similar arguments to the ones used in the proof of Lemma 8.3.5.

We finish this section by providing some context regarding the magnitude of the results. Consider the set of simple graphs having 7 nodes, of which there are 1,044 distinct graphs. Of these graphs, 393 are chordal, and 180 are cographs, with some overlap between these two classes. These numbers can be compared to the 37 forests and 11 trees that were known to satisfy OLoP. Similarly, when considering the set of simple 11 node graphs, the number of chordal graphs is 1,392,387, compared to 710 forests and 235 trees. To summarize, our understanding of the OLoP-Satisfying class has expanded significantly beyond the trees and forest graphs. However, note that the number of chordal graphs is small relative to the total number of simple graphs in this case $(1,018,997,864)$.

8.3.2 Non-perfect graphs

The *OLoP-Satisfying* class includes graphs that are not perfect. We first use the numerical observations of [29] to identify non-perfect graphs that satisfy OLoP. The graph C_5, which

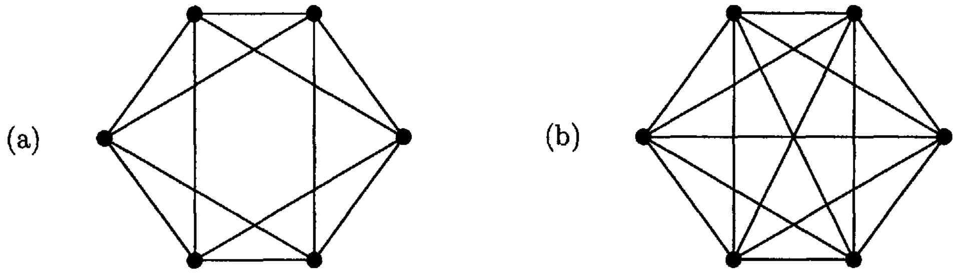


Figure 8-4: (a) 2-hop and (b) 3-hop interference graphs of a 6-ring network graph

is the only non-perfect graph having 5 vertices, satisfies OLoP. Moreover, since all graphs with 6 vertices except C_6 satisfy OLoP, all non-perfect graphs having 6 vertices must satisfy OLoP. Finally, all graphs with 7 vertices satisfy OLoP besides a specific one illustrated in [29, Figure 3] and those that have an induced C_6, which leads us to the observation that *134 out of the 138 non-perfect graphs with* 7 *vertices satisfy OLoP.* In Figure 8-2 all these graphs appear in a single class (containing C_5 and C_7) within the OLoP-Satisfying class.

We now show that all non-perfect graphs that have an induced C_n with n odd and $n \geq 9$ fail OLoP (these are represented as the Odd class in Figure 8-2).

Lemma 8.3.7 *All odd cycles C_n with $n \geq 9$ fail SLoP.*

Proof: See Appendix 8.G. ■

8.4 Local pooling under multihop interference

In this section, we show that counter-intuitively, *more interference often assists the operation of distributed algorithms.* Denote the stability region under k-hop interference by Λ_k^*. It is clear that Λ_k^* cannot increase with k (and often decreases with k), as interference between the links of the network can only increase. Thus, although an increase in k can lead to a smaller stability region, such an increase makes it more likely that the OLoP conditions hold, and thereby more likely that simple distributed algorithms will achieve Λ_k^*.

8.4.1 Interference graphs

We first demonstrate the intuition on which the above observation is based. Consider the network graph C_6 (a 6 node ring), whose interference graph under primary interference is also C_6. According to [51], C_6 *does not satisfy OLoP* and, in general, a MWIS algorithm does not achieve 100% throughput. The best known result then provides that a MWIS algorithm guarantees 50% throughput [90]. Under 2-hop interference, the interference graph has 6 more edges (see Figure 8-4(a)). According to [29], this specific graph satisfies OLoP, and therefore, a MWIS algorithm achieves 100% throughput. Under 3-hop (or higher) interference, the interference graph becomes a clique (see Figure 8-4(b)) which satisfies OLoP [29]. Hence, although under 1-hop interference, a maximal weight algorithm guarantees 50% throughput, under k-hop interference ($k \geq 2$) 100% throughput is guaranteed.

Under k-hop interference, the interference graph becomes an OLoP-Satisfying clique when k equals the network diameter. It seems reasonable to expect that for many network graphs, as the interference degree increases, there exists an *interference threshold* above which OLoP is satisfied. We tested this property by considering small graphs. In [29] it was shown that out of 1,252 simple interference graphs of up to 7 nodes, 14 fail OLoP. The following observation is obtained by exhaustively considering the corresponding k-hop ($k \geq 2$) interference graphs.

Observation 8.4.1 *All k-hop ($k \geq 2$) interference graphs corresponding to network graphs with up to 7 edges satisfy OLoP.*

Applying our acquired knowledge from Section 8.3 regarding the OLoP-Satisfying class, we will now proceed to study multihop interference properties of graphs. We focus on graph classes that appear in Figure 8-2.

First, we indicate that due to Observation 8.4.1, a number of 1-hop interference graphs outside the OLoP-Satisfying class yield k-hop interference graphs that are OLoP-Satisfying. These graphs are the 6-ring, the 6-wheel, and the four non-perfect 7-node graphs outside the OLoP-Satisfying class.

We next introduce the Strongly Chordal class, a subclass of the chordal graphs, which exhibits an interference threshold property.

Definition 8.4.1 (Strongly Chordal [25]) A graph G is *strongly chordal* if G is chordal and each cycle in G of even length at least 6 has an odd chord (a chord (i, j) is odd if the distance in the cycle between i and j is odd).

Denote by G^k the k-th power of G: G^k has the same vertex set V as G, and $u, v \in V$ are adjacent in G^k, if the minimum path length between u and v in G is at most k. Given a 1-hop interference graph G_I^1, the corresponding k-hop interference graph is G_I^k.

Since the strongly chordal graphs belong to the chordal class, Lemma 8.3.1 implies that strongly chordal graphs are OLoP-Satisfying. A property of the the strongly chordal class is that it is *strongly closed under power*. Namely, if an interference graph G_I^k is strongly chordal, then G_I^{k+j} is strongly chordal for all $j \geq 1$ [25]. Therefore, even if the 1-hop interference graph is not strongly chordal, once an interference graph becomes strongly chordal (and thereby OLoP-Satisfying), increased interference degree will generate OLoP-Satisfying graphs. Based on this property, the following theorem establishes that every graph has an interference threshold k^* *above which* all interference graphs satisfy OLoP.

Theorem 8.4.1 *There exists a k^* such that for $k \geq k^*$, G_I^k satisfies OLoP.*

Proof: For any finite interference graph G_I^1, there exists an interference degree at which every component of the interference graph is a clique, which is strongly chordal. The theorem follows, since every strongly chordal graph is strongly closed under power. ■

The following lemmas show that certain graphs, identified in Section 8.3.1, exhibit interference threshold $k^* = 1$ (Lemma 8.4.1 immediately follows from the above mentioned property of the the strongly chordal class).

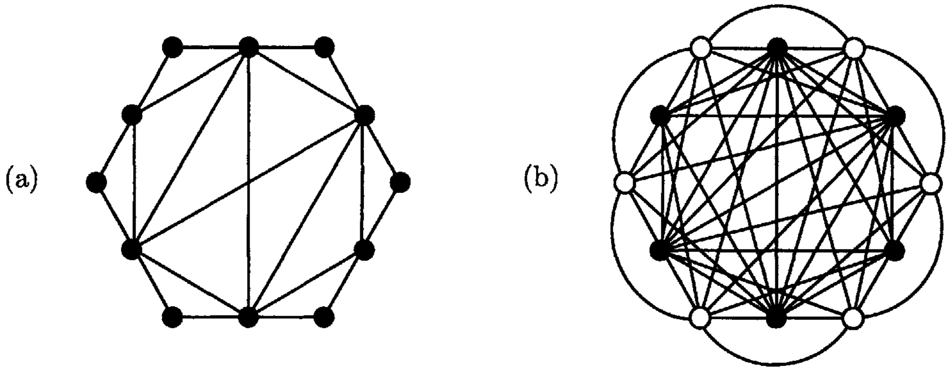


Figure 8-5: (a) A chordal 1-hop interference graph and (b) the corresponding 2-hop interference graph that fails OLoP.

Lemma 8.4.1 *If the 1-hop interference graph G_I^1 is a strongly chordal graph, such as a tree or a clique tree, then G_I^k satisfies OLoP for every $k \geq 1$.*

Lemma 8.4.2 *If the 1-hop interference graph G_I^1 is a cograph, then G_I^k satisfies OLoP for every $k \geq 1$.*

Proof: According to [25] every connected subgraph of a cograph has diameter of at most 2. Therefore, the corresponding $G_I^k \, \forall \, k \geq 2$ is a clique and according to [29] satisfies OLoP. ■

Lemma 8.4.3 *If the 1-hop interference graph G_I^1 is a strip-of-cliques, then G_I^k satisfies OLoP for every $k \geq 1$.*

Proof: See Appendix 8.H. ■

When we study the transition from G_I^k to G_I^{k+1}, we find that there *are* cases where increasing the interference degree can result in a graph that fails OLoP. Namely, although any interference graph has an interference threshold, the transition to this threshold may not be smooth. Namely, below the interference threshold, the interference graphs may alternate between being OLoP-Satisfying and OLoP-Failing for different values of k. The following lemma summarizes this result.

Lemma 8.4.4 There are OLoP-Satisfying k-hop interference graphs for which OLoP is not satisfied in a corresponding j-hop ($j > k$) interference graph.

Proof: Our proof is by example. Consider the 1-hop interference graph G_I in Figure 8-5(a). This is a chordal graph, and therefore, according to Lemma 8.3.1 it satisfies OLoP. The corresponding 2-hop interference graph G_I^2 appears in Figure 8-5(b). The subgraph induced by the white nodes is a 6-ring, which fails SLoP. Therefore, OLoP fails in the 2-hop interference graph. ■

8.4.2 Network graphs

Thus far, we have studied the LoP properties under multihop interference for most graphs represented in Figure 8-2. We next turn our attention to particular *network graph* structures. An example of an interference graph G_I^1 resulting from 1-hop interference is given in Figure 8-1.

A second example is the ring network graph C_n, whose 1-hop interference graph is also C_n. Recall from Section 8.3 that C_n fails OLoP for $n = 6$ and $n \geq 8$. Our numerical tests show that the 2-hop interference graph of any C_n with $n \leq 8$ satisfies OLoP. Hence, we observe that rings are network graphs that benefit from additional interference degrees.

Clearly, any network graph whose corresponding interference graph is one of the structures indicated in lemmas 8.4.1, 8.4.2, and 8.4.3 satisfies OLoP for any $k \geq 1$. In particular, we can derive the following result.

Theorem 8.4.2 *Distributed MWIS algorithms achieve* 100% *throughput in a tree* network graph *under any interference degree* k.

Proof: The interference graph G_I^1 of a tree network graph is a clique tree. According to Lemma 8.4.1 for such an interference graph, the corresponding G_I^k satisfies OLoP for any $k \geq 1$. ∎

The 2-hop interference model is important, since it represents the IEEE 802.11 transmission constraints [15,142,161]. We obtain the following result that applies to this model by using results regarding squares of line graphs[3] studied in [31,32].

Theorem 8.4.3 *Distributed MWIS algorithms achieve 100% throughput in a chordal* network graph *under a k-hop interference model, with any even k.*

Proof: According to [31], given a chordal network graph G_N, the corresponding 2-hop interference graphs G_I^2 is chordal. According to Lemma 8.3.1, OLoP is satisfied in a chordal interference graph, and therefore, distributed MWIS algorithms achieve 100% throughput.

It was shown in [25] that if G^k is chordal, then G^{k+2} is chordal but it is not guaranteed that G^{k+1} is chordal. Therefore, if the 2-hop interference graph G_I^2 is chordal, the corresponding k-hop interference graph G_I^k, with any even k, satisfies OLoP. ∎

Several subclasses of chordal graphs have the potential to allow a MWIS algorithm to be throughput-optimal under a k-hop interference model, with even k. One of the subclasses is the class of interval graphs [25,32]. For that class the following stronger result holds.

Lemma 8.4.5 *Distributed MWIS algorithms achieve 100% throughput in an interval network graph under a k-hop interference model, where $k \geq 2$.*

Proof: According to [32], given an interval network graph G_N, the corresponding 2-hop interference graph G_I^2 is an interval graph. Interval graphs are strongly chordal [25], and therefore, the corresponding k-hop ($k \geq 2$) interference graphs G_I^k are strongly chordal and OLoP-Satisfying. ∎

[3] In graph theoretic terminology, the interference graph resulting from 1-hop interference is called line graph [69].

8.5 Conclusions

The consideration of Local Pooling has the potential to enable efficient distributed operation of wireless networks. However, since previous works focused mostly on deriving the LoP conditions [51] and on networks with primary interference (Chapter 7, in this chapter we focused on the graph implications of the conditions and on multihop interference. We identified several graph subclasses of the OLoP-Satisfying class and increased the number of known graphs that satisfy LoP by a few orders of magnitude. Using these observations, we showed that increasing the interference degree usually has a positive effect on the performance of simple distributed algorithms. For example, it was proved that under *secondary* interference constraints, a maximal weight scheduling algorithm achieves 100% throughput in chordal network graphs.

We emphasize that our objective in this chapter is to obtain a better *theoretical* understanding of LoP that will assist the development of future algorithms. Hence, although a theoretical contribution has been made, there remain many algorithmic open problems. For example, LoP-based algorithms can partition the network into LoP-satisfying subnetworks or add artificial interference constraints to generate a LoP-satisfying network. Our identification of several LoP-satisfying graph classes that can serve as building blocks for these networks, and the understanding of multihop traffic and interference effects are advances toward such algorithms. For instance, one can now develop algorithms that add artificial edges to the interference graph to yield a chordal graph.

Moreover, there are a number of theoretical issues that remain unresolved. For example, Lemma 8.4.4 demonstrates that further study is necessary to determine the general evolution of the LoP property with varying interference degree. Additionally, the complete characterization of the OLoP-Satisfying and the OMLoP-Satisfying graph classes is a subject for further research.

Appendix

8.A Proof of Lemma 8.3.1

It was shown in [94,124] that any graph G that is a chordal graph, or an induced subgraph of a chordal graph, has at least one vertex v for which the vertices in the set $N(v)$ induce a clique in G. Such a vertex is called a simplicial vertex. We claim that G satisfies SLoP. Since the vertices in $N(v)$ induce a clique in G, any maximal independent set in G will include either the simplicial vertex v or exactly one of the vertices in $N(v)$. Consequently, the vector $\boldsymbol{\alpha}$ having all zero entries except at the indices corresponding to the simplicial vertex v and corresponding to the vertices in $N(v)$, where the entries are set to 1, yields $\boldsymbol{\alpha}^T\mathbf{M}(V_I) = \mathbf{e}^T$. Thus, G satisfies SLoP. Since this applies to any chordal graph or an induced subgraph of a chordal graph, it must follow that any chordal interference graph satisfies OLoP.

