UNIVERSITY OF CALIFORNIA, SAN DIEGO

Throughput Optimal Routing in Wireless Ad-hoc Networks

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering
(Communication Theory and Systems)

by

Hairuo Zhuang

Committee in charge:

Professor Rene L. Cruz, Chair
Professor Robert Bitmead
Professor Philip E. Gill
Professor Tara Javidi
Professor Bhaskar D. Rao

2010

The dissertation of Hairuo Zhuang is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2010

DEDICATION

To my parents.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to first thank my advisor and committee chair Dr. Cruz, for his encouragement and support during my most difficult time. I would never be able to accomplish this work without his help. His kindness, wisdom, research philosophy and inspirational guidance are more than I could ever ask for.

I would also like to thank Prof. Masry and Prof. Rao for training me patiently to think and write engineering problems in a rigorous way during my early Ph.D. years. I am thankful to Prof. Javidi who led me into this exciting research topic and offered great training and support. I am also thankful to Prof. Bitmead and Prof. Gill for taking valuable time to serve in my committee.

I would like to thank Prof. Fedja Nazarov and Prof. Justin Roberts for their patient email support on geometric measure theory and differential topology. It is a pity that the these results never got published. Thank Prof. John Proakis, Prof. Larry Larson, Dr. James R. Zeidler and Prof. Laurence Milstein for their great advisory when I had hesitation about my career. Also thank the ECE staffs, especially Gennie Miranda and Robert Rome for their excellent administrative support.

I would like to thank my roommates during my seven years of stay in UCSD: Andus, Jay, Alex, Luke, Sheu-sheu, and Shuo. I wish them all the best with their study and careers. I would like to thank my friends: Liwen Yu, Yuzhe Jin, Zheng Wu, Lingyun Guo, and Liang Xiao for their constant support, encouragement and instructive suggestions. Thank mitbbs and numerical anonymous mitbbsers whom I have never met. I had a good time there and it was part of my life.

Finally, I would like to thank to my parents and my sister. Without their love, I would never have gone so far.

Chapter 2 and Chapter 3 are in part reprints of the material in paper: M. Naghshvar, H. Zhuang, and T. Javidi, “A General Class of Throughput Optimal

Routing Policies in Multi-hop Wireless Networks", *submitted for IEEE Transactions on Information Theory*. Chapter 4 and Chapter 5 are in part reprints of material in paper: H. Zhuang and R. Cruz, "Throughput Optimal Routing in Multi-commodity Wireless Ad-hoc Networks", *in preparation*. Appendix B, in full, is a reprint of the material in paper: H. Zhuang, E. Masry and B. Rao, "Optimal Power And Rate Allocation Framework For The Uplink With Individual Average Rate Requirements", *ICASSP*, 2008. The dissertation author was the primary investigator and author of these papers.

VITA AND PUBLICATIONS

2001	B. S. in Electrical Engineering, Tsinghua University, Beijing, China
2003	M. E. in Electrical Engineering, Tsinghua University, Beijing, China
2010	Ph. D. in Electrical Engineering (Communication Theory and Systems), University of California, San Diego

H. Zhuang and R. Cruz, "Throughput Optimal Routing in Multi-commodity Wireless Ad-hoc Networks", *in preparation*.

M. Naghshvar, H. Zhuang, and T. Javidi, "A General Class of Throughput Optimal Routing Policies in Multi-hop Wireless Networks,", *submitted for IEEE Transactions on Information Theory*.

H. Zhuang, E. Masry and B. Rao, "Optimal Power And Rate Allocation Framework For The Uplink With Individual Average Rate Requirements", *ICASSP*, 2008

H. Zhuang, L. Dai, S. Zhou, and Y. Yao, "Low Complexity Per-antenna Rate and Power Control Approach for Closed-loop V-BLAST," *IEEE Transactions on Communications*, vol. 51, Nov. 2003 pp. 1783 - 1787.

H. Zhuang, L. Dai and Y. Yao, "A Spatial Multiplexing Technique Based on Large-scale Fading in Distributed Antenna Systems," *The 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)* 2003.

H. Zhuang, L. Dai, L. Xiao, and Y. Yao, "Spectral Efficiency of Distributed Antenna System with Random Antenna Layout," *IEE Electronics Letters*, vol. 39 no. 6, pp. 495-496, March 20th, 2003.

ABSTRACT OF THE DISSERTATION

Throughput Optimal Routing in Wireless Ad-hoc Networks

by

Hairuo Zhuang

Doctor of Philosophy in Electrical Engineering
(Communication Theory and Systems)

University of California, San Diego, 2010

Professor Rene L. Cruz, Chair

This dissertation considers the problem of routing multi-commodity data over a multi-hop wireless ad-hoc network. The few well-known throughput optimal routing algorithms in literature are all based on backpressure principle, which shows poor delay performance under many network topologies and traffic conditions. In contrast, heuristic routing algorithms which incorporated information of closeness to destination are either not throughout optimal or the thoughts optimality was unknown (e.g. opportunistic routing policy with congestion diversity aka. ORCD). The primary goal of this dissertation is to find routing policies beyond

backpressure type that not only ensure throughput optimality but also maintain satisfactory average delay performance.

In the single commodity scenario, by considering a class of continuous, differentiable, and piece-wise quadratic Lyapunov functions, we propose a large class of throughput optimal routing policies called K policies, which include both backpressure algorithm and ORCD as special cases. The proposed class of Lyapunov functions allow the routing policies to control the traffic along short paths for a large portion of state-space while ensuring a negative expected drift, hence, enabling the design of routing policies with much improved delay performances.

We then extend K-policy to multi-commodity case by considering non-quadric Lyapunov functions. A multi-commodity version of ORCD algorithm is proposed based on the generalized K-policy and is shown to be throughput optimal under mild conditions. Interestingly, the algorithm selects the commodity that has the maximum backlogs ratio instead of the maximum difference of backlogs as in the original backpressure algorithm. Simulation results show that the proposed algorithms have better delay performances in all scenarios we considered.

Chapter 1

Introductions

1.1 Routing in Wireless Ad-hoc Networks

Wireless ad-hoc networks consist of a group of nodes which communicate with each other over a wireless channel without any centralized control. Each node in a wireless ad-hoc network can act as and terminal point (i.e. source and destination nodes) as well as a store-and-forward relay. Source nodes transmit data to their designated destination nodes through a shared wireless channel, with or without the help of intermediate relays. The infrastructureless, dynamic and broadcast nature of wireless ad-hoc networks give rise to many design issues at the network, medium access, and physical layers, which have no counterparts in the wired networks and the cellular networks where each cell has a central base station through which all cell data is transmitted. This, along with the diverse applications of these networks in many different scenarios demand new set of networking strategies to be implemented in order to provide efficient end-to-end communication.

One main research area in wireless ad-hoc networks is routing. The general routing problem is to define a policy which chooses which nodes should next transmit which packet, given the history of past transmissions. The main challenges in designing an efficient routing algorithm include

1) Unreliability of wireless links: Due to fading and noise of wireless channel, transmission over wireless links is not reliable. The outcome of a particular transmission is usually unpredictable. It is usually assumed that certain statistic estimates of the transmission outcome are available for these links.

2) Dynamic of network topology: Due to mobility of nodes, fading and interference, the topology of networks changes much more frequently than wired network. A node may join or drop from the network randomly, and a link may be created or broken. All these make the traditional routing algorithms for wired networks difficult to be implemented in wireless ad-hoc networks.

3) Multiple data flows compete for limited network resources: Since multiple streams of data flows share the same wireless channels and simultaneous transmission is allowed, the transmission from one node causes interference to its neighboring nodes. Due to the broadcast nature of wireless channel, the capacity of a wireless network in interference-limited. It is shown in [1] that the throughput per node decreases in order of $\Theta(\frac{1}{\sqrt{n}})$ as the number of nodes n increases.

4) Lack of centralized control: As a result, it is usually desirable that the policy can be implemented in a distributed manner, so that the transmission decisions can be made locally without knowledge of the rest parts of the network.

The above mentioned difficulty have made designing of an efficient and reliable routing strategy a very challenging task. The last decade has seen a rapid growth of research interests in wireless ad-hoc networks, and routing has received a tremendous amount of research attentions from researchers. Most conventional routing strategies in wireless ad-hoc networks can be categorized as proactive protocols and reactive protocols. Proactive protocols periodically update topology information in the network independent of traffic flow. (See, for example [2][3][4][5][6][7] and [8]). Reactive protocols, on the other hand are algorithms which only update routing tables when a new message arrives. (See, for example [9][10][11][12][13][5] and [14])

1.2 Opportunistic Routing

Opportunistic routing for multi-hop wireless ad-hoc networks has seen recent research interest to overcome deficiencies of conventional routing (see for example, [15][16][17][18] and [19]). Opportunistic routing mitigates the impact of poor wireless links by exploiting the broadcast nature of wireless transmissions and the path diversity. More precisely, the routing decisions are made in an online manner by choosing the next relay based on the actual transmission outcomes as well as a rank ordering of neighboring nodes. The authors in [19] provided a Markov decision theoretic formulation for opportunistic routing. In particular, it is shown that for any given packet and at any relaying epoch, the optimal routing decision, in the sense of minimum cost or hop-count, is to select the next relay node based on an index. This index is equal to the expected cost or hop-count of relaying the packet along the least costly or the shortest feasible path to the destination. Furthermore, this index is computable in a distributed manner and with low complexity using a time-invariant probabilistic description of wireless links and the time-invariant transmission costs or transmission times. As such, [19] provides a unifying framework for different versions of opportunistic routing [15][16][17], where the variations are due to the authors' choices of costs. e.g. for ExOR [17], the cost to be minimized is the expected hop-counts (ETX). When multiple streams of packets are to traverse the network, however, it might be necessary to route some packets along longer paths, if these paths eventually lead to links that are less congested. More precisely, and as noted in [20][21], the above opportunistic routing schemes can potentially cause severe congestion and unbounded delays (see examples given in [21]). In other words, these routing schemes are said to fail to stabilize otherwise stabilizable traffic. In contrast, it is known that a simple routing policy, known as backpressure [22], ensures bounded expected total backlog for all stabilizable arrival rates. Backpressure routing policy is essentially a variant of maximum weight matching (MWM) policy [23][24] originally proposed in package switches implementated in a multi-hop network setup. The routing policy provides throughput optimality without knowledge of the network topology or the traffic rates. In the opportunistic context, diversity backpressure routing (DIVBAR) proposed in [25] and [20] provides an opportunistic generalization of backpressure which incorporates the wireless local transmission diversity.