8.B Proof of Lemma 8.3.2

If graph $B = (V, E)$ is bipartite, the edge $(u, v) \in E$ is called *bisimplicial* if the vertices in $N(u) \cup N(v)$ induce a *complete* bipartite subgraph in B [25]. It was shown in [66] that if graph B is chordal bipartite, any induced subgraph B' of B has a bisimplicial edge. Let (u, v) be a bisimplicial edge of B' and assume that there exists a maximal independent set in B' that does not include either vertex u or v. Such an independent set must include a neighbor of u and a neighbor of v, since otherwise, either u or v could be added to the independent set, which violates that the independent set is maximal. However, since $N(u) \cup N(v)$ induces a *complete* bipartite subgraph, an independent set cannot include vertices from both $N(u)$ and $N(v)$, which provides a contradiction. Therefore, every maximal independent set must include either u or v, but not both vertices. Consequently, the vector $\boldsymbol{\alpha}$ having all zero entries except at the indices corresponding to the vertices of the bisimplicial edge (u, v), where the entries are set to 1, yields $\boldsymbol{\alpha}^T\mathbf{M}(V_I) = \mathbf{e}^T$. Thus, SLoP is satisfied for B'. Since B' is either chordal bipartite or an induced subgraph of a chordal bipartite graph, we must have that OLoP is satisfied for any chordal bipartite interference graph.

8.C Proof of Lemma 8.3.3

In every induced subgraph of a cograph, the intersection of any maximal clique and any maximal independent set contains precisely one vertex [25]. Hence, consider any maximal clique of the graph. By the above property, every maximal independent set of the graph contains precisely one vertex in the clique. The vector $\boldsymbol{\alpha}$ having entries of one at the indices corresponding to nodes in this clique and having entries of zero otherwise, yields $\boldsymbol{\alpha}^T\mathbf{M}(V_I) = \mathbf{e}^T$. Therefore, SLoP holds for all the induced subgraphs of a cograph, which implies that OLoP holds for any cograph.

8.D Proof of Lemma 8.3.4

For the interference graph $C_6 = (V_6, E_6)$, it was shown in [51] that there is no $\boldsymbol{\alpha} \geq 0, c > 0$ such that $\boldsymbol{\alpha}^T\mathbf{M}(V_6) = c\mathbf{e}^T$. Consider $n \geq 8$, with n even. Denote $C_n = (V_n, E_n)$, using node labels $v_1, v_2, \ldots, v_n$. Then, the following are valid maximal independent sets

$$\{v_1, v_3, v_5, \ldots, v_{n-7}, v_{n-4}, v_{n-2}\} \tag{8.1}$$

$$\{v_1, v_3, v_5, \ldots, v_{n-7}, v_{n-4}, v_{n-1}\} \tag{8.2}$$

$$\{v_2, v_4, v_6, \ldots, v_{n-6}, v_{n-4}, v_{n-2}, v_n\} \tag{8.3}$$

$$\{v_2, v_4, v_6, \ldots, v_{n-6}, v_{n-4}, v_{n-1}\} \tag{8.4}$$

$$\{v_2, v_4, v_6, \ldots, v_{n-6}, v_{n-3}, v_{n-1}\} \tag{8.5}$$

$$\{v_2, v_4, v_6, \ldots, v_{n-6}, v_{n-3}, v_n\} \tag{8.6}$$

From the requirement of $\boldsymbol{\alpha}^T\mathbf{M}(V_n) = c\mathbf{e}^T$, we draw the following conclusions. Equations (8.1) and (8.2) imply $\alpha_{n-2} = \alpha_{n-1}$. Combining this fact with (8.3) and (8.4) yields $\alpha_n = 0$. Finally, combining the fact that $\alpha_n = 0$ with (8.5) and (8.6) provides $\alpha_{n-1} = 0$. Thus, it is without loss of generality that we discard the two rows of $\mathbf{M}(V_n)$ corresponding to nodes v_{n-1}, v_n.

We now claim that the remaining rows of $\mathbf{M}(V)$ provide all the constraints corresponding to interference graph C_{n-2}. Consider any maximal independent set of C_n containing node v_1 and node v_{n-1}. Note that this configuration mimics C_{n-2} by disallowing node v_{n-2} to be active simultaneously with v_1. Thus, all maximal independent sets of this type in C_n are maximal in C_{n-2}, and it can be easily seen that all maximal independent sets in C_{n-2} containing v_1 yield maximal independent sets in C_n when v_{n-1} is active. Further, consider any maximal independent set of C_n containing node v_2 and node v_n. Similar reasoning to above provides that all maximal independent sets in C_{n-2} containing v_2 are represented under this configuration. Finally, consider any maximal independent set of C_n containing nodes v_3, v_{n-2}, v_n. Again, it can be easily shown that all maximal independent sets in C_{n-2}containing v_3 and v_{n-2} are represented. This completes the characterization of all maximal independent sets of C_{n-2}, since each independent set in C_{n-2} contains either v_1 or v_2, or contains both v_3 and v_{n-2}. Thus, it must be true that the matrix of maximal independent sets of C_{n-2}, $\mathbf{M}(V_{n-2})$, is a submatrix of that of C_n, $\mathbf{M}(V_n)$.

Since $\alpha_{n-1} = \alpha_n = 0$, the existence of $\boldsymbol{\alpha} \geq 0$ and $c > 0$ such that $\boldsymbol{\alpha}^T\mathbf{M}(V_n) = c\mathbf{e}^T$ implies that

$$(\alpha_1, \ldots, \alpha_{n-2})\mathbf{M}(V_{n-2}) = c\mathbf{e}^T.$$

Applying this reasoning inductively, if the SLoP condition for C_n, where $n \geq 8$ and n is even, is satisfied, then SLoP must be satisfied for C_6. This is a contradiction and we conclude that every C_n fails SLoP for $n \geq 8$ and n even.

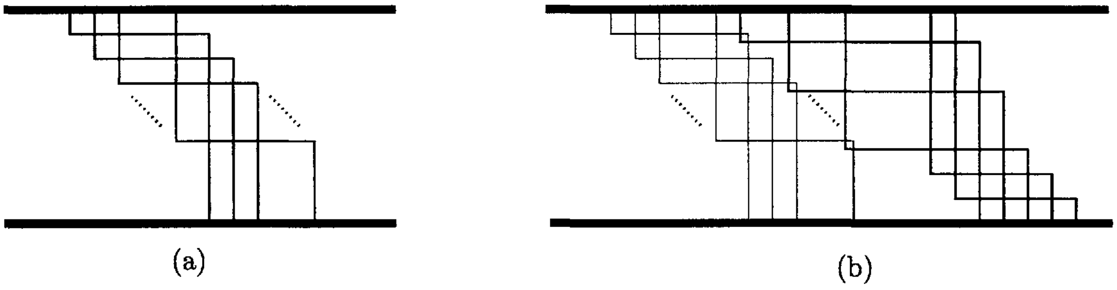


Figure 8-6: Demonstrating that the strip-of-cliques is a co-comparability graph, with (a) a set of curves whose intersection graph is a clique, and (b) the introduction of a neighboring clique, where the curves corresponding to the original clique are thinner than the new ones.

8.E Proof of Lemma 8.3.5

It is clear that every connected induced subgraph of a strip-of-cliques is a strip-of-cliques. If the induced subgraph is disconnected, each component is a strip-of-cliques. According to Proposition 7.4.1, if each component satisfies SLoP, then the combined graph satisfies SLoP. Thus, OLoP is satisfied for any strip-of-cliques if every connected strip-of-cliques satisfies SLOP.

Consider any connected strip-of-cliques graph. If the graph is a clique, then according to Lemma 7.4.2, it satisfies SLoP. Otherwise, designate one of the two cliques that connected to only a single clique as an *edge clique.* For example, in Figure 8-3 K_{n_1} is an edge clique.

If the edge clique includes only a single vertex v, then it is connected by an edge to a vertex u in the neighboring clique. Either u or v must belong to every maximal independent set. Therefore, the vector $\boldsymbol{\alpha}$ having all zero entries except at the indices corresponding to the vertices v and u, where the entries are set to 1, yields $\boldsymbol{\alpha}^T\mathbf{M}(V_I) = \mathbf{e}^T$. If the edge clique includes more than one vertex, exactly one vertices in the edge clique will be active in every maximal independent set. Therefore, the vector $\boldsymbol{\alpha}$ having all zero entries except at the indices corresponding to the vertices of the edge clique, where the entries are set to 1, yields $\boldsymbol{\alpha}^T\mathbf{M}(V_I) = \mathbf{e}^T$. Hence, the connected strip-of-cliques satisfies SLoP, as desired.

8.F Proof of Lemma 8.3.6

According to Definition 8.3.6, if the strip-of-cliques is a co-comparability graph, then each vertex of the strip-of-cliques can be represented as a curve joining two parallel lines. An edge exists between two vertices in the strip-of-cliques if and only if the corresponding curves intersect at some point. We will describe a procedure for constructing the curves that represent an arbitrary strip-of-cliques.

Begin with the leftmost clique, having n_1 vertices. Cascade n_1 curves as shown in Figure 8-6(a), making sure that each of the curves is exposed on the right, in a staircase fashion. Clearly, each of the curves intersects with all others, which implies a clique intersection graph, K_{n_1}.

We next demonstrate how to introduce the i-th clique in the strip-of-cliques, $i \geq 2$. Consider the curves that represent the $(i-1)$-th clique, in order, by descending the staircase on the right. If the vertex v_1 corresponding to one of these curves shares an edge with a vertex v_2 in the i-th clique, then a curve is drawn to represent v_2, by intersecting with the stair corresponding to v_1. This is depicted in Figure 8-6(b), where the first, third, and last curves on the staircase intersect with curves corresponding to the adjacent clique. Any remaining vertices in the i-th clique that do not intersect vertices in the $(i-1)$-th clique are simply included as curves that do not intersect the staircase of the $(i-1)$-th clique. There are two such curves in Figure 8-6(b). Note that the curves corresponding to the i-th clique are once again organized to form a staircase on the right.

This procedure can be repeated iteratively until the entire strip-of-cliques is represented as an intersection graph of curves between two parallel lines. Consequently, the strip-of-cliques is a co-comparability graph.

8.G Proof of Lemma 8.3.7

In Lemma 8.3.4 it was shown by contradiction that every C_n fails SLoP for $n \geq 8$ and n even. The proof for $C_n = (V_n, E_n)$, $n \geq 9$ and n odd is based on a similar idea. First, the matrix of maximal independent sets for $C_9 = (V_n, E_n)$ is characterized:

$$\mathbf{M}(V_9) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Using the same node labeling described above, we study the equation $\boldsymbol{\alpha}^T M(V_9) = c\mathbf{e}^T$. Columns 1 and 2 of $\mathbf{M}(V_9)$ imply $\alpha_7 = \alpha_8$. Columns 2 and 3 imply $\alpha_5 = \alpha_6$. Columns 3 and 4 imply $\alpha_3 = \alpha_4$. Columns 4 and 6 imply $\alpha_1 = \alpha_2$. Columns 6 and 7 imply $\alpha_8 = \alpha_9$. Columns 7 and 8 imply $\alpha_6 = \alpha_7$. Columns 8 and 9 imply $\alpha_4 = \alpha_5$. Columns 9 and 11 imply $\alpha_2 = \alpha_3$. Thus, all values α_i must be equal. But, note that columns 11 and 12 imply $\alpha_5 + \alpha_7 = \alpha_6$, which must give $\alpha_5 = 0$, and consequently $\alpha_i = 0$ for all i. We conclude that C_9 fails SLoP.

The remainder of the proof demonstrating that all rings C_n, for $n \geq 9$ with n odd, fail SLoP follows identically to the even case considered in Lemma 8.3.4, by reducing any such case to the C_9 SLoP condition, which cannot be satisfied.

8.H Proof of Lemma 8.4.3

We adopt similar terminology to that used in the proof of Lemma 8.3.5. According to Lemma 8.3.5, if G_I^1 is a strip-of-cliques, it satisfies OLoP. The interference graph G_I^k is composed of cliques that share some vertices with their neighboring cliques. In particular, consider the maximum clique containing all vertices belonging to the edge clique of G_I^1. We refer to this clique as the k-edge-clique.

If the k-edge-clique equals G_I^k, then clearly G_I^k satisfies OLoP. Otherwise, there are vertices of the G_I^1 edge clique that are not shared with neighboring cliques. In that case, one node of the k-edge-clique will be active in any independent set. Therefore, the vector $\boldsymbol{\alpha}$ having all zero entries except at the indices corresponding to the vertices of the k-edge-clique, where the entries are set to 1, yields $\boldsymbol{\alpha}^T\mathbf{M}(V_I) = \mathbf{e}^T$. Hence, the interference graph G_I^k satisfies SLoP. Using a similar reasoning it can be shown that any subgraph of G_I^k satisfies SLoP, and therefore, G_I^k satisfies OLoP.

Chapter 9

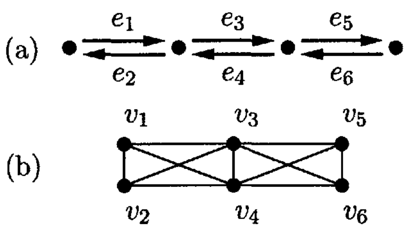
Distributed throughput maximization in wireless networks: Multihop routing

An important challenge in the design and operation of wireless networks is to jointly route packets and schedule transmissions to efficiently share the common spectrum among links in the same area. In Chapter 7 we presented an overview of the work of Dimakis and Walrand [51] where it was shown that there exist network topologies in which distributed *scheduling* algorithms *achieve* 100% *throughput.* In Chapter 8 we studied interference and network graphs that satisfy LoP, thereby deepening our understanding of LoP from the cursory study of Chapter 7. In this chapter we develop sufficient conditions and study topologies in which simple distributed *joint routing and scheduling* algorithms achieve 100% throughput.

9.1 Overview and summary of contributions

Networks with *multihop traffic* have been studied in [160, 161], where it was shown that, in general, only a fraction of the throughput is attainable when using distributed algorithms. Since the LoP results of [51] and Chapters 7 and 8 have been constrained to single-hop traffic, it is desirable to identify specific topologies in which distributed algorithms can obtain 100% throughput in the multihop network setting.

In this chapter, we show that the single-hop LoP conditions introduced in [51] are *insufficient* to guarantee stability in the multihop routing environment. Therefore, we study the LoP properties of a distributed routing and scheduling framework which is based on the backpressure mechanism of [150]. In this framework the edge weights are obtained by the backpressure mechanism but unlike in [150], a *distributed* maximal scheduling algorithm is used to determine which edges should be activated. We derive new multihop LoP conditions that are sufficient for guaranteeing that a distributed joint scheduling and routing mechanism employing maximal weight link activation achieves 100% throughput. Then, we present network topologies that satisfy the multihop LoP conditions, and show that the



(c) $$\mathrm{M}(V_I) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Figure 9-1: (a) Network graph G_N, (b) the corresponding interference graph G_I under primary interference, and (c) the matrix of maximal link activations.

class of topologies satisfying these conditions is strictly included within the class of single-hop LoP-Satisfying graphs. Consequently, the single-hop LoP conditions introduced in [51] are *insufficient* to guarantee stability in the multihop routing environment.

9.2 Network model

We adopt the same network model employed in Chapter 7 with a few changes to aid in our pursuit of multihop properties of Local Pooling. In particular, the wireless network $G_N = (V, E_N)$ is treated in this chapter as a *directed* graph. In G_N, if two nodes $v_1, v_2 \in V$ are within communication range, then the directed edges $e_{12} = (v_1, v_2)$ and $e_{21} = (v_2, v_1)$ both belong to E_N. To clarify the notion of an interference graph when the network graph is a directed graph, we provide in Figure 9-1 a network graph G_N and the corresponding interference graph G_I under primary interference constraints.

9.3 Backpressure-based scheduling and routing

Recall from Algorithm 1 that the optimal centralized scheduler (2.10) makes *maximum* weight service decisions based on *backpressure* link weights. In our framework we consider the distributed *Maximal* Weight Independent Set (MWIS) algorithm used in the single-hop setting, but change the link weights to backpressure link weights. Thus, the MWIS algorithm operates on the interference graph with node weights derived from the backpressure link weights. This enables scheduling decisions for joint link activation and packet routing. As in the single-hop case, which we have considered in Chapters 7 and 8, the framework is *independent of the global network topology and traffic statistics.*

In step 4, the framework uses the MWIS algorithm to select a *maximal* weight link activation based upon maximum link backpressures, obtained in step 3. In step 5, the framework makes routing decisions to service commodities achieving maximum backpressure.

9.4 Multihop local pooling conditions

In this section, we derive the multihop local pooling conditions that are sufficient for stability of the backpressure-based scheduling framework.

Algorithm 16 Backpressure-based MWIS scheduling framework

```
for time index $t = 1, 2, \dots$ do
    For each directed edge $e \in E_N$ assign
        $Z_{ej}(t) \leftarrow (Q_{\sigma(e)j}(t) - Q_{\tau(e)j}(t))$
    Assign $Z_e^*(t) = \max_j Z_{ej}(t)$
    Obtain a maximal link activation $\boldsymbol{\pi}^*(t) \in \Pi_N$ using a decentralized MWIS algorithm, based on the edge weight vector $\mathbf{Z}^*(t) = (Z_e^*(t), e \in E_N)$
    For each $e \in E_N$ such that $\pi_e^*(t) = 1$, choose $j^* \in \arg\max_j Z_{ej}(t)$. Route $\min\{1, Q_{\sigma(e)j^*}(t)\}$ packets of commodity $j^*$ across $e$
end for
```

9.4.1 Preliminaries

Recall that the OLoP conditions consider all possible vertex subsets of the interference graph, $V \subseteq V_I$. By the definition of the interference graph, the node set V corresponds to a subset of the network graph edges, $E \subseteq E_N$. Thus, the OLoP conditions effectively consider every subset of network graph edges $E \subseteq E_N$. In the multihop routing scenario, we must again consider each set of network graph edges $E \subseteq E_N$. Since routing across network graph edges is not unique in the multihop scenario, we must *additionally* consider various combinations of commodities associated with network graph edges. We formalize the possible edge/commodity combinations by introducing the Maximum Commodity Family.

Definition 9.4.1 (Maximum Commodity Family - $\mathcal{J}_E$) *The Maximum Commodity Family for $E \subseteq E_N$, $E \neq \emptyset$, is given by $\mathcal{J}_E = \{(J_e^{\tilde{\mathbf{Q}}}, e \in E_N) : \tilde{\mathbf{Q}} \in \mathcal{Q}_E, \tilde{\mathbf{Q}} \neq 0\}$, where*

$$\mathcal{Q}_E = \{(\tilde{Q}_{ij},\ i,j \in V, i \neq j) : \tilde{Q}_{ij} \in \mathbb{R}_+ \forall i,j,\ E = \arg\max_e \max_j(\tilde{Q}_{\sigma(e)j} - \tilde{Q}_{\tau(e)j})\},$$

$$J_e^{\tilde{\mathbf{Q}}} = \{j \in V : j \neq \sigma(e),\ \tilde{Q}_{\sigma(e)j} - \tilde{Q}_{\tau(e)j} \geq \tilde{Q}_{\sigma(e)j'} - \tilde{Q}_{\tau(e)j'}\ \forall j' \in V\}.$$

The above definition relates closely to the fluid limit model for the queueing system. In order to better understand the Maximum Commodity Family, we next explore some of its properties. To this end, we introduce for each commodity $j \in V$ the directed *commodity graph* $G_j = (V, E_j)$, where $E_j = \{e \in E : j \in J_e\}$.

Lemma 9.4.1 *For $E \subseteq E_N$, $E \neq \emptyset$, the commodity collection $J = (J_e, e \in E_N) \in \mathcal{J}_E$ satisfies:*

1. *$J_e \neq \emptyset$, $\forall e \in E_N$.*
2. *$J_e \subseteq V \setminus \{\sigma(e)\}$.*
3. *For $j \in \cup_{e \in E} J_e$, G_j has no directed cycles.*
4. *If G_j has a directed path between vertices $v_1, v_2 \in V$ of length L, then*

(a) *the minimum length path between v_1 and v_2 in the network graph G_N is L, and*

(b) *the edges of all paths in G_N between v_1 and v_2 of length L are in G_j.*

5. *If G_j has a path of length L originating at vertex v, then*

(a) *G_N has no paths of length less than L originating at vertex v and terminating at vertex j, and*

(b) *the edges of all paths of length L in G_N, originating at vertex v and terminating at vertex j belong to G_j.*

Proof: See Appendix 9.A. ■

Under the backpressure framework, when the set of directed edges $E \subseteq E_N$ have backpressures exceeding those of the other edges in the graph, there must exist a commodity collection $(J_e, e \in E_N) \in \mathcal{J}_E$ for which J_e is the set of commodities maximizing differential backlog across $e \in E_N$. In this case, a MWIS algorithm must select a link activation π^* that is maximal among the edges in E: i.e. $\pi^*_E \in \mathbf{M}(E)$. Additionally, the commodity j that is routed across edge $e \in E_N$ must belong to J_e. These properties characterize the Maximal Service Activation Set (an example is given in Section 9.4.2):

Definition 9.4.2 (Maximal Service Activation Set - $\mathcal{S}_{E,J}$) *For $E \subseteq E_N$ and $J = (J_e, e \in E_N) \in \mathcal{J}_E$,*

$$\mathcal{S}_{E,J} = \left\{ \mathbf{S} \in \mathcal{S} : \sum_j \mathbf{S}_{Ej} \in \mathbf{M}(E), \mathbf{S}_{ej} = 1 \textit{ implies } j \in J_e \textit{ when } e \in E_N \right\}$$

In order to characterize the stability properties of the backpressure framework, we will track the dynamics of the link differential backlogs. Hence, we must understand how each service matrix $\mathbf{S} \in \mathcal{S}$ affects the distribution of commodity backpressures over the network links. We next introduce the Backpressure Service Vector. Recall from Chapter 2 that $d_{ij}(\mathbf{S})$ is the service to queue Q_{ij} under activation matrix $\mathbf{S} \in \mathcal{S}$: $d_{ij}(\mathbf{S}) = \sum_k R^j_{ik} S_{kj}$.

Definition 9.4.3 (Backpressure Service Vector - $\mathbf{u}_{E,J}(\mathbf{S})$) *For $E \subseteq E_N$, $J = (J_e, e \in E_N) \in \mathcal{J}_E$, and service matrix $\mathbf{S} \in \mathcal{S}$, the vector $\mathbf{u}_{E,J}(\mathbf{S}) = (u_{ej}(\mathbf{S}), e \in E, j \in J_e)$ contains the* decrease *in differential backlog of commodity j across link e under service matrix $\mathbf{S}$ for every edge/commodity pair (e, j) where $e \in E, j \in J_e$: $u_{ej}(\mathbf{S}) = d_{\sigma(e)j}(\mathbf{S}) - d_{\tau(e)j}(\mathbf{S})$.*

9.4.2 Some examples

In this section, we consider the network graph G_N of Figure 9-2(a), with the convention that the directed edge from node v_i to v_j is labeled e_{ij}.

We begin by considering a specific feasible combination of edges and commodities. In the next section we will show that certain conditions have to hold for each such combination. The subset E of network edges of interest is $E = \{e_{32}, e_{35}, e_{42}, e_{53}, e_{54}\}$, as depicted in Figure 9-2(b). Each edge in E has associated with it a set of commodities: $J_{e_{32}} = \{v_1, v_2\}$, $J_{e_{35}} = \{v_2\}$, $J_{e_{42}} = \{v_1\}$, $J_{e_{53}} = \{v_1\}$, $J_{e_{54}} = \{v_1\}$. These commodity sets are elements

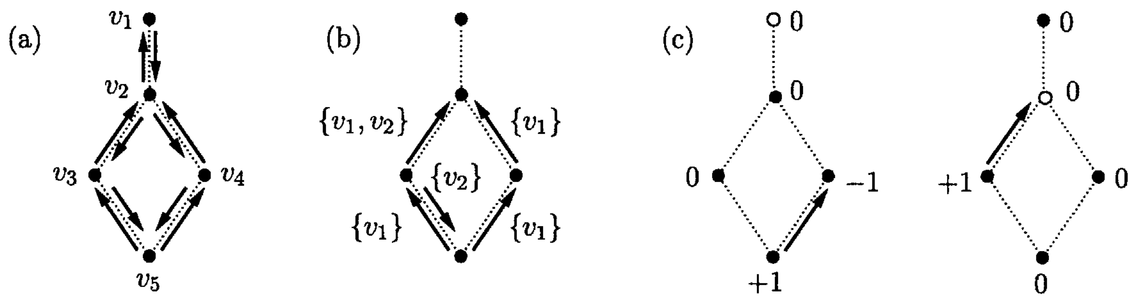


Figure 9-2: (a) Network graph G_N, (b) the subset E of network graph edges, with corresponding commodity sets labeled at each edge, and (c) commodity graphs G_{v_1} (left) and G_{v_2} (right) for a particular maximal service activation.

of commodity collection $J = (J_e, e \in E_N)$. This collection is a member of the Maximum Commodity Family.

Assuming primary interference constraints, the Maximal Service Activation Set $\mathcal{S}_{E,J}$ is summarized by the following table of valid edge/commodity pairs. For example, activation (e_{32}, v_1) means that commodity v_1 is sent over link e_{32}. Additionally, each activation $\mathbf{S}$ is translated in the table below to backpressure service vectors $\mathbf{u}_{E,J}(\mathbf{S})$. The service vectors are ordered by (link, commodity) pairs as follows: $(e_{32}, v_1), (e_{42}, v_1), (e_{53}, v_1), (e_{54}, v_1), (e_{32}, v_2), (e_{35}, v_2)$.

Service activation $\mathbf{S}$	Backpressure service vector $\mathbf{u}_{E,J}(\mathbf{S})$
$\{(e_{32}, v_1), (e_{54}, v_1)\}$	$(2, 0, 0, 2, 0, 0)$
$\{(e_{42}, v_1), (e_{53}, v_1)\}$	$(0, 2, 2, 0, 0, 0)$
$\{(e_{32}, v_2), (e_{54}, v_1)\}$	$(0, -1, 1, 2, 1, 1)$
$\{(e_{35}, v_2), (e_{42}, v_1)\}$	$(1, 2, 0, -1, 1, 2)$

Consider the third service activation from the table, which activates edge e_{32} for service of commodity v_2, and edge e_{54} for service of commodity v_1. We have depicted in Figure 9-2(c) the active link for servicing commodity v_1 packets in the graph on the left, and the active link for servicing commodity v_2 packets in the graph on the right. At each node of the graph, we indicate the number of packets *departed* from that node under that service activation. The backpressure service for each edge/commodity combination (e, j), where $e \in E$ and $j \in J_e$, is then obtained by calculating on the graph corresponding to commodity j the difference between the quantity indicated at the source node of e and that indicated at the destination node of e. Edge e_{54} has a $+1$ at its source and a -1 at its destination in the graph for commodity v_1, which indicates a backpressure service of 2 commodity v_1 packets. Through similar computation, we find that edge e_{32} sees a backpressure service of 1 commodity v_2 packet. Note that although no other edge is active, some inactive edges do incur service under this service activation: edge e_{53} sees a backpressure service of 1 commodity v_1 packet, while edge e_{42} sees an *increase* of commodity

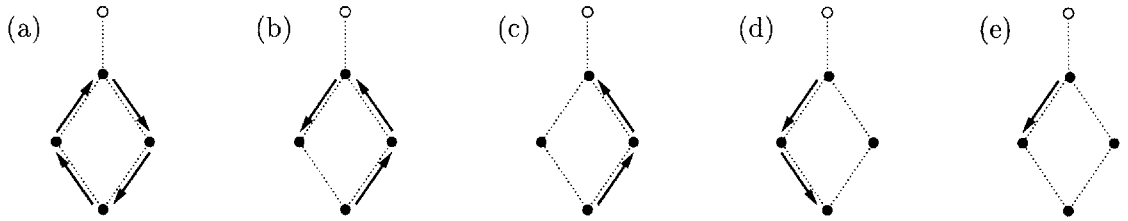


Figure 9-3: Commodity graphs for commodity v_1, that are invalid based on the properties of Lemma 9.4.1.

v_1 backpressure of 1 packet (this implies -1 units of backpressure service). Finally, edge e_{35} sees a service of 1 commodity v_2 packet. No other edge/commodity pairs (e, j) where $e \in E$ and $j \in J_e$, see service. Thus, we have determined each entry in the backpressure service vector corresponding to this particular service activation.