Note that to ensure throughput optimality, backpressure-based algorithms [22][25] and [20] do something very different from [15]-[19]. Rather than using the metric of closeness to the destination, they choose the forwarder with the largest positive differential queue backlog (routing responsibility is retained by the sender if no such forwarder exists). This very property of ignoring the cost to the destination, however, becomes the bane of this approach, leading to poor delay performance (see [20],[21]). As a consequence, various enhanced versions of backpressure algorithms were proposed recently intended to improve the delay performance of original backpressure algorithm (e.g. [26][27][28] and [20]). However, these policies are still based on the backpressure principle.

In [21], the authors consider the single commodity routing problem, i.e. multiple arrivals from different nodes are destined to a single destination. A routing policy, known as Opportunistic Routing with Congestion Diversity (ORCD) are proposed with an improved delay performance. ORCD combines the congestion information with the shortest path calculations inherent in opportunistic routing [21]. The throughput optimality of ORCD was conjectured in [21] but was left unproven, due to the difficulty of identifying appropriate (and universal) Lyapunov functions with negative expected drift. In fact backpressure [22] and its variants [26][27][28][25][20][29][30][31], with quadratic Lyapunov function, and randomized strategies [32] with an exponential Lyapunov function remain to be the only known throughput optimal routing policies. However, using simple quadratic or exponential Lyapunov function fails to guarantee a negative expected drift under ORCD. Therefore, a more general and complicated construction of Lyapunov function is required, which is a major motivation of this dissertation.

1.3 Dissertation Overview

In this dissertation, we first consider the single commodity routing problem: i.e. multiple arrivals from different nodes are destined to a single destination. We provide a large class of throughput optimal policies called K policy by considering a class of piece-wise quadratic Lyapunov functions. The proposed class of Lyapunov functions allow for the routing policies to control the traffic along short paths for a large portion of state-space while ensuring a negative expected drift, hence, enabling the design of routing policies without many of the deficiencies of backpressure-based algorithms. We also specialize our result to recover the throughput optimality of two known routing policies, backpressure (already known to be throughput optimal) and ORCD (whose throughput optimality only was conjectured in [21]).

We then extend K policy to multi-commodity scenario by considering non-quadratic Lyapunov function. A multi-commodity version of ORCD algorithm is proposed and is shown to be throughout optimal under mild conditions. The algorithm is first proposed under the assumption that CSI (channel state information) is known at the transmitters. We then modify it for the case when only CDI (channel distribution information) is known at the transmitters and an opportunistic implementation is proposed. Interestingly, the proposed algorithms are designed to select the commodity with the maximum ratio of backlogs instead of the maximum difference of backlogs as in the classical backpressure algorithm. Indeed, we show via a counter example that selecting commodity based on the difference of backlog might not be throughput optimal when K policy is used.

The organization of the rest chapters is as follows:

Chapter 2 provides a brief review of some preliminaries results that will be used throughout the dissertation, including the stability of Markov chain, Lyapunov drift criterion for stability, and capacity region of a wireless ad-hoc network.

Chapter 3 is devoted to the single commodity scenario, where multiple arrivals from different nodes are destined to a single destination. A large class of policy called K policy is proposed and shown to be throughput optimal by considering a class of piece-wise quadratic Lyapunov function. Some application of K policy is then discussed. We show that both single commodity backpressure algorithm and ORCD algorithm can be viewed as a special implementation of K policy.

Chapter 4 and Chapter 5 are devoted to the multi-commodity scenario. In Chapter 4, we first propose an generalized form of backpressure algorithm by considering non-quadratic Lyapunov function. As a special case of generalized backpressure algorithm, we propose BP-R as an improvement over classical backpressure algorithm in multi-commodity scenario. Using the similar technique, we extend K policy to multi-commodity scenario and K-R policy is shown to be throughput optimality. Some application of K-R policy is considered in Chapter 5, including both CSI-Tx (channel state information at transmitters) case and CDI-Tx (channel distribution information at transmitters) case. Delay performance of these routing algorithms are simulated and compared.

A brief summary and some concluding remarks are made in Chapter 6.

Chapter 2

System Model and Preliminaries

2.1 System Model

2.1.1 On/Off Network Model

In this work, we consider the on/off network model, which can be regarded as a special case of the more general discrete network model used in [33].

Consider a network with N nodes and transmission link set $\mathcal{L}$. We label the nodes by 1,2,...,N and let $\mathcal{N} = \{1, 2, ..., N\}$ denote the set of nodes. Each transmission link in $\mathcal{L}$ is an ordered node pair, labelled by its corresponding ordered node pair (a, b) $(a, b \in \mathcal{N})$, representing a communication channel for direct transmission from a given node a to another node b. Note that link (a, b) is distinct from link (b, a). We say node a is *able to transmit* to node b if $(a, b) \in \mathcal{L}$. As a convention, we always assume that $(a, a) \in \mathcal{L}$ for any $a \in \mathcal{N}$, since a node can always 'transmit' to itself by actually transmit nothing.

Let $\mathcal{L}_{all} := \{(a, b) : a, b \in \mathcal{N}\}$ represent the set of all ordered pairs of nodes in a network with node set $\mathcal{N}$. In a wireless network, direct transmission between two nodes may or may not be possible and this capability, may change over time due to weather conditions, mobility or node interference. Hence in the most general case one can consider that $\mathcal{L}$ consists of all ordered pairs of nodes, i.e. $\mathcal{L} = \mathcal{L}_{all}$, where the transmission rate of link (a, b) is zero if direct communication is impossible. However, in cases where direct communication between some nodes is never possible, it is helpful to consider that $\mathcal{L}$ is a strict subset of $\mathcal{L}_{all}$.

The network is assumed to operate in slotted time with slots normalized to integral units, so that slot boundaries occur at times $t \in \{0, 1, 2, ...\}$. Hence, time slot t refers to the time interval $[t, t + 1)$. Let $h_{ab}(t)$ be the state process of link (a, b) taking values in $\{0, 1\}$, where $h_{ab} = 1$ if node b is able to successfully receive a packet from node a at time slot t, and $h_{ab}(t) = 0$ otherwise. We say node b can *hear* node a at time t if $h_{ab} = 1$. Note that b can *hear* node a ($h_{ab}(t) = 1$) at time t should not be confused with node a is *able to transmit* to node b ($(a, b) \in \mathcal{L}$). Clearly, b can *hear* node a at time t implies a is *able to transmit* to node b. e.g. if $(a, b) \notin \mathcal{L}$, then $h_{ab}(t) \equiv 0$ for all t. As a convention, we assign $h_{aa}(t) \equiv 1$ for all t and $a \in \mathcal{N}$, which can be interpreted as the fact that a node can always reserve the packet for itself without forwarding to any other nodes at a given time slot. Such an definition is only for technical purpose to simplify some of the writing later.

Define the potential forwarder set of node a at time t as $\mathcal{S}_a(t) := \{b : h_{ab}(t) = 1\}$, which is the set of nodes that can hear node a and successfully receive a packet from node a at time t. Note that the definition of potential forwarder set here is slightly different from other literature such as [21], which doesn't include node a itself. In this work, since $h_{aa}(t) \equiv 1$ for all t, we always have $a \in \mathcal{S}_a(t)$ for any node a.

The topology state process is defined as the joint state process $\boldsymbol{H}(t) = (h_{ab}(t))_{ab}$. For simplicity, we assume in the work that the topology state process $\boldsymbol{H}(t)$ is i.i.d. Note that even though $\boldsymbol{H}(t)$ is i.i.d. implies that the individual state process $h_{ab}(t)$ is i.i.d., the state process of different links can be dependent. It is easy to see that the topology state $\boldsymbol{H}(t)$ can be fully characterized by the potential forwarder sets of all nodes in $\mathcal{N}$, i.e. $\{\mathcal{S}_a(t), a \in \mathcal{N}\}$. Indeed, if we write $\boldsymbol{H}(t)$ as an N by N matrix, $\mathcal{S}_a(t)$ simply corresponds to the a-th row of $\boldsymbol{H}(t)$.

2.1.2 Routing in an On/Off Network

All data that enters the network is associated with a particular commodity, which can be defines the destination of the data, but might also specify other information, such as the source node of the data or its priority service class. Let $\mathcal{K}$ represent the set of commodities in the network.

Let $A_a^c(t)$ represent the amount of new commodity c data that exogenously arrives to source node a during slot t (for all $a \in \mathcal{N}$ and all $c \in \mathcal{K}$). We assume that $A_a^c(t)$ takes units of packets, although it can take other units when appropriate (such as units of bits). We assume $A_a^c(t)$ is bounded, i.e. there exists a constant A_{max} such that

$$A_a^c(t) \leq A_{max} \tag{2.1}$$

for all c, a and t. (2.1) it imposes a limit on the number of packets arrived during a time slot. The assumption in (2.1) always hold in practice since the number of arrival is always bounded due to the physical limitation in a real network.

We assume $A_a^c(t)$ is an i.i.d process with rate $\mathbb{E}[A_a^c(t)] = \lambda_a^c$. Define $\boldsymbol{\lambda} = (\lambda_a^c)_{ac}$ as the matrix of arrival rates. We assume that the input rate matrix is stabilizable, and in particular that is within the relative interior of the capacity region Λ, i.e.there exists a value $\boldsymbol{\epsilon} > 0$ such that $\boldsymbol{\lambda} + \boldsymbol{\epsilon} \in \Lambda$. The notion of stability and network capacity region will be formally defined in the next section.

Each node a maintains a set of internal queues for storing network layer data according to its commodity. Let $Q_a^c(t)$ represent the current backlog, or unfinished work, of commodity c data stored in a network layer queue at node a.

The queue backlog $Q_a^c(t)$ can contain both data that arrived exogenously from the transport layer at node a as well as data that arrived endogenously through network layer transmissions from other nodes. In the special case when node a is the destination of commodity c data, we formally define $Q_a^c(t) = 0$ for all t, so that any data that is successfully delivered to its destination is assumed to exit the network layer.