We next provide examples to illustrate the properties of Lemma 9.4.1. Figs. 9-3(a)-9-3(d) show graphs that are inadmissible as the commodity v_1 graph, G_{v_1}, for the network graph depicted in Figure 9-2(a): Figure 9-3(a) fails Property 3 because G_{v_1} contains a directed cycle; Figure 9-3(b) fails Property 4a since edge e_{53} provides a shorter path between vertices v_5, v_3; Figure 9-3(c) fails Property 4b since edges e_{53}, e_{32} are not included in G_{v_1}; Figure 9-3(d) fails Property 5a since the path $v_2 \to v_3 \to v_5$ belongs to G_{v_1}, while path $v_2 \to v_1$ belongs to G_N; and Figure 9-3(e) fails Property 5b since edge e_{21} does not belong to G_{v_1}.

9.4.3 Stability of the backpressure-based framework

Here, we derive new LoP conditions that are sufficient for stability of the backpressure-based scheduling framework. Recall that the quantity $d_{ij}(\mathbf{S})$ is the amount of service at queue Q_{ij} resulting from applying service activation $\mathbf{S}$ for one time slot. Denote vector $\mathbf{d}(\mathbf{S}) = (d_{ij}(\mathbf{S}),\ i, j \in V)$.

Definition 9.4.4 (Subgraph Multihop Local Pooling - SMLoP) *The directed network graph $G = (V, E)$ with commodity collection $J \in \mathcal{J}_E$ satisfies SMLoP if there exist vectors $\boldsymbol{\alpha}, \boldsymbol{\beta} \geq 0$ with $\boldsymbol{\alpha} \neq 0$, and a constant $c \geq 0$ such that*

$$\boldsymbol{\alpha}^T \mathbf{u}_{E,J}(\mathbf{S}) + \boldsymbol{\beta}^T \mathbf{d}(\mathbf{S}) \leq c, \quad \forall \mathbf{S} \in \mathcal{S}, \tag{9.1}$$

$$\boldsymbol{\alpha}^T \mathbf{u}_{E,J}(\mathbf{S}) \geq c, \quad \forall \mathbf{S} \in \mathcal{S}_{E,J}. \tag{9.2}$$

The SMLoP conditions associate with each link/commodity pair (e, j) a non-negative weight $\alpha_{e,j}$, where $e \in E, j \in J_e$. Further, for each node/commodity pair (v, j), the conditions associate a non-negative weight $\beta_{v,j}$, where $v, j \in V$.

Definition 9.4.5 (Overall Multihop Local Pooling - OMLoP) *The network graph $G_N = (V, E_N)$ satisfies OMLoP if SMLoP is satisfied by each subgraph $G'_N = (V, E)$ with commodity collection $J \in \mathcal{J}_E$, where $E \subseteq E_N$.*

We next state the main theorem regarding the stability of the backpressure-based framework.

Theorem 9.4.1 *If network graph G_N satisfies OMLoP, then the MWIS backpressure-based scheduling framework achieves* 100% *throughput.*

Proof: The proof demonstrates that stability can be inferred if there exists no convex combination of backpressure service vectors that exceeds any convex combination of maximal backpressure service vectors for each set E and commodity collection $J_E \in \mathcal{J}_E$. For each $E \subseteq E_N$, $J_E \in \mathcal{J}_E$, this condition can be expressed as a linear program whose dual can be translated to the SMLoP conditions. The full proof can be found in Appendix 9.B. ■

Theorem 9.4.1 demonstrates the sufficiency of the OMLoP conditions for stability under the backpressure-based framework. In the next section, we consider natural questions that arise out of these conditions.

9.5 Studying the OMLoP conditions

We now seek to understand graph properties of the OMLoP conditions. We find that the OMLoP conditions are distinct from the single-hop LoP conditions studied in [51] and Chapters 7 and 8. We also demonstrate stability for a specific class of networks.

9.5.1 OLoP versus OMLoP

We begin by demonstrating that the class of network graphs that are OLoP-Satisfying contains all OMLoP-Satisfying graphs.

Lemma 9.5.1 *If G_N fails OLoP, then it also fails OMLoP.*

Proof: See Appendix 9.C. ■

In terms of Figure 8-2, Lemma 9.5.1 implies that the class of graphs that are not OLoP-Satisfying can not contain OMLoP-Satisfying graphs. Namely, all network graphs having interference graphs with induced subgraphs that are bipartite and not weakly chordal, or induced C_n when $n = 6$ or $n \geq 8$ must fail OMLoP.

The next theorem demonstrates that the OMLoP conditions are in fact *more restrictive* than their single-hop counterparts. Thus, the family of OMLoP-satisfying graphs is *strictly* smaller than that depicted in Figure 8-2. It was indicated in Section 8.3.2 that C_4 satisfies the single-hop OLoP conditions. Here we show that OMLoP fails for C_4.

Theorem 9.5.1 C_5 *(the 5-ring) fails OMLoP.*

Proof: See Appendix 9.D. ■

9.5.2 Graph classes

We now verify that the OMLoP conditions hold for a class of graphs in which the backpressure-based framework is known to achieve 100% throughput. This class is the *forest of stars,* where every connected component of the network graph is a star graph, consisting of a central node v_0, connected to one or more vertices of degree 1. Under any k-interference model, the star's interference graph is a clique (appearing in Figure 8-2 within the intersection region of the chordal and cograph classes). Therefore, only one edge can ever be active at once. Accordingly, a maximal weight edge activation is identical to a *maximum* weight edge activation, thereby achieving 100% throughput. The following lemma shows that OMLoP is satisfied in such graphs.

Lemma 9.5.2 *The star network graph satisfies OMLoP.*

Proof: See Appendix 9.E. ■

Applying the multihop analogous result to Proposition 7.4.1, we have the following corollary.

Corollary 9.5.1 *Every forest of stars satisfies OMLoP.*

In Chapter 8, we completely characterized the LoP properties of cycle graphs, C_n for $n \geq 3$. By Lemma 9.5.1, we can conclude that under primary interference, every network graph C_n, where $n = 6$, or $n \geq 8$ fails OMLoP. The following theorem completes the characterization of the OMLoP properties of network graphs that are cycles.

Theorem 9.5.2 *C_3 is the only cycle network graph satisfying OMLoP under primary interference.*

Proof: The proof that C_4 and C_7 fail OMLoP follows similarly to the proof of Theorem 9.5.1, and is omitted. The proof for C_3 is provided in Appendix 9.F. ■

The above results are reassuring, since clearly maximal weight matching in stars as well as in C_3 provides the *maximum* weight solution. This follows because these graphs yield complete interference graphs. Consequently, Theorem 2.3.1 guarantees that the algorithm achieves 100% throughput. The results of this section do not however provide any indication of the OMLoP properties of general graphs, particularly in cases where maximal weight solutions do not equal the maximum weight solutions. We seek to explore such graphs next.

9.5.3 Exhaustive search

Similarly to Section 7.4.1, we now report the results of numerical studies of the OMLoP conditions. We identified all simple, connected graphs of up to 5 nodes from [156]. We treated each of these graphs as network graphs and investigated their OMLoP properties. We employed Matlab to identify all maximal configurations, (i.e. to obtain matrices $\mathbf{M}(E)$), and to test the OMLoP conditions for each network graph. To test the OMLoP conditions for network graph $G_N = (V, E_N)$, we considered every possible subset of edges $E \subseteq E_N$, as well as every possible commodity collection in $J_E \in \mathcal{J}_E$. We identified commodity collections, by considering any collection $J \in \{(J_e, e \in E_N) : J_e \subseteq V \forall e \in E_N\}$. To ensure that J belongs to $\mathcal{J}_E$, we tested the feasibility of the following linear program. Let

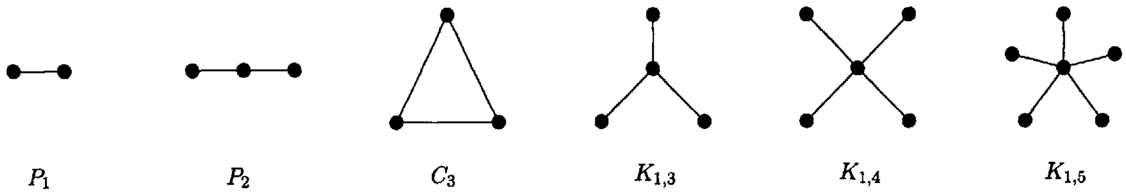


Figure 9-4: The only simple connected network graphs of up to 5 nodes satisfying OMLoP.

$E_j = \{e \in E : e \in J_e\}$.

$$\begin{aligned}
\gamma_j^* = \min \gamma \\
\text{s.t. } & q_{\sigma(e)} - q_{\tau(e)} = \gamma, \quad \forall e \in E_j \\
& q_{\sigma(e)} - q_{\tau(e)} \le \gamma - 1, \quad \forall e \notin E_j \\
& \gamma \ge 1 \\
& q_j = 0 \\
& q_i \ge 0, \quad \forall i
\end{aligned}$$

If the above linear program has a solution for each $j \in V$, then we conclude that the commodity collection J belongs to $\mathcal{J}_E$. This follows because the solutions for all j can be translated to a matrix $\tilde{\mathbf{Q}}$ satisfying the conditions of Definition 9.4.1.

To test the OMLoP conditions, we evaluated for each $E \subseteq E_N$ and $J \in \mathcal{J}_E$ the following linear program.

$$\begin{aligned}
c^* = \max c \\
\text{s.t. } & \sum_{\mathbf{S} \in \mathcal{S}} \mu_{\mathbf{S}} \mathbf{u}_{E,J}(\mathbf{S}) \ge \sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \nu_{\mathbf{S}} \mathbf{u}_{E,J}(\mathbf{S}) + c\mathbf{e} \\
& \mathbf{e}^T \boldsymbol{\mu} \le 1 \\
& \sum_{\mathbf{S} \in \mathcal{S}} \mu_{\mathbf{S}} \mathbf{d}(\mathbf{S}) \ge 0 \\
& \mathbf{e}^T \boldsymbol{\nu} = 1 \\
& \mu_{\mathbf{S}} \ge 0 \quad \forall \mathbf{S} \in \mathcal{S} \\
& \nu_{\mathbf{S}} \ge 0 \quad \forall \mathbf{S} \in \mathcal{S}_{E,J}
\end{aligned}$$

In the proof of Theorem 9.4.1 (specifically in Lemma 9.B.1), it was shown that if $c^* \le 0$ then the network graph $G = (V, E)$ with commodity collection J satisfies SMLoP.

Applying these optimizations in Matlab, the following numerical result was obtained.

Numerical Result 9.5.1 *All connected simple network graphs of up to 5 nodes, subject to primary interference, fail OMLoP, except $P_1, P_2, C_3, K_{1,3}, K_{1,4}, K_{1,5}$, depicted in Figure 9-4.*

This result tells us that the only graph components having at most five nodes that satisfy OMLoP under primary interference are those that have interference graphs that are cliques. Although this result provides a very limited sense of the general OMLoP properties of graphs, it provides a significant indication that the OMLoP-Satisfying class is a very small graph class.

9.6 Conclusions

This chapter came about from the recognition that many networking environments demand the use of multihop routing, particularly in scenarios where direct wireless links between each pair of nodes nodes do not exist. This is clearly the case in many wireless networks, where physical communication impairments, particularly *pathloss*, can lead to arbitrarily interconnected networks.

Consequently, we obtained the LoP conditions for networks with multihop traffic (OMLoP), and showed that they are distinct from the single-hop conditions, derived by Dimakis and Walrand [51]. We showed that the class of graphs satisfying the OMLoP conditions is a strict subclass of the OLoP-Satisfying class.

Much remains to be understood about the OMLoP conditions. The most important question is: Just how restrictive are the OMLoP conditions, and what are the graphs contained within the OMLoP-Satisfying class?

Appendix

9.A Proof of Lemma 9.4.1

Let $E \subseteq E_N$, with $E \neq \emptyset$. Consider any $J_E \in \mathcal{J}_E$, and suppose $J_E = (J_e^{\tilde{\mathbf{Q}}}, e \in E_N)$ for $\tilde{\mathbf{Q}} \in \mathcal{Q}_E$. Item 1 follows because the set $J_e^{\tilde{\mathbf{Q}}}$ can never be empty. Item 2 follows by the definition of $J_e^{\tilde{\mathbf{Q}}}$. For Item 3, suppose that graph G_j contains a directed cycle, $v_1 \to v_2 \to \cdots \to v_L \to v_1$. Then since $\tilde{\mathbf{Q}} \in \mathcal{Q}_E$, it must be true that $\tilde{Q}_{v_i j}$ strictly decreases across each edge in the cycle. This is clearly a contradiction. For Item 4a, suppose vertices v_1, v_2 are joined by a path of length L in G_j, and there exists a shorter path between v_1, v_2 in G_N. Then there must exist an edge e on this shorter path for which $\tilde{Q}_{\sigma(e)j} - \tilde{Q}_{\tau(e)j}$ exceeds the corresponding value across edges in the path joining v_1, v_2 in G_j. This violates that $\tilde{\mathbf{Q}} \in \mathcal{Q}_E$, which provides a contradiction. Item 4b follows similarly: suppose there exist two paths of length L in G_N, with every edge in the first path belonging to G_j. By definition, every edge e in the first path must have equal values $\tilde{Q}_{\sigma(e)j} - \tilde{Q}_{\tau(e)j}$. If this is not the case for the second path, then there must exist some edge e' whose corresponding value exceeds that of the edges in the first path. This violates that $\tilde{\mathbf{Q}} \in \mathcal{Q}_E$, which provides a contradiction. Item 5a follows by noting that $\tilde{Q}_{jj} = 0$, which implies that the differential backlog of commodity j along at least one edge on the shortest path from v to j exceeds that of the edges along the path of length L originating at v. This contradicts the set E. Item 5b follows similarly.

9.B Proof of Theorem 9.4.1

The proof of stability makes use of the *fluid limit* technique. We consider a countably infinite sequence of queueing systems, indexed by r, subject to the same arrival process, $A_{ij}(t), i,j \in \{1,\dots,n\}$, for $t \geq 0$. The queueing variables of the r-th system are given by $Q_{ij}^r(t), A_{ij}^r(t) = A_{ij}(t), U_{ij}^r(t)$ for all $i,j \in \{1,\dots,n\}$, and $F_{\mathbf{S}}^r(t)$ for all $\mathbf{S} \in \mathcal{S}$. At time $t = 0$, the r-th system is assumed to contain zero packets in every queue. The following are the queue evolution properties of the r-th system:

$$\begin{aligned}
Q_{ij}^r(t) &= A_{ij}^r(t) - U_{ij}^r(t), \quad t \geq 0 \\
U_{ij}^r(t) &= \sum_{\mathbf{S}\in\mathcal{S}} d_{ij}(\mathbf{S}) F_{\mathbf{S}}^r(t), \quad t \geq 0 \\
\sum_{\mathbf{S}\in\mathcal{S}} F_{\mathbf{S}}^r(t) &= t, \text{ and } F_{\mathbf{S}} \text{ is non-decreasing}, \quad t \geq 0 \\
A_{ij}^r(0) = 0, U_{ij}^r(0) &= 0, \forall i,j,\ F_{\mathbf{S}}^r(0) = 0, \forall \mathbf{S} \in \mathcal{S}
\end{aligned}$$

We extend the queueing variables to the reals using $Y(t) = Y(\lfloor t \rfloor)$ for $Y = Q_{ij}^r, A_{ij}^r, U_{ij}^r, F_{\mathbf{S}}^r$. Now each of these processes is scaled according to $q_{ij}^r(t) = Q_{ij}^r(rt)/r$. We obtain the scaled processes $q_{ij}^r, a_{ij}^r, u_{ij}^r, f_{\mathbf{S}}^r$. As in [9], we can infer the convergence with probability 1 of the scaled processes over some subsequence of system indices $\{r_k\}$ to a *fluid limit* $(q_{ij}, a_{ij}, u_{ij}, f_{\mathbf{S}})$

having the following key properties:

$$
\begin{aligned}
q_{ij}(t) &= a_{ij}(t) - u_{ij}(t), \quad t \geq 0\\
a_{ij}(t) &= \lambda_{ij} t, \quad t \geq 0\\
u_{ij}(t) &= \sum_{\mathbf{S} \in \mathcal{S}} d_{ij}(\mathbf{S}) f_{\mathbf{S}}(t), \quad t \geq 0\\
\sum_{\mathbf{S} \in \mathcal{S}} f_{\mathbf{S}}(t) &= t, \text{ and } f_{\mathbf{S}} \text{ is non-decreasing}, \quad t \geq 0\\
a_{ij}(0) = 0, u_{ij}(0) &= 0, \forall i, j, \; f_{\mathbf{S}}(0) = 0, \forall \mathbf{S} \in \mathcal{S}
\end{aligned}
$$

The convergence of each process is uniform on compact sets for $t \geq 0$, and it easily follows that the limiting processes $q_{ij}, a_{ij}, u_{ij}, f_{\mathbf{S}}$ are Lipschitz-continuous in $[0, \infty)$.

Consider $z_{ej}(t) = q_{\sigma(e)j}(t) - q_{\tau(e)j}(t)$, the fluid *differential backlog* of commodity j across the directed link e. Define the function $h : [0, \infty) \to [0, \infty)$ where $h(t) = \max_{e,j} z_{ej}(t)$. Consider a regular time[1] $t \geq 0$, at which $h(t) > 0$. Assign

$$E = \{e \in E_N : \exists j \text{ such that } z_{ej}(t) = h(t)\}, \tag{9.3}$$

and for $e \in E_N$, assign $J_e = \arg\max_j z_{ej}(t)$. Note that using $\tilde{\mathbf{Q}} = (q_{ij}(t), i, j \in V_N)$, we have $J \triangleq (J_e, e \in E_N) \in \mathcal{J}_E$. Under the backpressure-based algorithm, it is simple to demonstrate that no link activation outside of $\mathcal{S}_{E,J}$ can have an increasing value $f_{\mathbf{S}}(t)$. Thus we have,

$$\sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t) = 1.$$

Assuming an admissible arrival rate vector $\boldsymbol{\lambda} = (\lambda_{ij}, i, j \in V_N)$, we have for $e \in E$ and $j \in J_e$,

$$
\begin{aligned}
\dot{z}_{e,j}(t) &= \lambda_{\sigma(e)j} - \lambda_{\tau(e)j} - \sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t)(d_{\sigma(e)j}(\mathbf{S}) - d_{\tau(e)j}(\mathbf{S}))\\
&= \sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}}(d_{\sigma(e)j}(\mathbf{S}) - d_{\tau(e)j}(\mathbf{S})) - \sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t)(d_{\sigma(e)j}(\mathbf{S}) - d_{\tau(e)j}(\mathbf{S}))\\
&= \sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}} u_{ej}(\mathbf{S}) - \sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t) u_{ej}(\mathbf{S}) \qquad (9.4)
\end{aligned}
$$

for some $\boldsymbol{\phi} = (\phi_{\mathbf{S}}, \mathbf{S} \in \mathcal{S})$ satisfying $\phi_{\mathbf{S}} \geq 0$, $\sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}} \leq 1$. The following lemma provides a condition under which the fluid differential backlogs are guaranteed to be *non-increasing* at any regular time. Recall our notation that $\mathbf{e}$ denotes the all-ones vector.

Lemma 9.B.1 *Let $t \geq 0$ be a regular time at which $h(t) > 0$. Let $E \subseteq E_N$ satisfy (9.3) and $J_e = \arg\max_j z_{ej}(t)$ for each $e \in E_N$. Suppose that the solution θ^* to the following*

[1] A regular time is a point at which the system is differentiable. By the Lipschitz continuity of the fluid limit, almost every time in $[0, \infty)$ is regular.

optimization problem is $\theta^* \leq 0$:

$$\begin{aligned}
&\textit{Maximize} \quad \theta && (9.5)\\
&\textit{Subject to} \quad \sum_{\mathbf{S}\in\mathcal{S}} \mu_{\mathbf{S}} \mathbf{u}_{E,J}(\mathbf{S}) \geq \sum_{\mathbf{S}\in\mathcal{S}_{E,J}} \nu_{\mathbf{S}} \mathbf{u}_{E,J}(\mathbf{S}) + \theta \mathbf{e} &&\\
&\mathbf{e}^T \boldsymbol{\mu} \leq 1 &&\\
&\sum_{\mathbf{S}\in\mathcal{S}} \mu_{\mathbf{S}} \mathbf{d}(\mathbf{S}) \geq 0 && (9.6)\\
&\mathbf{e}^T \boldsymbol{\nu} = 1 && (9.7)\\
&\mu_{\mathbf{S}} \geq 0 \quad \forall \mathbf{S} \in \mathcal{S} &&\\
&\nu_{\mathbf{S}} \geq 0 \quad \forall \mathbf{S} \in \mathcal{S}_{E,J} && (9.8)
\end{aligned}$$

Then $\dot{h}(t) \leq 0$.

Proof: Suppose $\theta^* \leq 0$. For an admissible arrival rate vector $\boldsymbol{\lambda} = (\lambda_{ij}, i, j \in V_N)$, we have $\lambda_{ij} = \sum_{\mathbf{S}\in\mathcal{S}} \phi_{\mathbf{S}} d_{ij}(\mathbf{S}) \geq 0$, where $\phi_{\mathbf{S}} \geq 0 \forall \mathbf{S}$, and $\sum_{\mathbf{S}\in\mathcal{S}} \phi_{\mathbf{S}} \leq 1$. Furthermore, $\sum_{\mathbf{S}\in\mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t) = 1$ and $\dot{f}_{\mathbf{S}}(t) \geq 0 \forall \mathbf{S}$. Thus, the vectors $(\phi_{\mathbf{S}}, \mathbf{S} \in \mathcal{S})$ and $(\dot{f}_S(t), \mathbf{S} \in \mathcal{S}_{E,J})$ are feasible as vectors $\boldsymbol{\mu}, \boldsymbol{\nu}$ respectively, in the linear program (9.5). The solution $\theta^* \leq 0$ in the optimization clearly implies that there must exist $e \in E$ and $j \in J_e$ such that

$$\sum_{\mathbf{S}\in\mathcal{S}} \phi_{\mathbf{S}} u_{ej}(\mathbf{S}) - \sum_{\mathbf{S}\in\mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t) u_{ej}(\mathbf{S}) \leq 0. \tag{9.9}$$

By (9.4), equation (9.9) implies that $\dot{z}_{ej}(t) \leq 0$. Since t is a regular time, $\dot{z}_{ej}(t) = \dot{h}(t)$, which provides $\dot{h}(t) \leq 0$, as desired. ■

It only remains to demonstrate that the multihop local pooling conditions (9.1)-(9.2) are sufficient for stability. The following lemma demonstrates this property by studying the dual optimization problem to that in (9.5).

Lemma 9.B.2 *Consider graph $G = (V_N, E)$, where $E \subseteq E_N$. Then G satisfies SMLoP under commodity collection $J \in \mathcal{J}_E$ if and only if the corresponding optimization problem (9.5) has solution $\theta^* \leq 0$.*

Proof:

Suppose that the optimization (9.5) has solution $\theta^* \leq 0$. This implies that there exists a dual solution and complementary slackness conditions hold. It is a simple exercise to demonstrate that the dual problem to (9.5) is:

$$\begin{aligned}
&\text{Minimize} \quad c_1 + c_2 && (9.10)\\
&\text{Subject to} \quad \boldsymbol{\alpha}^T \mathbf{u}_{E,J}(\mathbf{S}) + \boldsymbol{\beta}^T \mathbf{d}(\mathbf{S}) \leq c_1, \quad \forall \mathbf{S} \in \mathcal{S} &&\\
&\boldsymbol{\alpha}^T \mathbf{u}_{E,J}(\mathbf{S}) \geq -c_2, \quad \forall \mathbf{S} \in \mathcal{S}_{E,J} &&\\
&\mathbf{e}^T \boldsymbol{\alpha} = 1 &&\\
&\boldsymbol{\alpha}, \boldsymbol{\beta}, c_1 \geq 0 &&
\end{aligned}$$

Since the solution to (9.5) is $\theta^* \leq 0$, the dual solution is attained at the point $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, c_1^*, c_2^*)$, where $c_1^* + c_2^* \leq 0$. Then the values $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*, \boldsymbol{\beta} = \boldsymbol{\beta}^*, c = c_1^*$ satisfy the SMLoP conditions, as desired.

Conversely, suppose that the SMLoP conditions are satisfied, with values $(\boldsymbol{\alpha}, \boldsymbol{\beta}, c) \geq 0$, where $\boldsymbol{\alpha} \neq 0$. Then, the point $(\boldsymbol{\alpha}/(\mathbf{e}^T\boldsymbol{\alpha}), \boldsymbol{\beta}, c, -c)$ is a feasible point in the dual optimization problem (9.10). This feasible point has cost 0. By duality, this implies that the primal problem must attain a solution $\theta^* \leq 0$, as desired. ■

Combining Lemmas 9.B.1 and 9.B.2, we conclude that if SMLoP is satisfied for any $E \subseteq E_N$, with any commodity collection $J \in \mathcal{J}_E$, then $\dot{h}(t) \leq 0$ for any regular time t at which $h(t) > 0$. Noting that $h(0) = 0$, and applying [49, Lemma 1], Lemma 9.B.1 allows us to conclude that $h(t) = 0$ for almost every $t \geq 0$. This immediately implies that $q_{ij}(t) = 0$ for almost every $t \geq 0$, which gives the rate stability of the backpressure based algorithm. Thus the OMLoP conditions are sufficient for stability, as desired.

9.C Proof of Lemma 9.5.1

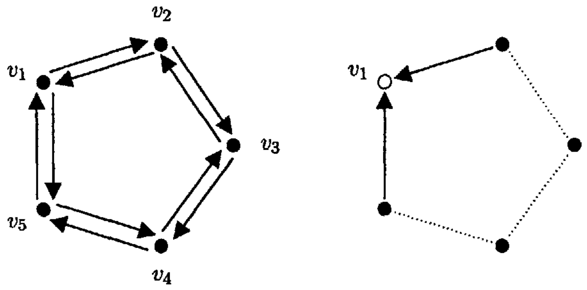
Suppose G_N fails single-hop OLoP. Then, there exists a set of edges E of G_N for which the single-hop SLoP conditions fail. E can be considered without loss of generality as a set of directed edges, each of arbitrary directionality between its end nodes.

To demonstrate that SMLoP fails, consider the set of directed edges E, and commodity sets $J_e = \{\tau(e)\}$ for $e \in E_N$. It can be seen that $J = (J_e, e \in E_N) \in \mathcal{J}_E$. By definition, any active edge in a service activation $\mathbf{S} \in \mathcal{S}_{E,J}$ must be employed for single-hop service. This implies for each $\mathbf{S} \in \mathcal{S}_{E,J}$ that vector $\boldsymbol{\beta}$ can only lead to nonnegative contributions on the lefthand side of (9.1), as follows: each active edge has a value 1 associated with its origin vertex and a value 0 associated with its destination vertex, for the commodity being single-hopped across it. Since we require $\boldsymbol{\beta} \geq 0$, this implies that we can at best treat the second term on the left in (9.1) as zero for every $\mathbf{S} \in \mathcal{S}_{E,J}$.

Thus we must find nonzero $\boldsymbol{\alpha} \geq 0$, $c \geq 0$ such that $\boldsymbol{\alpha}^T\mathbf{u}_{E,J}(\mathbf{S}) = c\mathbf{e}^T$ for each $\mathbf{S} \in \mathcal{S}_{E,J}$. For any such $\mathbf{S}$, each active edge e services a packet to vertex $\tau(e)$, leading to a backpressure reduction across e of a single commodity $\tau(e)$ packet. Because each edge services a different commodity, all inactive edges in E see no change in the backpressure of their respective single-hop commodities. This implies $\mathbf{u}_{E,J}(\mathbf{S}) \in \mathbf{M}(E)$. Since all maximal activations over the edge set E are included in $\mathcal{S}_{E,J}$, the set of backpressure service vectors over $\mathcal{S}_{E,J}$ must then equal $\mathbf{M}(E)$. But $\mathbf{M}(E)$ fails the SLoP conditions: there does not exist nonzero $\boldsymbol{\alpha} \geq 0$, $c > 0$ such that $\boldsymbol{\alpha}^T\mathbf{M}(E) = c\mathbf{e}^T$. Finally, $c = 0$ is invalid, because by its definition as the set of maximal link activations, each row of $\mathbf{M}(E)$ is nonzero, which means the inner product of any nonzero $\boldsymbol{\alpha} \geq 0$ with some column of $\mathbf{M}(E)$ exceeds $c = 0$. Thus G_N fails OMLoP.

9.D Proof of Theorem 9.5.1

Consider the network graph G_N depicted on the left below, and the subset of edges E depicted on the right. We denote by e_{ij} the directed edge from vertex v_i to v_j.