We assume that all network layer queues have infinite buffer storage space. Our primary goal for this layer is to ensure that all queues are stable. Even thought the infinite buffer assumption is not realistic in practice system, this performance criterion tends to yield algorithms that also perform well when network queues have finite buffers that are sufficiently large.

Let $\boldsymbol{Q}(t) := (Q_a^c(t))_{ac}$ denote the joint queue backlog state. And let $\mathcal{Q}$ denote the space of $\boldsymbol{Q}(t)$. We assume $\mathcal{Q}$ is a discrete set (and hence a countable state space), which is always the case in practical applications as data is always transmitted in a discrete manner (e.g. in unit of bits or packets). This assumption allows us to take advantage of existing results on stability of a Markov decision process in a countable state space.

Figure 2.1 shows an example of a network with 8 nodes and 2 commodities. where commodity 1 data input from node 7 and 8 are destined to node 1, and commodity 1 data input from node 7 and 5 are destined to node 2.

Define $\mu_{ab}^c(t)$ as the routing control variables, representing the amount of commodity c data delivered over link (a, b) during slot t. For our problem, $\mu_{ab}^c(t) \in \{0, 1\}$, i.e. at time t, node a either transmits a packet to node b or doesn't transmit any packet. We assume that only the data currently in node a at the beginning of slot t can be transmitted during that slot. Hence, the slot-to-slot dynamics of the queue backlog $Q_a^c(t)$ satisfies the following inequality:

$$Q_a^c(t+1) \leq \max[Q_a^c(t) - \mu_{a,out}^c(t), 0] + A_a^c(t) + \mu_{a,in}^c(t) \tag{2.2}$$

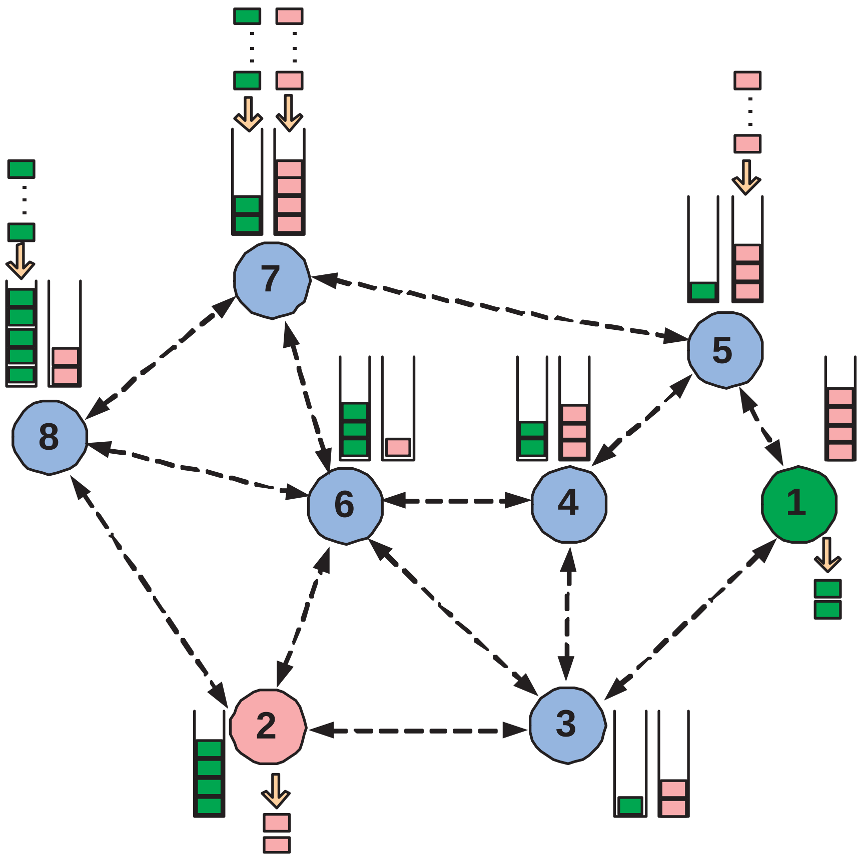

Figure 2.1: Example of a network with 8 nodes and 2 commodities

where $\mu^c_{a,out}(t) := \sum_b \mu^c_{ab}(t)$ and $\mu^c_{a,in}(t) := \sum_a \mu^c_{ai}(t)$. The above expression is an inequality rather than an equality because the actual amount of commodity c data arriving to node a during slot t may be less than $\mu^c_{a,in}(t)$ if the neighboring nodes have no commodity c data to transmit.

Let $\boldsymbol{\mu}(t) := (\mu^c_{ab}(t))_{abc}$ denote the collection of these routing control variables at time t. We refer to $\boldsymbol{\mu}(t)$ as a routing control action at time t, representing the routing decision chosen by the network controller. Due to the network and topology constraints, not all routing control actions are feasible. We assume that the controller chooses the routing control actions $\boldsymbol{\mu}(t)$ subject to the following routing constraints:

$$\sum_{c\in\mathcal{K}} \mu_{ab}^c(t) \leq h_{ab}(t) \tag{2.3}$$

$$\sum_{b}\sum_{c\in\mathcal{K}} \mu_{ab}^c(t) = 1 \tag{2.4}$$

Constraint in (2.3) says that node a can transmit to node b only when node b can hear node a during time slot t ($h_{ab}(t) = 1$). And only a single commodity can be transmitted over a link during a single time slot. Constraint in (2.4) says that at most one packet can be transmitted from any given node during a single time slot. The above constraints can also be rewritten as

$$\sum_{b\in\mathcal{S}_a(t)}\sum_{c\in\mathcal{K}} \mu_{ab}^c(t) = 1 \tag{2.5}$$

$$\mu_{ab}^c(t) = 0, \quad \text{if } b \notin \mathcal{S}_a(t) \tag{2.6}$$

Note that $\mu_{aa}(t) = 1$ implies that node a doesn't forward a packet to any neighboring node during time t . We refer to the routing control action $\boldsymbol{\mu}(t) = (\mu_{ab}^c(t))_{abc}$ that satisfies (2.3) and (2.4) (or equivalently, (2.5) and (2.6)) as an *admissible control action*. Let $\mathcal{M}(\boldsymbol{H})$ denote the set of all admissible routing control actions under topology state $\boldsymbol{H}$.

A network layer routing policy makes decisions about routing, scheduling, and resource allocation in reaction to the topology state process $\boldsymbol{H}(t)$ and queue backlog state process $\boldsymbol{Q}(t)$. Generally, a routing policy $\pi = \{\pi_t\}_t$ is a collection of (potentially randomized) functions taking values among admissible control actions, each corresponds to one time slot and is a function of the network's past history. That is, π_t belongs to the σ-field generated by

$$\{\boldsymbol{Q}(0), \boldsymbol{H}(0), \boldsymbol{Q}(1), \boldsymbol{H}(1), ..., \boldsymbol{Q}(t), \boldsymbol{H}(t)\}.$$

A policy is called stationary if it is independent of time t, in which case the policy π can be represented by a single function. A policy is called Markov if it only depends on current topology state and queue state (i.e. it doesn't depends on the past history). So the policy can be written as a bi-variable (potentially randomized) function $\pi(\boldsymbol{H}, \boldsymbol{Q})$, which takes values among the set of control actions. In this work, we will focus on the class of Markov stationary policy.

A routing policy is called admissible if the control actions under any topology state and queue state are admissible. i.e.

$$\pi(\boldsymbol{H}, \boldsymbol{Q}) \in \mathcal{M}(\boldsymbol{H}) \quad \text{for all} \quad \boldsymbol{H} \in \mathcal{H}, \boldsymbol{Q} \in \mathcal{Q}$$

Let Π denote the set of all admissible routing policies.

Given a Markov policy and i.i.d. packet arrivals process and i.i.d. topology state process, the network queue state process $\{\boldsymbol{Q}(t)\}$ forms a Markov chain on the countable queue state space $\mathcal{Q}$. The stochastic evolution of queues affects the delay and throughput performance of the network. Hence, the problem of routing under Markov policies can be mapped to a Markov decision problem [34][35].

Though in this work, we restrict our attention to the i.i.d. topology state process and i.i.d. arrival process, it worths noting that the results can be generalized to the non-i.i.d case with general admissible inputs and Markov modulated topology state process. In [36] [37], the classical backpressure algorithm is discussed under the general admissible inputs and Markov modulated topology state process. Similar techniques can be used to prove the results in this work.

2.1.3 Other Assumptions

Inter-channel interference

We assume each network node transmits over an orthogonal channel, so that there is no inter-channel interference. This assumption allows for a clear presentation of the routing problem and illuminate the main concepts in their simplest forms. However, we emphasize that the generalization to the networks with inter-channel interference can follow as shown in [20]. In [20]. the price of this generalization is shown to be the centralization of the routing/scheduling globally across the network or a constant factor performance loss of the distributed variants.

Trapping nodes

For simplicity, we assume that there is no "trapping nodes". A "trapping node" is a node with no outgoing links or a node that is part of a group of notes with outgoing links that only connect to other nodes of the group [33].

Routing restrictions

In addition to the set of transmission links $\mathcal{L}$ for the whole network, it is also useful to impose routing restrictions for each commodity, and we define $\mathcal{L}_c \subseteq \mathcal{L}$ as the set of all links (a, b) that commodity c data is allowed to use. To ensure more predictable performance and to potentially reduce these delay problem, the link sets $\mathcal{L}_c$ can be designed in advance to ensure that all transmission move commodities closer to their destinations. Another special case is single-hop networks where only direct transmission between nodes is allowed. Such a restriction can be accomplished by setting $\mathcal{L}_c\{(a, b)\}$ for each commodity c whose traffic is originated at node a and destined to node b.

In this work, for easier of presentation, we assume $\mathcal{L}_c = \mathcal{L}$ and all nodes except for the destination nodes keep internal queues for the each commodity. However, it is worthy pointing out that additional routing restrictions can be easily applied. A handling of these restrictions for the classical backpressure algorithm case can be found in [33]. Imposing these routing restrictions to our problem is similar and trivial.