We consider the commodity collection $J = (J_e, e \in E_N)$, where for $e \in E_N$, $J_e = J_e^{\tilde{\mathbf{Q}}}$ and

$$\tilde{\mathbf{Q}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

It can be seen that $\tilde{\mathbf{Q}} \in \mathcal{Q}_E$, which implies that J is a member of the maximum commodity family $\mathcal{J}_E$.

Each of the following edge/commodity activations is represented in the maximal service activation set $\mathcal{S}_{E,J}$:

$$\{(e_{21}, v_1), (e_{45}, v_1)\}, \quad \{(e_{51}, v_1), (e_{32}, v_1)\}.$$

When we consider the backpressure service vectors associated with these activations, the second set of SMLoP conditions (9.2) require the existence of $\boldsymbol{\alpha}, c \geq 0$, $\boldsymbol{\alpha} \neq 0$, such that $\boldsymbol{\alpha}^T \mathbf{M}^1 \geq c$, where

$$\mathbf{M}^1 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Since c is required to be nonnegative, this immediately implies that $c = 0$.

Each of the following edge/commodity activations is represented in the set $\mathcal{S}$:

$$\{(e_{21}, v_1), (e_{54}, v_1)\}, \quad \{(e_{51}, v_1), (e_{23}, v_1)\},$$
$$\{(e_{21}, v_1), (e_{34}, v_1)\}, \quad \{(e_{51}, v_1), (e_{34}, v_1)\},$$
$$\{(e_{32}, v_1), (e_{45}, v_1)\}, \quad \{(e_{32}, v_1)\}, \quad \{(e_{45}, v_1)\}.$$

When we consider the backpressure service vectors and queue backlog service associated with these activations, the first set of SMLoP conditions (9.1) require the existence of

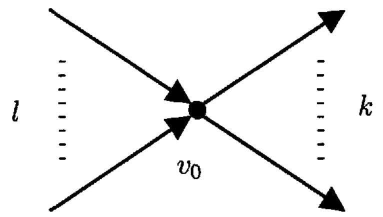


Figure 9-5: Graph G^j of edges carrying commodity j in the commodity collection J_E.

$\boldsymbol{\alpha}, \boldsymbol{\beta} \geq 0$, $\boldsymbol{\alpha} \neq 0$, such that $\boldsymbol{\alpha}^T \mathbf{M}^2 + \boldsymbol{\beta}^T \mathbf{M}^3 \leq 0$, where

$$\mathbf{M}^2 = \begin{bmatrix} 1 & 1 & 1 & 0 & -1 & -1 & 0 \\ 1 & 1 & 0 & 1 & -1 & 0 & -1 \end{bmatrix},$$

$$\mathbf{M}^3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & -1 & -1 & 0 \\ 0 & -1 & 1 & 1 & 1 & 1 & 0 \\ -1 & 0 & -1 & -1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & -1 & 0 & -1 \end{bmatrix}.$$

Simple algebraic manipulation (which we forgo) can be used to demonstrate that there exists no such $\boldsymbol{\alpha}, \boldsymbol{\beta}$. Thus, C_5 fails SMLoP under edge set E and commodity collection J, which implies that C_5 fails OMLoP.

9.E Proof of Lemma 9.5.2

Consider a set of edges $E \subseteq E_N$ and the commodity collection $J = (J_e, e \in E_N) \in \mathcal{J}_E$. Consider the commodity graph $G_j = (V, E_j)$, where $E_j = \{e \in E : j \in J_e\}$. By the definition of $\mathcal{J}_E$, there *can not* exist two oppositely directed edges $(v, v'), (v', v)$ in E_j for all j. Graph $G_j = (V, E_j)$ is a star having $k \geq 0$ edges facing outward from v_0 and $l \geq 0$ edges facing inwards to v_0, with $k + l \geq 1$, as depicted in Figure 9-5.

For the proof, we will use the value $c = 1$. Recall that only a single edge in the star can ever be active at one time. Thus, if we arrange in a matrix the backpressure service vectors corresponding to all $\mathbf{S} \in \mathcal{S}$, the columns of the matrix can be arranged to yield a block diagonal matrix $\mathbf{U}$, with each block corresponding to service activations involving different commodities. We will consider each commodity $j \in \cup_{e \in E} J_e$ in turn and determine the required assignment of the elements of $\boldsymbol{\alpha}$ for j.

Consider commodity $j \in \cup_{e \in E} J_e$:

Case 1. Suppose that $v_0 = j$. Then by the definition of $\mathcal{J}_E$, we must have $k = 0$. In this case, if edge $e \in E_j$ is selected for service of commodity j, link e sees a decrease in backpressure of 1 commodity j packet, and no other of the l links sees a change in backpressure, since the packet departs at v_0. If any other edge not in E_j is selected for service of commodity j packets to v_0, no change in backpressure occurs for any of the l links. Thus, the non-zero component of the block-diagonal matrix corresponding to commodity j is an identity matrix. In this case we assign $\alpha_{e,j} = 1$ for all (e,j) where $e \in E_j$. We also assign $\beta_{v,j} = 0$ for all v.

Case 2. Suppose that $v_0 \neq j$ and that none of the k outward-facing links terminates at node j. In this case, if an outward-facing edge $e \in E_j$ is selected for service of commodity j, e sees a service of 2 units, each of the other outward facing edges in E_j sees a service of 1 unit, and each of the l inward-facing edges sees a service of -1 units. Similarly, if an inward-facing edge $e \in E_j$ is selected for service of commodity j, e sees a service of 2 units, each of the other inward-facing edges in E_j sees a service of 1 unit, and each of the k outward-facing edges sees a service of -1 units. If any other edge e not in E_j is selected for service of commodity j, this leads to a service of 1 at all links facing v_0 in the same direction as e and a service of -1 at all links facing v_0 in the opposite direction to e. The non-zero component of the block-diagonal matrix corresponding to commodity j has the form,

$$\left[\begin{array}{cc|cc} \mathbf{I}_k + \mathbf{e}_{k,k} & -\mathbf{e}_{k,l} & \mathbf{e}_{k,1} & -\mathbf{e}_{k,1} \\ -\mathbf{e}_{l,k} & \mathbf{I}_l + \mathbf{e}_{l,l} & -\mathbf{e}_{l,1} & \mathbf{e}_{l,1} \end{array}\right], \tag{9.11}$$

where $\mathbf{I}_p$ is the identity matrix of size p, and $\mathbf{e}_{p,q}$ is the $p \times q$ matrix of ones. The separator in (9.11) separates the activations in $\mathcal{S}_{E,J}$ (at left) from the remaining commodity j edge activations (at right). The rightmost two columns of (9.11) may or may not exist and there may be multiple copies of either column. Also these columns can dominate other inferior service vectors. In this case, we set $\alpha_{e,j} = (2l+1)/(k+l+1)$ for each $e \in E_j$ facing outwards from v_0, and set $\alpha_{e,j} = (2k+1)/(k+l+1)$ for each $e \in E_j$ facing inwards to v_0. It can be verified that for $k, l \geq 0$ with $k + l \geq 1$, the inner product of $\boldsymbol{\alpha}$ with the leftmost columns before the separator in (9.11) yields 1, while the remaining nonzero columns result in values less than 1. We assign $\beta_{v,j} = 0$ for all v.

Case 3. Suppose one of the k outward-facing links terminates at node j. Through similar analysis as above, we obtain the non-zero component of the block-diagonal matrix corresponding to commodity j as,

$$\left[\begin{array}{ccc|cc} \mathbf{I}_{k'} + \mathbf{e}_{k',k'} & \mathbf{e}_{k',1} & -\mathbf{e}_{k',l} & \mathbf{e}_{k',1} & -\mathbf{e}_{k',1} \\ \mathbf{e}_{1,k'} & 1 & -\mathbf{e}_{1,l} & 1 & -1 \\ \mathbf{e}_{l,k'} & \mathbf{e}_{l,1} & \mathbf{I}_{l,l} + \mathbf{e}_{l,l} & -\mathbf{e}_{l,1} & \mathbf{e}_{l,1} \end{array}\right], \tag{9.12}$$

where $k' = k - 1$. Note that (9.12) only differs from (9.11) in one column to the left of the separator, where the 2 is replaced by a 1. This corresponds to the edge whose destination is j. We assign $\alpha_{e,j} = 2$ for each of the inward-facing links, and $\alpha_{e,j} = (1+2l)/k$ for each of the outward-facing links. In this case, the inner product of $\boldsymbol{\alpha}$ with the first $k-1$ columns of (9.12) yields $1 + (1+2l)/k$, and the remaining columns to the left of the separator yield 1. Since we seek the value $c = 1$, the values $1 + (1+2l)/k$ are too high to satisfy (9.1).

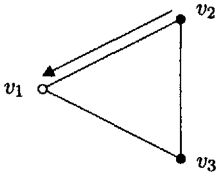
Consequently, we assign $\beta_{v,j} = (1+2l)/k$ for all vertices v terminating the k outward-facing edges. Thus, activation of any one of these edges leads to a contribution of the $\boldsymbol{\beta}$ term in (9.1) of $-(1+2l)/k$, leading to satisfaction of (9.1) as desired.

For every commodity j not belonging to $\cup_{e \in E} J_e$ we assign $\boldsymbol{\alpha}_{e,j} = 0$ for all e, and $\boldsymbol{\beta}_{v,j} = 0$ for all v. The vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are then guaranteed to satisfy SMLoP, as desired. Since this holds for any $E \in E_N$, and any $J \in \mathcal{J}_E$, OMLoP is satisfied.

9.F C_3 satisfies OMLoP

We need only consider each commodity graph individually. By the symmetry of C_3, this yields only three cases to consider.

Case 1. Suppose commodity graph G_j is the following graph.

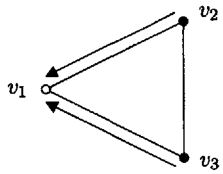


In this case, it is straightforward to demonstrate that the only service activation belonging to $\mathcal{S}_{E,J}$ relevant to this commodity is edge e_21 activated for commodity v_1. The SMLoP condition requires that there exists $\alpha > 0$, $c \geq 0$ such that $\alpha \geq c$. The set $\mathcal{S}$ contains the following edge/commodity activations, which are relevant to this case: $\{(e_{21}, v_1)\}, \{(e_{31}, v_1)\}, \{(e_{23}, v_1)\}, \{(e_{32}, v_1)\}$. Thus, denoting

$$\mathbf{M}_1 = \begin{bmatrix} 1 & 0 & 1 & -1 \end{bmatrix},$$
$$\mathbf{M}_2 = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & -1 & 1 \end{bmatrix},$$

the SMLoP conditions require the existence of $\boldsymbol{\beta} \geq 0$ satisfying $\alpha \mathbf{M}_1 + \boldsymbol{\beta}^T \mathbf{M}_2 \leq c$. Clearly, the values $\alpha = 1, \boldsymbol{\beta} = (0,1), c = 1$ satisfy these conditions.

Case 2. Suppose commodity graph G_j is the following graph.



Here, the relevant service activations belonging to $\mathcal{S}_{E,J}$ for this commodity are: $\{(e_{21}, v_1)\}$, $\{(e_{31}, v_1)\}$. Thus, the SMLoP conditions require the existence of nonzero $\boldsymbol{\alpha} \geq 0$ and $c \geq 0$

such that $\boldsymbol{\alpha}^T\mathbf{M}_3 \geq (c, c)$, where

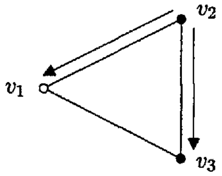
$$\mathbf{M}_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The set $\mathcal{S}$ contains the same relevant edge/commodity activations as in Case 1 above. Denoting

$$\mathbf{M}_4 = \mathbf{M}_5 = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & -1 & 1 \end{bmatrix},$$

the SMLoP conditions require the existence of $\boldsymbol{\beta} \geq 0$ satisfying $\boldsymbol{\alpha}^T\mathbf{M}_4 + \boldsymbol{\beta}^T\mathbf{M}_5 \leq c$. The values $\boldsymbol{\alpha} = (1, 1), \boldsymbol{\beta} = (0, 0), c = 1$ satisfy these conditions.

Case 3. Suppose commodity graph G_j is the following graph.



Here, the relevant service activations belonging to $\mathcal{S}_{E,J}$ for this commodity are: $\{(e_{21}, v_1)\}$, $\{(e_{23}, v_1)\}$. Thus, the SMLoP conditions require the existence of nonzero $\boldsymbol{\alpha} \geq 0$ and $c \geq 0$ such that $\boldsymbol{\alpha}^T\mathbf{M}_6 \geq (c, c)$, where

$$\mathbf{M}_6 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}.$$

The set $\mathcal{S}$ contains the same relevant edge/commodity activations as in Case 1 above. Denoting

$$\mathbf{M}_7 = \begin{bmatrix} 1 & 1 & 0 & -1 \\ 1 & 2 & -1 & -2 \end{bmatrix},$$

$$\mathbf{M}_8 = \begin{bmatrix} 1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 1 \end{bmatrix},$$

the SMLoP conditions require the existence of $\boldsymbol{\beta} \geq 0$ satisfying $\boldsymbol{\alpha}^T\mathbf{M}_7 + \boldsymbol{\beta}^T\mathbf{M}_8 \leq c$. The values $\boldsymbol{\alpha} = (1, 0), \boldsymbol{\beta} = (0, 0), c = 1$ satisfy these conditions.

Chapter 10

Conclusions

We have considered algorithms for scheduling and routing in switched data networks. An important feature of any such network is that there are a finite number of ways in which the network links can be simultaneously activated for transmitting data. Throughput optimal algorithm design for this general network setting was first analyzed by Tassiulas and Ephremides [150]. This thesis, as well as a range of results in the wireless and switching contexts, is a testament to the importance of this model in the design and analysis of modern data networks.

We have applied this networking model to design algorithms and to analyze the performance of optical and wireless networks, and of input-queued switches. Remarkably, though each networking environment potentially leads to a different set of available link activations, our common underlying model implies that each of these environments can be studied in the same framework. This has led us to propose joint WDM reconfiguration and electronic layer routing algorithms for achieving throughput optimality in configurable WDM networks. Building upon the characterization of throughput optimality in the general network setting, we used properties of optical networks to determine analytical performance measures of reconfigurable WDM networks. In the context of input-queued switches, we demonstrated the attractive throughput properties of reduced-complexity scheduling algorithms. For wireless networks, we developed algorithms to allocate links to channels in order to maximize on the achievable throughput under distributed scheduling algorithms. We additionally studied graph properties that are amenable to achieving throughput optimality under distributed schedulers, we determined the implications of interference on the performance of distributed schedulers, and we determined conditions for the throughput optimality of distributed joint scheduling and routing algorithms in wireless networks.