2.2 Network Stability and Network Capacity

2.2.1 Network Stability

First consider a single discrete time queue with arrival process $A(t)$ and departure process $\mu(t)$, where $A(t)$ is the amount of data that enter the queue during time slot t, and $\mu(t)$ is the transmission rate of the server during time slot t. We assume that the $A(t)$ arrivals occurs at the end of the time slot t, so that they cannot be transmitted during that time slot. Let $Q(t)$ be backlog in the queue at time slot t. Then the process $Q(t)$ evolves according to the following discrete time queueing law:

$$Q(t+1) = \max[U(t) - \mu(t), 0] + A(t) \tag{2.7}$$

The queue might be located within a larger network, in which case the arrival process $A(t)$ is composed of random exogenous arrivals as well as endogenous arrivals resulting from routing and transmission decisions from other nodes of the network. Similarly, the transmission rate $\mu(t)$ can be determined by a combination of random channel state variations and controlled network resource allocations, both of which can change from slot to slot. Therefore, it is important to develop a general definition of queueing stability that handles arbitrary $A(t)$ and $\mu(t)$ processes.

We begin with the general definition of stability.

Definition 2.1 *A queue with backlog process $Q(t)$ is* stable *if*

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t} Pr[Q(\tau) > M] \to 0 \tag{2.8}$$

as $M \to \infty$. A network of queues is stable *if all individual queues are stable*

The above stability definition is general in that it does not require a countably infinite state space, nor does it require ergodicity. A more detailed discussion of stability issues is given in [38][39][36][34].

2.2.2 Stability On a Countable State Space Markov Chain

The general definition of stability in Definition 2.1 is not very intuitive for our problem. By exploiting the special structure of countable state space Markov chain, we can obtain more structural results on the stability.

To begin with, let us first review some basic concept on a countable state space Markov chain:

Consider a Markov chain $\mathbf{\Phi} = \{\Phi_n\}$ on some countable state space X.

A state i is said to be transient if, given that we start in state i, there is a non-zero probability that we will never return to i. Formally, let the random variable τ_{ij} be the first time to state j from state i (the "hitting time"):

$$\tau_{ij} = \inf\{n \geq 1 : \Phi_n = j | \Phi_0 = i\}. \tag{2.9}$$

Then, state i is transient if and only if

$$\Pr(\tau_{ii} = \infty) > 0 \tag{2.10}$$

where τ_{ii} is also called return time since it represents the first time return time to state i. If a state i is not transient (it has finite hitting time with probability 1), then it is said to be recurrent.

The occupation time η_i is the number of visits by $\mathbf{\Phi}$ to i after time zero, and is given by

$$\eta_i := \sum_{n=1}^{\infty} \mathbb{I}\{\Phi_n = i\} \tag{2.11}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. It is easy to see that

$$\eta_i = \sum_{n=1}^{\infty} p_{ii}^{(n)} \tag{2.12}$$

where $p_{ii}^{(n)}$ the n step transit probability from state i to i. It can be shown that a state is recurrent if and only if

$$\eta_i = \sum_{n=1}^{\infty} p_{ii}^{(n)} = \infty \tag{2.13}$$

Although the hitting time is finite, it need not have a finite expectation. Let M_i be the expected return time, i.e. $M_i = E[\tau_{ii}]$. Then, state i is positive recurrent if M_i is finite; otherwise, state i is null recurrent.

A Markov chain $\mathbf{\Phi}$ is called recurrent if all its states are recurrent; A Markov chain $\mathbf{\Phi}$ is called transient if all its states are transient. It can be shown that if $\mathbf{\Phi}$ is irreducible, then either $\mathbf{\Phi}$ is transient or $\mathbf{\Phi}$ is recurrent.

A Markov chain is called positive recurrent if all its states are positive recurrent. A state i is said to be ergodic if it is aperiodic and positive recurrent. If all states in a Markov chain are ergodic, then the chain is said to be ergodic.

If the Markov chain is a time-homogeneous Markov chain, so that the process is described by a single, time-independent matrix p_{ij}, then the vector $\boldsymbol{\pi}$ is called a stationary distribution (or invariant probability distribution) if its entries π_j sum to 1 and it satisfies

$$\pi_j = \sum_{i \in S} \pi_i p_{ij} \tag{2.14}$$

An irreducible chain has a stationary distribution if and only if chain is positive recurrent (i.e. all of its states are positive recurrent) In that case, π is unique and is related to the expected return time:

$$\pi_j = \frac{1}{M_j} \tag{2.15}$$

Further, if the chain is ergodic (i.e. all states are positive recurrent and aperiodic), then for any i and j,

$$\lim_{n \to \infty} p_{ij}^{(n)} = \frac{1}{M_j} \tag{2.16}$$

Note that there is no assumption on the starting distribution; the chain converges to the stationary distribution regardless of where it begins. Such π is called the equilibrium distribution of the chain.

In the special case when queue backlog evolves according to an ergodic Markov chain with a countably infinite state space, then the notion of stability is equivalent to the existence of a steady state probability distribution for the chain [22]. Generally, irreducibility of the queue length process is not guaranteed. In that case, the state space is partitioned into transient and recurrent states. We consider the system to be stable if all recurrent states are positive recurrent and the queue length process hits the recurrent states with probability one. That is, $\mathbf{\Phi}$ does not remain in the set of transient states forever.

Formally, we say a state i is reachable by some state j if $Pr(\Phi_{n+m} = i|\Phi_n = j) > 0$ for some $m \geq 1$. The states i and j communicate if they are reachable by each other. A set of states R is closed if $Pr(\Phi_{n+1} = i|\Phi_n = j) = 0$ for all $j \in R$, $i \notin R$. The state space of the chain is partitioned in the sets T, R_1, R_2,..., where R_j, $j = 1, 2, ..$, are closed sets of communicating states and T contains all states which do not belong to any closed set of communicating states and therefore are transient. For any $i \in T$ consider the time

$$\tau_i = \inf\{n \geq 0 : \Phi_n \notin T|\Phi_0 = i\}. \tag{2.17}$$

at which the chain hits some of the sets R_j for the first time when it starts from state $i \in T$. Note that $\tau_i = \infty$ if $\Phi_n \in T$ for any $n > 0$. We can now define stability as follows

Definition 2.2 *The system is stable if for the Markov process* $\mathbf{\Phi}$*, we have*

$$Pr(\tau_i < \infty) = 1 \quad \forall i \in T \tag{2.18}$$

and all states $i \in \bigcup_j R_j$ *are positive recurrent.*

2.2.3 Lyapunov Stability

The key mathematical tool we are going to use to prove stability of queueing networks and develop stabilizing control algorithms is the technique of Lyapunov drift. It utilizes some recurrence criterion based on the existence of a so called Lyapunov function with certain property. The idea is to define a non-negative function, called a Lyapunov function, as a scalar measure of the aggregate congestion of all queues in the network. Network control decisions are then evaluated in terms of how they affect the change in the Lyapunov function from one slot to the next. This type of criterion first appeared in the literature in paper of F.G Foster [40]. Some literature also called it Foster's criterion.

Let $\mathbf{\Phi} = \{\Phi_n\}_{n=0}^{\infty}$ be a Markov chain on a countable state space X with transmission probability $\{p_{ij}\}_{i,j \in X}$.

Let $L : S \mapsto [0, \infty)$ be a test function, which we referred as Lyapunov function. Then the Lyapunov drift $\Delta L(x)$ is defined as

$$\Delta L(x) := \sum_{y \in X} L(y) p_{xy} - L(x), \quad x \in X \tag{2.19}$$

Theorem 2.1 *(Lyapunov drift criteria for recurrence [34]) The irreducible Markov chain $\mathbf{\Phi}$ is recurrent if there exists a finite set $C \subset X$ and a Lyapunov function L such that*

$$\Delta L(x) \leq 0, \quad x \in C^c \tag{2.20}$$

and

$$\{x : L(x) \leq n\} \tag{2.21}$$

is finite for all n.

It is well known that positive recurrence provides a much stronger kind of stochastic stability than recurrence, since it is implies the existence of steady state distribution of a Markov chain. The following theorem provides a Lyapunov drift criterion for positive recurrence, which is a stronger version than Theorem 2.1.

Theorem 2.2 *(Lyapunov drift criteria for positive recurrence [34]) The irreducible Markov chain* $\mathbf{\Phi}$ *is positive recurrent if there exists a finite set* $C \subset X$ *and a Lyapunov function* L *such that*

$$\Delta L(x) \leq -1, \quad x \in C^c \tag{2.22}$$

and

$$\Delta L(x) \leq M, \quad x \in C \tag{2.23}$$

Condition (2.22) and (2.23) can be compactly written as

$$\Delta L(x) \leq -1 + b\mathbb{I}\{x \in C\} \tag{2.24}$$

for some constant $b < \infty$.

Note that Theorem 2.1 and Theorem 2.2 are for irreducible Markov chain only. The following theorem is an extension of Theorem 2.2 for a general Markov chain on a countable state space.

Theorem 2.3 *(Lyapunov drift criteria of stability of a general Markov chain [22]) Consider a Markov chain* $\mathbf{\Phi}$*. If there exists a finite set* $C \subset X$ *and a Lyapunov function* L *such that*

$$\Delta L(x) \leq -1, \quad x \in C^c \tag{2.25}$$

and

$$\Delta L(x) \leq M, \quad x \in C \tag{2.26}$$

then for the time τ_i as defined in (2.18), we have

$$Pr(\tau_i < \infty) = 1 \quad \forall i \in T \tag{2.27}$$

and all states $i \in \bigcup_j R_j$ are positive recurrent. So by Definition 2.2, $\mathbf{\Phi}$ is stable.

Now we are going to utilize Theorem 2.3 to get some useful criterion on the queue stability.

Let $L(\boldsymbol{Q})$ denote the Lyapunov function of queue state. From (2.19), it can be seen that the Lyapunov drift can be calculated by

$$\Delta L(\boldsymbol{Q}(t)) := \mathbb{E}\{L(\boldsymbol{Q}(t+1)) - L(\boldsymbol{Q}(t)) | \boldsymbol{Q}(t)\} \tag{2.28}$$

which is the expected change in the Lyapunov function from one slot to the next.

Assume that $\boldsymbol{Q}(t)$ evolves according to some probabilistic law, and that the initial conditions are such that $\mathbb{E}\{L(\boldsymbol{Q}(0))\} < \infty$. Then we have the following criterion on queue stability, which will serve as the basis for our main results.