References

[1] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A multi-radio unification protocol for IEEE 802.11 wireless networks. In *Proc. Broadnets'04*, Oct. 2004.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows.* Prentice Hall, 1993.

[3] M. Ajmone Marsan, A. Bianco, P. Giaccone, E. Leonardi, and F. Neri. Multicast traffic in input-queued switches: Optimal scheduling and maximum throughput. *IEEE/ACM Trans. Netw.*, 3(11):465–477, Jun. 2003.

[4] M. Ajmone Marsan, A. Bianco, P. Giaccone, E. Leonardi, and Fabio Neri. Packet-mode scheduling in input-queued cell-based switches. *IEEE/ACM Trans. Netw.*, 10(5):666–678, Oct. 2002.

[5] M. Ajmone Marsan, P. Giaccone, E. Leonardi, and F. Neri. On the stability of local scheduling policies in networks of packet switches with input queues. *IEEE J. Select. Areas Commun.*, 21(4):642–655, May 2003.

[6] M. Ajmone Marsan, E. Leonardi, M. Mellia, and F. Neri. On the stability of isolated and interconnected input-queueing switches under multiclass traffic. *IEEE Trans. Inform. Theory*, 45(3):1167–1174, Mar. 2005.

[7] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, Mar. 2005.

[8] M. Alicherry, R. Bhatia, and L. E. Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *Proc. ACM MOBICOM'05*, Sep. 2005.

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

[10] M. Andrews and L. Zhang. Achieving stability in networks of input-queued switches. *IEEE/ACM Trans. Netw.*, 11(5):848–857, Oct. 2003.

[11] H. Anton. *Elementary Linear Algebra.* John Wiley & Sons, Inc., seventh edition, 1994.

[12] M. Armony and N. Bambos. Queueing dynamics and maximal throughput scheduling in switched processing systems. *Queueing Systems*, 44(3), Jul. 2003.

[13] S. Asmussen. *Applied Probability and Queues.* Springer-Verlag New York, Inc., New York, NY, second edition, 2000.

[14] D. Avis. A survey of heuristics for the weighted matching problem. *Networks*, 13:75–493, 1983.

[15] H. Balakrishnan, C. L. Barrett, V. S. A. Kumar, M. V. Marathe, and S. Thite. The distance-2 matching problem and its relationship to the MAC-layer capacity of ad hoc wireless networks. *IEEE J. Select. Areas Commun.*, 22(6):1069–1079, Aug. 2004.

[16] I. Baldine and G. Rouskas. Traffic adaptive WDM networks: a study of reconfiguration issues. *J. Lightwave Technol.*, 19:433–455, Apr. 2001.

[17] M. Bayati, D. Shah, and M. Sharma. Maximum weight matching via max-product belief propagation. In *International Symposium on Information Theory*, pages 1763–1767, Sep. 2005.

[18] R. Berezdivin, R. Breinig, and R. Topp. Next-generation wireless communications concepts and technologies. *IEEE Commun. Mag.*, 40(3):108–116, Mar. 2002.

[19] R. Berry and E. Modiano. Optimal transceiver scheduling in WDM/TDM networks. *IEEE J. Sel. Areas Commun.*, 23(8):1479–1495, Aug. 2005.

[20] Q. Bi, G. L. Zysman, and H. Menkes. Wireless mobile communications at the start of the 21st century. *IEEE Commun. Mag.*, 39(1):110–116, Jan. 2001.

[21] A. Biance, P. Giaccone, E. Leonardi, F. Neri, and P. Rosa Brusin. Multi-hop scheduling for optical switches with large reconfiguration overhead. In *IEEE Workshop on High performance Switching and Routing (HPSR 2004)*, 2004.

[22] G. Birkhoff. Tres observaciones sobre el algebra lineal. *Rev. Universidad Nacional de Tucuman*, pages 147–150, 1946.

[23] M. Bramson. Stability of two families of queueing networks and a discussion of fluid limits. *Queueing Systems*, 28:7–31, 1998.

[24] M. Bramson. A stable queueing network with unstable fluid model. *Ann. Appl. Probab.*, 9(3):818–853, 1999.

[25] A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph Classes: A Survey.* SIAM, 1999.

[26] A. Brzezinski and E. Modiano. Dynamic reconfiguration and routing algorithms for IP-over-WDM networks with stochastic traffic. *IEEE/OSA J. Lightw. Technol.*, 23(10):3188–3205, Oct. 2005.

[27] A. Brzezinski and E. Modiano. Dynamic reconfiguration and routing algorithms for IP-over-WDM networks with stochastic traffic. In *IEEE Proc. INFOCOM'05*, Miami, FL, Mar. 2005.

[28] A. Brzezinski and E. Modiano. Greedy weigthed matching for scheduling the input-queued switch. In *Proc. CISS'06*, pages 1738–1743, Mar. 2006.

[29] A. Brzezinski, G. Zussman, and E. Modiano. Enabling distributed throughput maximization in wireless mesh networks - a partitioning approach. In *Proc. ACM MOBICOM'06*, Sep. 2006.

[30] E. Buracchini. The software radio concept. *IEEE Commun. Mag.*, 38(9):138–143, Sep. 2000.

[31] K. Cameron. Induced matchings. *Discrete Appl. Math.*, 24(1–3):97–102, 1989.

[32] K. Cameron. Induced matchings in intersection graphs. *Discrete Math.*, 278(1–3):1–9, Mar. 2004.

[33] X. Cao, V. Anand, and C. Qiao. Multi-layer versus single-layer optical cross-connect architectures for waveband switching. In *INFOCOM '04*, pages 1830–1840, Mar. 2004.

[34] C. S. Chang, W. J. Chen, and H. Y. Huang. Birkhoff-von Neumann input buffered crossbar switches. In *IEEE Proc. Information Communications (INFOCOM)*, pages 1614–1623, 2000.

[35] C.-Y. Chang, A. J. Paulraj, and T. Kailath. A broadband packet switch architecture with input and output queueing. In *Proc. GLOBECOM'94*, pages 448–452, 1994.

[36] P. Chaporkar, K. Kar, and S. Sarkar. Throughput guarantees through maximal scheduling in wireless networks. In *Proc. Allerton Conf. on Commun., Control, and Comp.*, Sep. 2005.

[37] H. Chen. Fluid approximations and stability of multiclass queueing networks: work-conserving disciplines. *Ann. Appl. Probab.*, 5(3):637–665, Aug. 1995.

[38] H. Chen and H. Zhang. Stability of multiclass queueing networks under FIFO service discipline. *Math. Oper. Res.*, 22(3):691–725, Aug. 1997.

[39] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle. Optimal cross-layer congestion control, routing and scheduling design in ad hoc wireless networks. In *Proc. IEEE INFOCOM'06*, Apr. 2006.

[40] L. Chen and E. Modiano. Efficient routing and wavelength assignment for reconfigurable WDM networks with wavelength converters. In *Proc. IEEE INFOCOM'03*, Apr. 2003.

[41] L. Chen and E. Modiano. Dynamic routing and wavelength assignment with optical bypass using ring embeddings. *Opt. Switch. Netw.*, 1(1):35–49, Jan. 2005.

[42] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: an approach to high bandwidth optical WAN's. *IEEE Trans. Commun.*, 40(7):1171–1182, 1992.

[43] J. S. Choi, N. Golmie, F. Lapeyrere, F. Mouveaux, and D. Su. A functional classification of routing and wavelength assignment schemes in DWDM networks: static case. In *Proc. OPNET 2000*, pages 1109–1115, Jan. 2000.

[44] S.-T. Chuang, A. Goel, N. McKeown, and B. Prabhakar. Matching output queueing with a combined input/output-queued switch. *IEEE J. Select. Areas Commun.*, 17(6):1030–1039, Jun. 1999.

[45] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms.* The MIT Press, 2001.

[46] J. G. Dai. On the positive Harris recurrence for open multiclass queueing networks: a unified approach via fluid limit models. *Ann. Appl. Probab.*, 5(1):49–77, Feb. 1995.

[47] J. G. Dai. A fluid limit model criterion for instability of multiclass queueing networks. *Ann. Appl. Probab.*, 6(3):751–757, Aug. 1996.

[48] J. G. Dai and S. P. Meyn. Stability and convergence of moments for open multiclass queueing networks via fluid limit models. *IEEE Trans. Automat. Contr.*, 40(11):1889–1904, Nov. 1995.

[49] J. G. Dai and B. Prabhakar. The throughput of data switches with and without speedup. In *Proc. IEEE INFOCOM'00*, Mar. 2000.

[50] N. Devroye, P. Mitran, and V. Tarokh. Limits on communications in a cognitive radio channel. *IEEE Commun. Mag.*, 44(6):44–49, Jun. 2006.

[51] A. Dimakis and J. Walrand. Sufficient conditions for stability of longest queue first scheduling: second order properties using fluid limits. *Adv. Appl. Probab.*, 38(2):505–521, Jun. 2006.

[52] V. Dumas. A multiclass network with non-linear, non-convex, non-monotonic stability conditions. *Queueing Systems*, 25:1–43, 1997.

[53] J. Edmonds. Maximum matching and a polyhedron with (0,1) vertices. *J. of Research of the National Bureau of Standards*, 69B:125–130, 1965.

[54] A. Eryilmaz. *Efficient and Fair Scheduling for Wireless Networks.* PhD thesis, University of Illinois at Urbana-Champaign, 2005.

[55] A. Eryilmaz and R. Srikant. Joint congestion control, routing and MAC for stability and fairness in wireless networks. *IEEE J. Select. Areas Commun.*, 24(8):1514–1524, Aug. 2006.

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

[57] Federal Communications Commission. http://www.fcc.gov/oet/cognitiveradio/.

[58] A. Fumagalli and L. Valcarenghi. IP restoration vs. WDM protection: is there an optimal choice? *IEEE Network*, 14(6):34–41, Nov-Dec 2000.

[59] H. N. Gabow and H. H. Westermann. Forests, frames, and games: algorithms for matroid sums and applications. *Algorithmica*, 7(5&6):465–497, 1992.

[60] O. Gerstel, G. Sasaki, S. Kutten, and R. Ramaswami. Worst-case analysis of dynamic wavelength allocation in optical networks. *IEEE/ACM Trans. Netw.*, 7(6):833–845, Dec. 1999.

[61] N. Ghani, S. Dixit, and T. Wang. On IP-over-WDM integration. *IEEE Commun. Mag.*, pages 72–84, Mar. 2000.

[62] P. Giaccone, E. Leonardi, and F. Neri. On the interaction between tcp-like sources and throughput-efficient scheduling policies. Technical report - Politecnico di Torino, Jul. 2006.

[63] P. Giaccone, E. Leonardi, and D. Shah. On the maximal throughput of networks with finite buffers and its application to buffered crossbars. In *Proc. IEEE INFOCOM'05*, Miami, FL, Mar. 2005.

[64] P. Giaccone, B. Prabhakar, and D. Shah. Randomized scheduling algorithms for high-aggregate bandwidth switches. *IEEE J. Select. Areas Commun.*, 21(4):546–559, May 2003.

[65] P. Giaccone, D. Shah, and B. Prabhakar. An implementable parallel scheduler for input-queued switches. *IEEE Micro*, 22(1):19–25, Jan.–Feb. 2002.

[66] M. C. Golumbic and C. F. Goss. Perfect elimination and chordal bipartite graphs. *J. Graph Theory*, 2:155–163, 1978.

[67] M. C. Golumbic, D. Rotem, and J. Urrutia. Comparability graphs and intersection graphs. *Discrete Math.*, 43(1):37–46, 1983.

[68] B. Hajek and G. Sasaki. Link scheduling in polynomial time. *IEEE Trans. Inform. Theory*, 34(5):910–917, Sep. 1988.

[69] F. Harary. *Graph Theory*. Addison-Wesley, Reading, MA, 1969.

[70] S. Haykin. Cognitive radio: brain-empowered wireless communications. *IEEE Commun. Mag.*, 23(2):201–220, Feb. 2005.

[71] J.-H. Hoepman. Simple distributed weighted matchings. eprint cs.DC/0410047, Oct. 2004.

[72] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. *ACM/Springer Wireless Networks*, 11(4):471–487, Jul. 2005.

[73] A. C. Kam and K. Siu. Linear-complexity algorithms for QoS support in input-queued switches with no speedup. *IEEE J. Select. Areas Commun.*, 17(6):1040–1056, Jun. 1999.

[74] A. C. Kam, K. Siu, R. A. Barry, and E. A. Swanson. A cell switching WDM broadcast LAN with bandwidth guarantee and fair access. *J. Lightw. Technol.*, 16(12):2265–2280, Dec. 1998.

[75] K. Kar, D. Stiliadis, T.V. Lakshman, and L. Tassiulas. Scheduling algorithms for optical packet fabrics. *IEEE J. Sel. Areas Commun.*, 21(7):1143–1155, Sep. 2003.

[76] I. Keslassy and N. McKeown. Analysis of scheduling algorithms that provide 100% throughput in input-queued switches. In *Proc. 39th Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2001.

[77] M. Kodialam and T. Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *Proc. ACM MOBICOM'05*, Sep. 2005.

[78] C. E. Koksal. *Providing Quality of Service over High Speed Electronic and Optical Switches.* PhD thesis, MIT, 2003.

[79] C. E. Koksal, R. G. Gallager, and C. Rohrs. Rate quantization and service quality for variable rate traffic over single crossbar switches. In *Proc. INFOCOM'04*, Mar. 2004.

[80] P. Krishna, N. S. Patel, A. Charny, and R. J. Simcoe. On the speedup required for work-conserving crossbar switches. *IEEE J. Select. Areas Commun.*, 17(6), Jun. 1999.

[81] P. R. Kumar and S. P. Meyn. Stability of queueing networks and scheduling policies. *IEEE Trans. Automat. Contr.*, 40(2):251–260, Feb. 1995.

[82] S. Kumar, P. Giaccone, and E. Leonardi. Rate stability of stable marriage scheduling algorithms in input-queued switches. In *Proceedings Allerton Conf. Communication, Control, and Computing*, Oct. 2002.

[83] J.-F. P. Labourdette and A. S. Acampora. Logically rearrangeable multihop lightwave networks. *IEEE Trans. Commun.*, 39:1223–1230, Aug. 1991.

[84] J.-F. P. Labourdette, F. W. Hart, and A. S. Acampora. Branch-exchange sequences for reconfiguration of lightwave networks. *IEEE Trans. Commun.*, 42:2822–2832, Oct. 1994.

[85] E. L. Lawler. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart and Winston, New York, 1976.

[86] E. Leonardi, M. Mellia, M. Ajmone Marsan, and F. Neri. Joint optimal scheduling and routing for maximum network throughput. In *Proc. IEEE INFOCOM'05*, Miami, FL, 2005.

[87] E. Leonardi, M. Mellia, F. Neri, and M. Ajmone Marsan. Bounds on average delays and queue size averages and variances in input-queued cell-based switches. In *Proc. IEEE INFOCOM'01*, 2001.

[88] E. Leonardi, M. Mellia, F. Neri, and M. Ajmone Marsan. On the stability of input-queued switches with speed-up. *IEEE/ACM Trans. Netw.*, 9(1):104–118, Feb. 2001.

[89] X. Lin and S. Rasool. Constant-time distributed scheduling policies for ad hoc wireless networks. technical report, Purdue University, 2006.

[90] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks. *IEEE/ACM Trans. Netw.*, 14(2):302–315, Apr. 2006.

[91] X. Lin, N. B. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. *IEEE J. Select. Areas Commun.*, 24(8):1452–1463, Aug. 2006.

[92] R. Madan and D. Shah. Capacity-delay scaling in arbitrary wireless networks. In *Proceedings Allerton Conf. Communication, Control, and Computing*, Sep. 2005.

[93] J. E. Marsden and M. J. Hoffman. *Elementary Classical Analysis.* W. H. Freeman and Company, second edition, 2000.

[94] T. A. McKee and F. R. McMorris. *Intersection Graph Theory.* SIAM, 1999.

[95] N. McKeown. *Scheduling Algorithms for Input-Queued Cell Switches.* PhD thesis, University of California, Berkeley, 1995.

[96] N. McKeown, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. In *Proc. INFOCOM'96*, pages 296–302, Mar. 1996.

[97] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. *IEEE Trans. Commun.*, 47(8):1260–1267, Aug. 1999.

[98] S. P. Meyn. Transience of multiclass queueing networks via fluid limit models. *Ann. Appl. Probab.*, 5(4):946–957, 1995.

[99] D. L. Mills. Internet time synchronization: the Network Time Protocol. *IEEE Trans. Commun.*, 39(10), Oct. 1991.

[100] J. Mitola. The software radio architecture. *IEEE Commun. Mag.*, 35(5):26–38, May 1995.

[101] J. Mitola and G. Q. Maguire Jr. Cognitive radio: making software radios more personal. *IEEE Pers. Commun.*, 6(4):13–18, Aug. 1999.

[102] E. Modiano and A. Chiu. Traffic grooming algorithms for minimizing electronic multiplexing costs in unidirectional SONET/WDM ring networks. In *Proc. CISS'98*, Princeton, NJ, Feb. 1998.

[103] E. Modiano and P. J. Lin. Traffic grooming in WDM networks. *IEEE Commun. Mag.*, 39(7):124–129, Jul. 2001.

[104] E. Modiano, D. Shah, and G. Zussman. Maximizing throughput in wireless networks via gossiping. In *Proc. ACM SIGMETRICS'06*, Jun. 2006.

[105] T. Moscibroda, R. Wattenhofer, and Y. Weber. Protocol design beyond graph-based models. In *5th Workshop on Hot Topics in Networks (HotNets)*, Irvine, CA, Nov. 2006.

[106] B. Mukherjee. *Optical Communication Networks.* McGraw-Hill, 1997.

[107] J. Musacchio. *Pricing and Flow Control in Communications Networks.* PhD thesis, University of California, Berkeley, 2005.

[108] A. Narula-Tam, P. J. Lin, and E. Modiano. Efficient routing and wavelength assignment for reconfigurable WDM networks. *IEEE J. Select. Areas Commun.*, 20(1):75–88, Jan. 2002.

[109] A. Narula-Tam and E. Modiano. Dynamic load balancing in WDM networks with and without wavelength constraints. *IEEE J. Select. Areas Commun.*, 18:1972–1979, Oct. 2000.

[110] M. Neely, E. Modiano, and C. Rohrs. Dynamic power allocation and routing for time varying wireless networks. In *IEEE Infocom 2003*, San Francisco, Ca, Apr. 2003.

[111] M. Neely, E. Modiano, and C. Rohrs. Dynamic power allocation and routing for time-varying wireless networks. *IEEE J. Sel. Areas Commun.*, 23(1):89–103, Jan. 2005.

[112] M. J. Neely. *Dynamic Power Allocation and Routing for Satellite and Wireless Networks with Time Varying Channels.* PhD thesis, Massachusetts Institute of Technology, 2003.

[113] M. J. Neely. Energy optimal control for time varying wireless networks. *IEEE Trans Inform. Theory*, 52(2), Jul. 2006.

[114] M. J. Neely. Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks. *IEEE J. Select. Areas Commun.*, 24(8):1489–1501, Aug. 2006.

[115] M. J. Neely, E. Modiano, and C. E. Rohrs. Power allocation and routing in multibeam satellites with time varying channels. *IEEE/ACM Trans. Netw.*, 11(1):138–152, Feb. 2003.

[116] L. Noirie, M. Vigoureux, and E. Dotaro. Impact of intermediate grouping on the dimensioning of multi-granularity optical networks. In *OFC*, pages TuG3–1–TuG3–3, Mar. 2001.

[117] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization.* Dover, 1998.

[118] B. Prabhakar and N. McKeown. On the speedup required for combined input and output queued switching. *Automatica*, 35(12):1909–1920, 1999.

[119] C. Qiao. Labeled optical burst switching for IP-over-WDM integration. *IEEE Commun. Mag.*, pages 104–114, Sep. 2000.

[120] S. Ramanathan and E. L. Lloyd. Scheduling algorithms for multihop radio networks. *IEEE/ACM Trans. Netw.*, 1(2):166–177, Apr. 1993.

[121] R. Ramaswami and K. Sivarajan. *Optical Networks: A Practical Perspective.* Morgan Kaufmann, second edition, 2001.

[122] A. Raniwala and T.-C. Chiueh. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *Proc. IEEE INFOCOM'05*, Mar. 2005.

[123] E. M. Reingold and R. Tarjan. On a greedy heuristic for complete matching. *SIAM J. Comput.*, 10(4):676–681, 1981.

[124] D. J. Rose. Triangulated graphs and the elimination process. *J. Math. Anal. Appl.*, 32(3):597–609, 1970.

[125] K. Ross. *Dynamic Scheduling in Queueing Systems with Applications to Communication Networks.* PhD thesis, Stanford University, 2004.

[126] K. Ross and N. Bambos. Projective cone schedules in queueing structures; geometry of packet scheduling in communication network switches. In *Proceedings of Allerton Conference on Communication, Control and Computing*, 2002.

[127] K. Ross, N. Bambos, K. Kumaran, I. Saniee, and I. Widjaja. Dynamic scheduling of optical data bursts in time-domain wavelength interleaved networks. In *High Performance Interconnects*, Aug. 2003.

[128] K. Ross, N. Bambos, K. Kumaran, I. Saniee, and I. Widjaja. Scheduling bursts in time-domain wavelength interleaved networks. *IEEE J. Select. Areas Commun.*, 21(9):1441–1451, Nov. 2003.

[129] W. Rudin. *Principles of Mathematical Analysis.* McGraw-Hill, Inc., third edition, 1976.

[130] A. N. Rybko and A. L. Stolyar. Ergodicity of stochastic processes describing the operation of open queueing networks. *Problems of Information Transmission*, 28:199–220, 1992.

[131] P. Saengudomlert. *Architectural Study of High-Speed Networks with Optical Bypassing.* PhD thesis, Massachusetts Institute of Technology, Sep. 2002.

[132] P. Saengudomlert, E. Modiano, and R. Gallager. Dynamic wavelength assignment for WDM all-optical tree networks. *IEEE/ACM Trans. Netw.*, 13(4):895–905, Aug. 2005.

[133] P. Saengudomlert, E. Modiano, and R. Gallager. On-line routing and wavelength assignment for dynamic traffic in WDM ring and torus networks. *IEEE/ACM Trans. Netw.*, 14(2):330–340, Apr. 2006.

[134] L. Sahasrabuddhe, S. Ramamurthy, and B. Mukherjee. Fault management in IP-over-WDM networks: WDM protection versus IP restoration. *IEEE J. Sel. Areas Commun.*, 20(1):21–33, Jan. 2002.

[135] A. Saleh. Dynamic multi-terabit core optical networks: Architecture, protocols, control and management. CORONET Proposer's Day Presentation, Aug. 2006. DARPA/Strategic Technologies Office.

[136] S. Sarkar, P. Chaporkar, and K. Kar. Fairness and throughput guarantees with maximal scheduling in multihop wireless networks. In *Proc. WiOpt'06*, Apr. 2006.

[137] E. R. Scheinerman and D. H. Ullman. *Fractional Graph Theory.* John Wiley & Sons, Inc., 1997.

[138] D. Shah, P. Giaccone, and B. Prabhakar. Efficient randomized algorithms for input-queued switch scheduling. *IEEE Micro*, 22(1):10–18, Jan.–Feb. 2002.

[139] D. Shah and M. Kopikare. Delay bounds for approximate maximum weight matching algorithms for input-queued switches. In *Proc. INFOCOM'02*, pages 1024–1031, Jun. 2002.

[140] D. Shah and D. Wischik. Optimal scheduling algorithms for input-queued switches. In *IEEE Proc. INFOCOM*, Barcelona, Spain, Mar. 2006.

[141] S. Shakkottai and A. L. Stolyar. Scheduling for multiple flows sharing a time-varying channel: The exponential rule. In Y. M. Suhov, editor, *Analytic Methods in Applied Probability: In Memory of Fridrikh Karpelevich*, Amer. Mathematical Soc. Translations–Series 2. 2002.

[142] G. Sharma, R. R. Mazumdar, and N. B. Shroff. On the complexity of scheduling in wireless networks. In *Proc. ACM MOBICOM'06*, Sep. 2006.

[143] J. Simmons and A. Saleh. Quantifying the benefit of wavelength add-drop in wdm rings with distance-independent and dependent traffic. *IEEE/OSA J. Lightw. Technol.*, 17(1):48–57, Jan. 1999.

[144] D. Son, B. Krishnamachari, and J. Heidemann. Experimental study of concurrent transmission in wireless sensor networks. In *ACM SenSys'06*, Boulder, CO, Oct. 2006.

[145] A. Stolyar. On the stability of multiclass queueing networks: A relaxed sufficient condition via limiting fluid processes. *Markov Processes and Related Fields*, 1(4):491–512, 1995.

[146] A. Stolyar. Maxweight scheduling in a generalized switch: state space collapse and workload minimization in heavy traffic. *Ann. Appl. Probab.*, 14(1):1–53, Jan. 2004.

[147] D. Stoyan. *Comparison Methods for Queues and other Stochastic Models.* John Wiley, 1983.

[148] L. Tassiulas. Scheduling and performance limits of networks with constantly varying topology. *IEEE Trans. Inform. Theory*, 43(5):1067–1073, May 1997.

[149] L. Tassiulas. Linear complexity algorithms for maximum throughput in radio networks and input queued switches. In *Proc. INFOCOM'98*, pages 533–539, 1998.

[150] 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.*, 37(12):1936–1948, Dec. 1992.

[151] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Trans. Inform. Theory*, 39(3):466–478, Mar. 1993.

[152] J. von Neumann. A certain zero-sum two-person game equivalent to the optimal assignment problem. *Contributions to the Theory of Games, vol. 2*, pages 5–12, 1953.

[153] J. Y. Wei. Advances in the management and control of optical Internet. *IEEE J. Sel. Areas Commun.*, 20(4):768–785, May 2002.

[154] G. Weichenberg, V. W. S. Chan, and M. Medard. On the capacity of optical networks: A framework for comparing different transport architectures. Technical Report LIDS-P-2655, MIT LIDS, Jul. 2005.

[155] G. Weichenberg, V. W. S. Chan, and M. Medard. On the capacity of optical networks: A framework for comparing different transport architectures. In *IEEE Proc. Information Communications (INFOCOM)*, Apr. 2006.

[156] E. W. Weisstein. Connected graph. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/ConnectedGraph.html.

[157] D. B. West. *Introduction to Graph Theory.* Prentice Hall, 1996.

[158] I. Widjaja, I. Saniee, R. Giles, and D. Mitra. Light core and intelligent edge for a flexible, thin-layered and cost-effective optical transport network. *IEEE Commun. Mag.*, 41:S30–S36, May 2003.

[159] A. Wiesler and F. K. Jondral. A software radio for second- and third-generation mobile systems. *IEEE Trans. Veh. Technol.*, 51(4):738–748, Jul. 2002.

[160] X. Wu and R. Srikant. Regulated maximal matching: a distributed scheduling algorithm for multi-hop wireless networks with node-exclusive spectrum sharing. In *Proc. IEEE CDC-ECC'05*, Dec. 2005.

[161] X. Wu and R. Srikant. Bounds on the capacity region of multi-hop wireless networks under distributed greedy scheduling. In *Proc. IEEE INFOCOM'06*, Apr. 2006.

[162] C. Xin and C. Qiao. A comparative study of OBS and OFS. In *IEEE/OSA Optical Fiber Conference (OFC)*, pages ThG7–1–ThG7–3, 2001.

[163] E. M. Yeh and A. S. Cohen. Throughput and delay optimal resource allocation in multiaccess fading channels. In *Proc. International Symposium on Information Theory (ISIT)*, Yokohama, Japan, May 2003.

[164] Y. Yi and S. Shakkottai. Hop-by-hop congestion control over a wireless multi-hop network. In *Proc. IEEE INFOCOM'04*, Mar. 2004.

[165] Y. Yinghua, C. Assi, S. Dixit, and M. A. Ali. A simple dynamic integrated provisioning/protection scheme in IP over WDM networks. *IEEE Commun. Mag.*, 39(11):174–182, Nov. 2001.

[166] M. Yoo, C. Qiao, and S. Dixit. QoS performance of optical burst switching in IP-over-WDM networks. *IEEE J. Sel. Areas Commun.*, 18(10):2062–2071, Oct. 2000.

[167] K. Yoshigoe and K. J. Christensen. An evolution to crossbar switches with virtual output queuing and buffered cross points. *IEEE Network*, 17(5):48–56, Sep.-Oct. 2003.

[168] X. Zhang and C. Qiao. An effective and comprehensive approach to traffic grooming and wavelength assignment in SONET/WDM rings. In *SPIE Proc. Conf. All-Opt. Networking*, number 3531, Boston, MA, Sep. 1998.

[169] G. Zussman and A. Segall. Capacity assignment in bluetooth scatternets - optimal and heuristic algorithms. *ACM/Kluwer Mobile Networks and Applications*, 9(1):49–61, Feb. 2004.