Lemma 2.1 *(Lyapunov Stability) If there exist constants $B > 0$, $\epsilon > 0$, such that for all timeslot t we have:*

$$\Delta L(\boldsymbol{Q}(t)) \leq B - \epsilon \sum_c \sum_a Q_a^c(t) \tag{2.29}$$

then the network is stable.

Proof:

Define set C as

$$C := \{\boldsymbol{Q} \in \mathcal{Q} : \sum_c \sum_i Q_a^c(t) \leq \frac{B+1}{\epsilon}, \} \tag{2.30}$$

It is easy to see that C is a finite set. This is because we can rewrite (2.30) as $C = \mathcal{Q} \cap \hat{C}$ where $\hat{C} := \{\boldsymbol{Q} \in \mathbb{R}_+^{N \times |\mathcal{K}|} : \sum_c \sum_i Q_a^c(t) \leq \frac{B+1}{\epsilon}\}$. Since $\hat{C}$ is a compact set and $\mathcal{Q}$ is discrete, $\mathcal{Q} \cap \hat{C}$ must be a finite set. Note that for $\boldsymbol{Q}(t) \in \mathcal{Q} \setminus C$, we have

$$\Delta L(\boldsymbol{Q}(t)) \leq B - \epsilon \sum_c \sum_a Q_a^c(t) \leq -1, \tag{2.31}$$

and for $\boldsymbol{Q}(t) \in C$, we have

$$\Delta L(\boldsymbol{Q}(t)) \leq B \tag{2.32}$$

By Lemma 2.3, the Markov process $\{\boldsymbol{Q}(t)\}$ is positive recurrent, i.e. the network is stable.

□

Remark: It turns out that the criteria in (2.29) not only implies stability, but also strong stability. The proof of strong stability can be found in [33] (Lemma 4.1).

2.2.4 Network Capacity

The capacity region Λ of a network is the closure of the set of all arrival rate matrices $\lambda_i^c(t)$ that can be stably supported by the network, considering all possible strategies for choosing the control variables to affect routing, scheduling, and resource allocation (including strategies that have perfect knowledge of future events).

The capacity region Λ_π of a given routing policy π of is the closure of the set of all arrival rate matrices $\lambda_i^c(t)$ that can be stably supported by the network under the policy

The capacity region of the network should be distinguished from the capacity region of a specific policy. Clearly $\Lambda_\pi \subseteq \Lambda$, i.e the network capacity is the superset of the capacity region of any policy. The capacity region of the network is the union of the individual policy capacity regions, taken over all possible policies. One way to characterize the performance of a policy is by its rate capacity region itself. The larger the capacity region the better the performance will be since the network will be stable for a wider range of traffic loads and therefore more robust to traffic fluctuations. Such a performance criterion makes even more sense in the context of wireless ad-hoc networks where both the traffic load as well as the network capacity may vary unpredictably. A policy A is termed better than B with respect to their capacity regions, if the capacity region of A is a superset of the capacity region of B.

A policy is said to be *throughput optimal* if its capacity region is the superset of the capacity region of any other policies. Clearly, a throughput optimal policy, if exist, stabling the network for all arrival rate that belongs to the interior of the capacity region. i.e. the capacity region of a throughput optimal policy coincides with the network capacity region.

Generally, it is not clear if a throughput optimal policy exists. However, it is well known that for any arrival rate within the capacity region, there exists a randomized policy that only depends on the topology of the network (independent of the queue backlogs) and stabilizes the network. Formally, we have the following theorem:

Theorem 2.4 *[Corollary 3.9 in [33]]. If the topology state $\boldsymbol{H}(t)$ is i.i.d. from slot to slot, then a rate matrix (λ_i^c) is within the capacity region Λ if and only if there exists a stationary randomized control algorithm that makes decisions $\mu_{ab}^c(t)$ based only on the current topology state $\boldsymbol{H}(t)$, and that yields for all i, c and all time t:*

$$\mathbb{E}\left\{\sum_b \mu_{ib}^c(t) - \sum_a \mu_{ai}^c(t)\right\} = \lambda_i^c, \quad \text{for all} \quad i, c, i \neq dest(c), \tag{2.33}$$

where $dest(c)$ denote the destination of commodity c, and the expectation is taken with respect to the random topology state $\boldsymbol{H}(t)$ and the (potentially) random control action based on this state.

Randomized policies are rarely used in practice since 1) finding such a randomized policy requires global network knowledge and extensive calculations and 2) ignoring queue state information generally results in poor delay performance.

The significance of Theorem 2.4 is that it provides a reference policy that is independent of backlog state. In later chapters, we will use the randomized policy to compare with other policies in order to prove their throughput optimality.

2.3 Acknowledgement

Chapter 2 is in part a reprint of the material in paper: M. Naghshvar, H. Zhuang, and T. Javidi, "A General Class of Throughput Optimal Routing Policies in Multi-hop Wireless Networks,", *submitted for IEEE Transactions on Information Theory.* The dissertation author was the primary investigator and author of the paper.

Chapter 3

Single Commodity Case: K Policy and Its Throughput Optimality

In this chapter, we will start with a simpler case when the set of commodity $\mathcal{K}$ is a singleton. In this case, multiple arrivals from different nodes are destined to a single destination (The destination could be a single node or a set of nodes).

For notation convenience, in this chapter we will assume there are N nodes in the network except the destination nodes and label them by $1, 2, 3..., N$. (So the total number of nodes in the network is N plus the number of destination nodes. Also because there is only one commodity, we will drop the commodity superscript notation in this chapter as this won't cause any confusion.

3.1 Priority Routing

For the single commodity case, we will focus on a special class of routing policies called priority routing policies. Priority routing policies forms a subset of stationary Markov policies. It turns out that when there is only a single commodity, by restricting to a smaller class of routing polices among the whole set of stationary

Markov policies, we are still able to achieve the capacity of the network. We will be able to prove this later in this chapter.

To define the general priority routing policy, we need some new definitions.

3.1.1 Priority Class and Priority List

A priority class is an non-empty subset of $\mathcal{N}$. The set of destination nodes, denoted by C_0, is a special priority class and is assigned with the lowest priority among all priority classes. In case there is a single destination, C_0 is a singleton. But generally C_0 can include more than one node. A priority list $P = (C_1, C_2, ..., C_{|P|})$ is an ordered list, where $|P|$ represents the number of priority classes of priority list P and $C_1, C_2, ..., C_{|P|}$ are priority classes that make up a complete partition of $\mathcal{N} \setminus C_0$ (all network nodes except for the destination node(s)). i.e. $C_i \cap C_j = \varnothing$, $i \neq j$ and $\bigcup_{i=1}^{|P|} C_i = \mathcal{N} \setminus C_0$.

Given a priority list $P = (C_1, C_2, ..., C_{|P|})$, if $a \in C_i$, $b \in C_j$ and $i < j$, then we say C_i is a lower priority class than C_j (C_j is a higher priority class than C_i) and node a has lower priority than node b under priority list P (node b has higher priority than node a under priority list P). We concisely write $a \prec^P b$. Similarly, if $a \in C_i$, $b \in C_j$ and $i \leq j$, we write $a \preceq^P b$.

Note that set C_i is unordered, while P is ordered. Let $\mathbb{P}_{all}$ denote the set of all possible priority lists.

The introduction of notion of priority classes allows us to group several nodes in the same priority class as a "super node" to study its behavior as a whole. For convenience, we introduce some more notations below:

Let $Q_{C_i}(t) := \sum_{a \in C_i} Q_a(t)$.

Let $A_{C_i}(t)$ denote the amount of data arriving to priority class C_i during time slot t, i.e.

$$A_{C_i}(t) := \sum_{a \in C_i} A_a(t) \tag{3.1}$$

And also define the arrival rate to priority class C_i as

$$\lambda_{C_i} := \sum_{a \in C_i} \lambda_a \tag{3.2}$$

We further define

$$\mu_{C_i,in}(t) := \sum_{a \notin C_i} \sum_{b \in C_i} \mu_{a,b}(t) \tag{3.3}$$

$$\mu_{C_i,out}(t) := \sum_{a \in C_i} \sum_{b \notin C_i} \mu_{a,b}(t) \tag{3.4}$$

$\mu_{C_i,in}(t)$ is the amount of data transmitted into priority class C_i from other nodes. $\mu_{C_i,out}(t)$ is the amount of data transmitted from priority class C_i into other nodes.

3.1.2 Priority Policy

Given a priority list P, an admissible control action $\{\mu^c_{ab}(t)\}$ is said to be a control action based on priority list P at time slot t if $\mu_{ab}(t) = 1$ implies $b \preceq^P b'$ for any $b' \in \mathcal{S}_a(t)$. That is, node a forwards a packet to b only if b has the lowest priority among the potential forwarder set of node a. Let $\mathcal{S}^*_a(t) \subseteq \mathcal{S}_a(t)$ denote the set of nodes with the lowest priority among $\mathcal{S}_a(t)$. There are 3 cases:

Case 1: $\mathcal{S}^*_a(t) = \{a\}$: i.e. node a has lower priority than any other nodes in $\{\mu^c_{ab}(t)\}$. In this case, node a doesn't forward any packet, i.e. $\mu_{aa}(t) = 1$.

Case 2: $\{a\} \subset \mathcal{S}^*_a(t)$. Then either node a holds the packet or forwards to one of the nodes in $\mathcal{S}^*_a(t) \setminus a$.

Case 3: $a \notin \mathcal{S}^*_a(t)$. Then node a forwards a packet to one of the nodes in $\mathcal{S}^*_a(t)$.

As a special priority class, the nodes in C_0 always have the lowest priority among all other nodes since other nodes should forward directly to a destination node whenever possible.

It can be seen from above that when $S_a^*(t)$ is not a singleton, the routing control action based on a given priority list is not unique. Let $\mathcal{M}(\boldsymbol{H}, P)$ denote the set of admissible routing control actions based on priority list P under topology state $\boldsymbol{H}$. Obviously, we have $\mathcal{M}(\boldsymbol{H}, P) \subseteq \mathcal{M}(\boldsymbol{H})$

A Markov stationary routing policy π is called a priority routing policy if there exists a so called priority function $\mathcal{P} : \mathbb{R}_+^N \mapsto \mathbb{P}_{all}$ such that

$$\pi(\boldsymbol{H}, \boldsymbol{Q}) \in \mathcal{M}(\boldsymbol{H}, \mathcal{P}(\boldsymbol{Q})) \tag{3.5}$$

for all $\boldsymbol{H}$, $\boldsymbol{Q}$

We call the priority function $\mathcal{P}$ in (3.5) as a *priority policy*. It should be noted that a priority policy π that takes values in $\mathbb{P}_{all}$ should not be confused with a priority routing policy π that takes values in the set of routing control actions. Since the right hand side of (3.5) is generally a set of routing control actions, a priority policy $\mathcal{P}$ generally defines a class of priority routing policy π. In the rest of this chapter, we will focus on priority policy $\mathcal{P}$ instead of priority routing policy π. Let us start with more definitions about $\mathcal{P}$.

A priority policy $\mathcal{P}$ is called a cone policy if

$$\mathcal{P}(\boldsymbol{Q}) = \mathcal{P}(c\boldsymbol{Q}) \tag{3.6}$$

for all $\boldsymbol{Q} \in \mathbb{R}_+^N$ and $c > 0$.

Given two priority lists P and P', P' is a refinement of P if node $a \prec^P b$ implies $a \prec^{P'} b$ for any $a, b \in \mathcal{N} \setminus C_0$. Note that by definition, a priority list is a refinement of itself.

Given two priority policies $\mathcal{P}$ and $\mathcal{P}'$, if $\mathcal{P}'(\boldsymbol{Q})$ is a refinement of $\mathcal{P}(\boldsymbol{Q})$ for any $\boldsymbol{Q} \in \mathbb{R}_+^N$, then we call $\mathcal{P}'$ a refined policy of $\mathcal{P}$.

Clearly, by definition, if P' is a refinement of P, then the routing control action based on P' is also a routing control action based on P. (The converse is generally not true). Formally, we have

Proposition 3.1 *If P' is a refinement of P, then $\mathcal{M}(\boldsymbol{H}, P') \subseteq \mathcal{M}(\boldsymbol{H}, P)$*

By Proposition 3.1, if $\mathcal{P}'$ is a refined policy of $\mathcal{P}$, then

$$\mathcal{M}(\boldsymbol{H}, \mathcal{P}'(\boldsymbol{Q})) \subseteq \mathcal{M}(\boldsymbol{H}, \mathcal{P}(\boldsymbol{Q})) \tag{3.7}$$

for all $\boldsymbol{H}$ and $\boldsymbol{Q}$. Then by the definition in (3.5), the set of priority routing policies π defined by $\mathcal{P}$ is superset of that defined by $\mathcal{P}'$. Hence we have the following simple but important result.

Proposition 3.2 *If a priority policy is throughput optimal, all its refined polices are throughput optimal.*

3.1.3 Critical Priority List

So far we have restricted ourself to the class of priority policies that take values among the set all possible priority lists $\mathbb{P}_{all}$. It turns out when the set of link $\mathcal{L}$ doesn't include all possible links, some priority lists are not required to achieve the throughput optimality, thus these priority lists are *non-critical*. In practice, by excluding these non-critical priority listings, the delay performance might be further improved.

Before formally classifying the priority lists into critical ones and non-critical ones, we need to first classify the nodes in a priority list. Given a priority list P, under a link set $\mathcal{L}$, each node a can be in one of the three following status:

1. Node a is *non-isolated* under P if there exists a node b such that $b \prec^P a$ and $(a, b) \in \mathcal{L}$. That is a is able to transmit directly to a node in a lower priority class than that of a.

2. Node a is *semi-isolated* under P if a is not non-isolated but there exists a non-isolated node b in the same priority class such that $(a, b) \in \mathcal{L}$ or

there exist intermediate nodes $a_1, a_2, ..., a_k$ from same priority class such that $(a, a_1), (a_1, a_2), ..., (a_k, b) \in \mathcal{L}$. Since we know that non-isolated nodes can transmit directly to a node in a lower priority class, semi-isolated nodes, in other words, are those who can indirectly transmit to a node in a lower priority class via a "*path*" within the priority class.

3. Node a is *isolated* under P if a is neither non-isolated nor semi-isolated.

Figure 3.1 shows an example of non-isolated nodes, semi-isolated nodes, and isolated nodes.

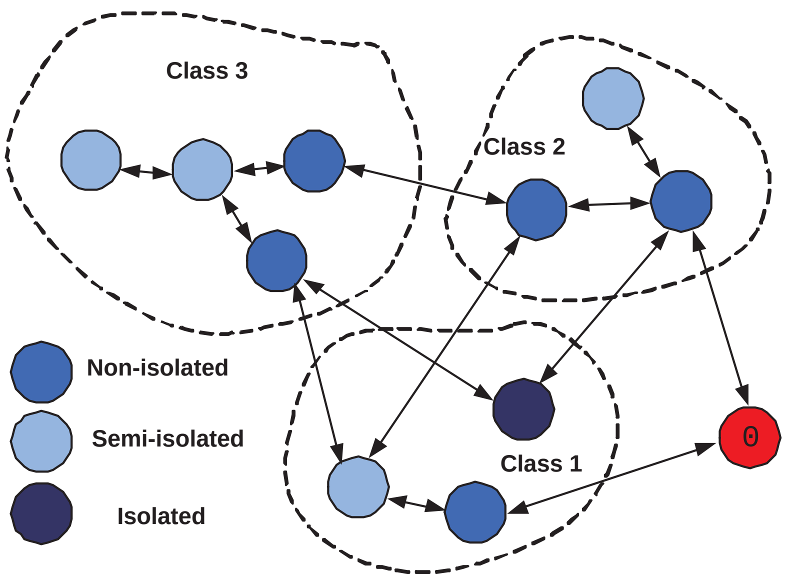

Figure 3.1: Example of classification of nodes in a priority list

A priority list is *critical* if each node is either non-isolated or semi-isolated (i.e. no isolated nodes). A priority list is *non-critical* if there exists an isolated node.

By this definition, if a priority list is critical, then for each node a (either semi-isolated or non-isolated) there exist intermediate nodes $a_1, a_2, ..., a_k \in \mathcal{N} \setminus C_0$ with priority order $a \succeq^P a_1 \succeq^P a_2 \succeq^P ... \succeq^P a_k$ and $b \in C_0$ such that

$(a, a_1), (a_1, a_2), ..., (a_k, b) \in \mathcal{L}$. That is, there exists a "*path*" to destination via nodes with priorities *non-increasing* in order.

Given a link set $\mathcal{L}$, denote the set of critical priority lists by $\mathbb{P}_{\mathcal{L}}$. Clearly $\mathbb{P}_{\mathcal{L}}$ is a subset of $\mathbb{P}_{all}$ since the latter contains all possible priority lists.

Proposition 3.3 *If P is a critical list, all its confinements are critical.*

Proof: By the definition of non-isolated node, if a node is non-isolated under P, it is either non-isolated or semi-isolated under the confinement of P. By the definition of semi-isolated node, if a node is semi-isolated under P, it is still semi-isolated under the confinement of P. So we conclude that if a priority list contains only non-isolated and semi-isolated nodes, all its confinements contain only non-isolated and semi-isolated nodes. By definition of critical list, all its confinements are critical.

□

Consider two link sets $\mathcal{L}'$ and $\mathcal{L}$. Assume $\mathcal{L}' \subset \mathcal{L}$, that is $\mathcal{L}'$ has less connectivity than $\mathcal{L}$ has. We refer to $\mathcal{L}'$ as a sub-network of $\mathcal{L}$, and $\mathcal{L}$ as a super-network of $\mathcal{L}'$. The following proposition states the node classification relationship between a sub-network and a super-network. Note that the reverse of the propostion is generally not true.

Proposition 3.4 *If $\mathcal{L}' \subset \mathcal{L}$, then for any priority list P, a non-isolated node under $\mathcal{L}'$ is also non-isolated under $\mathcal{L}$; a isolated node under $\mathcal{L}$ is also isolated under $\mathcal{L}'$. And any critical list under $\mathcal{L}'$ is critical under $\mathcal{L}$, i.e. $\mathbb{P}_{\mathcal{L}'} \subseteq \mathbb{P}_{\mathcal{L}}$.*

Proof: The proof is straightforward by definition.

□

The following figure illustrates the relationship between a sub-network and a super-network.

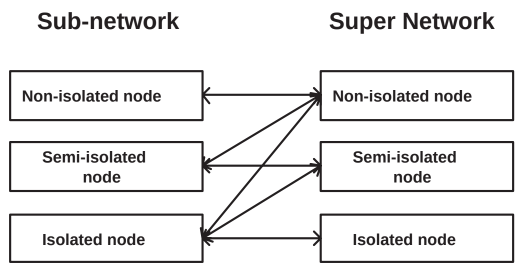

Figure 3.2: Sub-network and super-network

3.2 K Policy and Its Stability

In this section, we define a class of stationary priority policies called K policy and we show that K policy is throughput optimal. To formally define K policy, some more definitions need to be introduced first.

The following weight function will play a key role in defining K policy:

$$W_K(n,m) := \frac{1}{K^n(K^m - 1)} \tag{3.8}$$

Note that if we approximate $K^m - 1$ by K^m, the function can be approximated by

$$W_K(n,m) \approx K^{-(n+m)} \tag{3.9}$$

(3.9) is a good approximation when K is large. It will be useful to provide some intuitions later.

The following simple property of the weight function is very important and will be used later.

Proposition 3.5

$$\frac{1}{W_K(n, m_1 + m_2)} = \frac{1}{W_K(n, m_1)} + \frac{1}{W_K(n + m_1, m_2)} \tag{3.10}$$

Proof: By definition of $W_K(n,m)$ in (3.8)

□

Given two priority lists, $P = (C_1, C_2, ..., C_{|P|})$ and $P' = \{C'_1, C'_2, ..., C'_{|P'|}\}$, we say P' is a one-step refinement of P (and P is a one-step confinement of P') with regard to priority class $C_i(1 \leq i \leq |P|)$ if $|P| = |P'| + 1$ and

$$\begin{cases} C_k = C'_k & \text{if } 1 \leq k \leq i-1 \\ C_i = C'_i \cup C'_{i+1} & \\ C_k = C'_{k+1} & \text{if } i+1 \leq k \leq |P| \end{cases} \tag{3.11}$$

We also refer to P and P' as an adjacent list pair. Clearly, P' is a refinement of P. i.e. one-step refinement is a special case of refinement.

If P' is a one-step refinement of P with regard to priority class C_i, then the hyperplane that separates the adjacent list pair P and P' is defined as

$$H_K(P;P') := \Big\{ \boldsymbol{Q} \quad | \quad W_K(|C'_{<i}|, |C'_i|) Q_{C'_i} = W_K(|C'_{<i+1}|, |C'_{i+1}|) Q_{C'_{i+1}} \Big\} \tag{3.12}$$

where $|\cdot|$ denote the cardinality of a set, and $C'_{<i}$ denote the set of nodes that has lower priority than nodes in the i-th priority class, i.e.

$$C'_{<i} := \bigcup_{j=1}^{i-1} C'_j \tag{3.13}$$

(3.12) separates the whole space into two half spaces. The half space of the one-step refinement list P' is given by

$$H^+_K(P;P') := \Big\{ \boldsymbol{Q} \quad | \quad W_K(|C'_{<i}|, |C'_i|) Q_{C'_i} \leq W_K(|C'_{<i+1}|, |C'_{i+1}|) Q_{C'_{i+1}} \Big\} \tag{3.14}$$

The half space of the one-step confinement list P is given by

$$H^-_K(P;P') := \Big\{ \boldsymbol{Q} \quad | \quad W_K(|C'_{<i}|, |C'_i|) Q_{C'_i} \geq W_K(|C'_{<i+1}|, |C'_{i+1}|) Q_{C'_{i+1}} \Big\} \tag{3.15}$$

By using the approximation in (3.9), the linear equation in (3.12) can be approximated by

$$K^{|C'_{>i}|} Q_{C'_i} = K^{|C'_{>i+1}|} Q_{C'_{i+1}} \tag{3.16}$$

or equivalently

$$K^{|C'_{i+1}|}Q_{C'_i} = Q_{C'_{i+1}} \tag{3.17}$$

As we can see, the equation compares the total queue length of two consecutive priority classes by a factor approximated by $K^{|C'_{i+1}|}$.

Recall that $\mathbb{P}_{\mathcal{L}}$ is the set of critical lists under link set $\mathcal{L}$. In what follows, we will construct K policy based on $\mathbb{P}_{\mathcal{L}}$. We are going to define priority cone for each $P \in \mathbb{P}_{\mathcal{L}}$ and these cones make up a complete partition of queue space $\mathbb{R}^N_+$.

Define $\mathbb{P}^+_{\mathcal{L}}(P)$ as the set of one-step critical refinements of P, i.e. $\mathbb{P}^+_{\mathcal{L}}(P) := \{P' \in \mathbb{P}_{\mathcal{L}} : P'$ is one-step refinement of $P\}$. And likewise, define $\mathbb{P}^-_{\mathcal{L}}(P)$ as the set of one-step confinements of P i.e. $\mathbb{P}^-_{\mathcal{L}}(P) := \{P' : P'$ is one-step confinement of $P\}$. By Proposition 3.3, $\mathbb{P}^-_{\mathcal{L}}(P) \subseteq \mathbb{P}_{\mathcal{L}}$.

Given K, for each $P \in \mathbb{P}_{\mathcal{L}}$, define $D^{\mathcal{L}}_K(P) \subset \mathbb{R}^N_+$ as follows:

$$D^{\mathcal{L}}_K(P) := \mathbb{R}^N_+ \cap \left(\bigcap_{P' \in \mathbb{P}^-_{\mathcal{L}}(P)} H^+_K(P'; P) \right) \cap \left(\bigcap_{P' \in \mathbb{P}^+_{\mathcal{L}}(P)} H^-_K(P; P') \right) \tag{3.18}$$

where $H^+_K(P'; P)$ and $H^-_K(P; P')$ are defined in (3.14) and (3.15), respectively.

Each $D^{\mathcal{L}}_K(P)$ is a cone since it is an intersection of half spaces. (That is, if $\boldsymbol{Q} \in D^{\mathcal{L}}_K(P)$ then $c\boldsymbol{Q} \in D^{\mathcal{L}}_K(P)$ for any positive real number c). Therefore we refer to $D^{\mathcal{L}}_K(P)$ as a priority cone of priority list P parameterized by K.

The following lemma gives two useful properties of queue length of a non-isolated node from a priority cone:

Lemma 3.1 *Given a critical priority list* $P = (C_1, ..., C_{|P|}) \in \mathbb{P}_{\mathcal{L}}$*, if* $\boldsymbol{Q} \in D^{\mathcal{L}}_K(P)$ *and* $a \in C_i$ *is a non-isolated node, then*

1)

$$Q_a \geq \frac{K-1}{K^{|C_i|}-1} Q_{C_i} \tag{3.19}$$

2)

$$Q_a > (K-1) Q_{C_{<i}} \tag{3.20}$$

where $Q_{C_{<i}} := \sum_{s=1}^{i-1} Q_{C_s}$.

Proof: See Appendix A.1.

$\square$

The first property in Lemma 3.1 states that the queue length of non-isolated node cannot be "too small" compared to the total queue length of within its priority class. The second property says that the queue length of non-isolated node cannot be "too small" compared to the total queue length of the nodes in lower priority classes. These two properties are very important for establishing our results.

To get some intuition about priority cone and its properties Lemma 3.1, notice that second terms on the right hand side of (3.18) can be interpreted as the inter-class constraints. It basically, compares any two consecutive priority classes C_i and C_{i+1} and impose the following constraint:

$$W_K(|C_{<i}|, |C_i|)Q_{C_i} \leq W_K(|C_{<i+1}|, |C_{i+1}|)Q_{C_{i+1}} \tag{3.21}$$

Using the approximation in (3.9), (3.21) is approximated by

$$K^{|C_{i+1}|}Q_{C_i} \leq Q_{C_{i+1}} \tag{3.22}$$

The intuition here is that in order to separate priority class C_{i+1} from C_i. The total queue length of C_{i+1} has to be $K^{|C_{i+1}|}$ times larger than that of C_i. The ratio increases as K increases.

The last term on the right hand side of (3.18) can be interpreted as the intra-class constraints. It basically require the nodes within the same priority class not to have a queue length significantly different from the others. To get some intuition, say we want to find the intra-class constraints for priority class C_i of P. And for simplicity, let us assume that all confinements of P with regards to

C_i are critical. Then the last term on the right hand side of (3.18) will impose the following constraint for priority class C_i of P.

$$W_K(|C_{<i}| + |U|, |V|)Q_V \leq W_K(|C_{<i}|, |U|)Q_U, \quad \text{for all} \quad U \cup V = C_i, \quad U, V \neq \varnothing \tag{3.23}$$

Using the approximation in (3.9), (3.23) is approximated by

$$Q_V \leq K^{|V|}Q_U, \quad \text{for all} \quad U \cup V = C_i, \quad U, V \neq \varnothing \tag{3.24}$$

Due to the symmetry of U and V, (3.24) can also be written as

$$K^{-|V|}Q_U \leq Q_V \leq K^{|V|}Q_U, \quad \text{for all} \quad U \cup V = C_i, \quad U, V \neq \varnothing \tag{3.25}$$

The intuition here is that any subset of nodes within priority class C_i should have the total queue length $K^{|V|}$ times less than the total queue length of the rest nodes in C_i but $K^{-|V|}$ times greater than the total queue length of the rest nodes in C_i. The ratio increases as K increases.

To summarize, roughly speaking, $D_K^{\mathcal{L}}(P)$ has inter-class constraints

$$K^{|C_{i+1}|}Q_{C_i} \leq Q_{C_{i+1}} \tag{3.26}$$

for any two consecutive priority classes and intra-class constraints with in each C_i in the following form

$$K^{-|V|}Q_U \leq Q_V \leq K^{|V|}Q_U, \quad \text{for all} \quad U \cup V = C_i, \quad U, V \neq \varnothing \tag{3.27}$$

Note that both (3.26) and (3.27) are approximations.

We haven defined $\{D_K^{\mathcal{L}}(P), P \in \mathbb{P}_{\mathcal{L}}\}$ for a given link set $\mathcal{L}$. To ensure that K policy is well defined, we need to answer two questions: First, are these $D_K^{\mathcal{L}}(P)$ are non-overlapped? Secondly, would these $D_K^{\mathcal{L}}(P)$ be able to cover the whole queue space? It turns out that the answers to both questions are YES, which is formally stated in the following proposition:

Proposition 3.6 *Given a link set* $\mathcal{L}$ *of a network and* $K \geq 2$*,* $\{D_K^{\mathcal{L}}(P), P \in \mathbb{P}_{\mathcal{L}}\}$ *are a complete partition of queue space* $\mathbb{R}_+^N$*, i.e.* $\bigcup_{P \in \mathbb{P}_{\mathcal{L}}} D_K^{\mathcal{L}}(P) = \mathbb{R}_+^N$ *and* $int(D_K^{\mathcal{L}}(P_1)) \cap int(D_K^{\mathcal{L}}(P_2)) = \varnothing$ *for* $P_1, P_2 \in \mathbb{P}_{\mathcal{L}}$ *and* $P_1 \neq P_2$*, where* $int(\cdot)$ *stands for interior.*

Proof: See Appendix A.2.

□

Remark: One might think of using the approximation relationship in (3.26) and (3.27) to characterize priority cones for its simplicity in form. However this resulted priority cones don't form a complete partition. The weight function $W_K(|C_{<i}|, |C_i|)$ is introduced to technically construct the priority cones to form a complete partition.

With Proposition 3.6 established, we are now ready to define K policy.

Definition 3.1 *(K policy) Given a link set* $\mathcal{L}$ *and* $K \geq 2$*, K policy* $\mathcal{P}_K$ *based on critical list set* $\mathbb{P}_{\mathcal{L}}$ *is a priority policy defined as follows:*

$$\mathcal{P}_K(\boldsymbol{Q}) = P \quad if \quad \boldsymbol{Q} \in D_K^{\mathcal{L}}(P) \tag{3.28}$$

In other words, the policy $\mathcal{P}_K$ routes the packet according to priority list P when $\boldsymbol{Q}(t) \in D_K^{\mathcal{L}}(P)$ at time slot t. By Proposition 3.6, K policy is a well defined priority policy as long as $K \geq 2$. Moreover, it is clear that K policy is a cone policy, since all $D_K^{\mathcal{L}}(P)$s are cones. By definition, we have $\boldsymbol{Q} \in D_K^{\mathcal{L}}(\mathcal{P}_K(\boldsymbol{Q}))$ for any $\boldsymbol{Q} \in \mathbb{R}_+^N$

The important property of K policy is its throughput optimality. The main result is stated as follows. The proof will be given in the next section.

Theorem 3.1 *Given a network with link set* $\mathcal{L}'$*, if the arrival rate* $\boldsymbol{\lambda}$ *is strictly within its capacity region* Λ*, then any K policy (*$K \geq 2$*) based on* $\mathbb{P}_{\mathcal{L}}$ *with* $\mathcal{L}' \subseteq \mathcal{L}$

stabilizes the network. i.e. K policy based on $\mathbb{P}_{\mathcal{L}}$ *is throughput optimal for any network with link set* $\mathcal{L}'$ *such that* $\mathcal{L}' \subseteq \mathcal{L}$.

Remark: $\mathcal{L}'$ is a link set with the same or less connectivity than $\mathcal{L}$. The theorem indicates certain robustness of K-policies. In practice, some links in a network might go down accidentally, resulting a sub-network with less connectivity than the original network. The theorem shows that the K policy designed based on the original network will still stabilize the network as long as the arrival rate is still within the capacity region of the sub-network.

3.3 Proof of Stability of K Policy

Similar to the proof of stability of backpressure algorithm, the central tool we are going to use to proof stability is Lemma 2.1. The main problem is then to find and construct a Lyapunov function, such that the one-step drift of the Lyapunov function is negative as long as the total backlog is sufficiently large. It is easy to see that the quadric form function used in the proof of stability of backpressure algorithm would not work. In what follows, we will construct a special Lyapunov function, which is weighted quadric in the cone of each critical priority list. The weights are different from cone to cone, but the function keep its continuality and smoothness on the boundary of the cones. Hence the entire function is piecewise quadric. By utilizing this special piecewise quadric function, we will be able to obtain negative drift for large total backlog.

3.3.1 Construction of the Lyapunov Function

Given a priority list $P = (C_1, C_2, ..., C_{|P|})$, the Lyapunov function under list P and parameter K is defined as

$$L_K(\boldsymbol{Q}; P) := \sum_{i=1}^{|P|} W_K(|C_{<i}|, |C_i|) Q_{C_i}^2 \tag{3.29}$$

where $W_K(n, m)$ is defined in (3.8). Note here n is substituted by the number of nodes that is in the lower priority classes than C_i and m is substituted by the number of nodes in priority class C_i. To get some intuitions, we use the approximation of $W_K(n, m)$ in (3.9) to rewrite (3.29) as

$$L_K(\boldsymbol{Q}; P) \approx \sum_{i=1}^{|P|} K^{|C_{>i}|} Q_{C_i}^2 \tag{3.30}$$

where $|C_{>i}|$ is the number of nodes that has higher priority than nodes in C_i. As we can see from (3.30), the function imposes lower cost on the higher priority class.

Proposition 3.7 *For any adjacent list pair P and P', $L_K(\boldsymbol{Q}; P)$ and $L_K(\boldsymbol{Q}; P')$ are equal and have equal gradient on separation hyperplane $H_K(P; P')$ defined in* (3.12)

Proof: Suppose we are given an adjacent list pair, $P = \{C_1, C_2, ..., C_{|P|}\}$ and $P' = (C'_1, C'_2, ..., C'_{|P'|})$. Without loss of generality, assume P' is a one-step refinement of P with regard to priority class $C_i (1 \leq i \leq |P|)$, then $|P'| = |P| + 1$ and

$$\begin{cases} C_k = C'_k & \text{if } 1 \leq k \leq i-1 \\ C_i = C'_i \cup C'_{i+1} & \\ C_k = C'_{k+1} & \text{if } i+1 \leq k \leq |P| \end{cases} \tag{3.31}$$

On the hyperplane that separates the adjacent list pair P and P', we have by definition

$$W_K(|C'_{<i}|, |C'_i|) Q_{C'_i} = W_K(|C'_{<i+1}|, |C'_{i+1}|) Q_{C'_{i+1}} \tag{3.32}$$

Rewrite (3.32) as

$$\frac{Q_{C'_i} + Q_{C'_{i+1}}}{W_K(|C'_{<i+1}|, |C'_{i+1}|)} = \frac{Q_{C'_{i+1}}}{W_K(|C'_{<i}|, |C'_i|)} + \frac{Q_{C'_{i+1}}}{W_K(|C'_{<i+1}|, |C'_{i+1}|)} \tag{3.33}$$

By using Proposition (3.5), we have

$$\frac{1}{W_K(|C'_{<i+1}|, |C'_{i+1}|)}(Q_{C'_i} + Q_{C'_{i+1}}) = \frac{1}{W_K(|C'_{<i}|, |C'_i| + |C'_{i+1}|)} Q_{C'_{i+1}} \tag{3.34}$$

By using (3.31) and after proper arrangement, we have

$$W_K(|C_{<i}|, |C_i|)Q_{C_i} = W_K(|C'_{<i}| + |C'_i|, |C'_{i+1}|)Q_{C'_{i+1}} \tag{3.35}$$

So on the hyperplane $H_K(P; P')$ that separates the adjacent list pair P and P', we have

$$W_K(|C_{<i}|, |C_i|)Q_{C_i} = W_K(|C'_{<i+1}|, |C'_{i+1}|)Q_{C'_{i+1}} = W_K(|C'_{<i}|, |C'_i|)Q_{C'_i} \tag{3.36}$$

By (3.29),

$$\frac{\partial L_K(\boldsymbol{Q}; P)}{\partial Q_a} = W_K(|C_{<i}|, |C_i|)Q_{C_i} \tag{3.37}$$

So for $a \in C'_i$, on $H_K(P; P')$, we have

$$\frac{\partial L_K(\boldsymbol{Q}; P')}{\partial Q_a} = 2W_K(|C'_{<i}|, |C'_i|)Q'_{C_i} = 2W_K(|C_{<i}|, |C_i|)Q_{C_i} = \frac{\partial L_K(\boldsymbol{Q}; P)}{\partial Q_a} \tag{3.38}$$

Similarly, for $a \in C'_{i+1}$, on $H_K(P; P')$, we have

$$\frac{\partial L_K(\boldsymbol{Q}; P')}{\partial Q_a} = 2W_K(|C'_{<i+1}|, |C'_{i+1}|)Q_{C'_{i+1}} = 2W_K(|C_{<i}|, |C_i|)Q_{C_i} = \frac{\partial L_K(\boldsymbol{Q}; P)}{\partial Q_a} \tag{3.39}$$

This proves that they have equal gradient on the separation hyperplane $H_K(P; P')$.

Now we are going to show that on the hyperplane $H_K(P; P')$, we have

$$L_K(\boldsymbol{Q}; P') = L_K(\boldsymbol{Q}; P) \tag{3.40}$$

Notice that

$$
\begin{aligned}
&W_K(|C'_{<i}|,|C'_i|)Q^2_{C'_i} + W_K(|C'_{<i+1}|,|C'_{i+1}|)Q^2_{C'_{i+1}} \\
=&\frac{1}{W_K(|C'_{<i}|,|C'_i|)}(W_K(|C'_{<i}|,|C'_i|)Q_{C'_i})^2 \\
&+\frac{1}{W_K(|C'_{<i+1}|,|C'_{i+1}|)}(W_K(|C'_{<i+1}|,|C'_{i+1}|)Q_{C'_{i+1}})^2 \\
=&\left(\frac{1}{W_K(|C'_{<i}|,|C'_i|)}+\frac{1}{W_K(|C'_{<i+1}|,|C'_{i+1}|)}\right)(W_K(|C_{<i}|,|C_i|)Q_{C_i})^2 \\
=&\frac{1}{W_K(|C_{<i}|,|C_i|)}(W_K(|C_{<i}|,|C_i|)Q_{C_i})^2 \\
=&W_K(|C_{<i}|,|C_i|)Q^2_{C_i}
\end{aligned}
\tag{3.41}
$$

So

$$
\begin{aligned}
&L_K(\boldsymbol{Q};P) \\
=&\sum_{k=1}^{|P|} W_K(|C_{<k}|,|C_k|)Q^2_{C_k} \\
=&\sum_{k=1}^{|P|} W_K(|C_{<k}|,|C_k|)Q^2_{C_k} + W_K(|C_{<i}|,|C_i|)Q^2_{C_i} + \sum_{k=i+1}^{|P|} W_K(|C_{<k}|,|C_k|)Q^2_{C_k} \\
=&\sum_{k=1}^{i-1} W_K(|C'_{<k}|,|C'_k|)Q^2_{C'_k} + W_K(|C'_{<i}|,|C'_i|)Q^2_{C'_i} + W_K(|C'_{<i+1}|,|C'_{i+1}|)Q^2_{C'_{i+1}} \\
&+\sum_{k=i+2}^{|P'|} W_K(|C'_{<k}|,|C'_k|)Q^2_{C'_k} \\
=&\sum_{i=1}^{|P'|} W_K(|C'_{<i}|,|C'_i|)Q^2_{C'_i} \\
=&L_K(\boldsymbol{Q};P')
\end{aligned}
\tag{3.42}
$$

This completes the proof.

□

Now, we are ready to write the Lyapunov function given a complete priority list set $\mathbb{P}_{\mathcal{L}}$:

$$L_K(\boldsymbol{Q};\mathbb{P}_{\mathcal{L}}) := \sum_{P\in\mathbb{P}_{\mathcal{L}}} L_K(\boldsymbol{Q};P)\mathbb{I}\{\boldsymbol{Q}\in D_K^{\mathcal{L}}(P)\} \tag{3.43}$$

where $\mathbb{I}\{\boldsymbol{Q}\in D_K^{\mathcal{L}}(P)\}$ is indicator function, which equals 1 when $\boldsymbol{Q}\in D_K^{\mathcal{L}}(P)$ and 0 otherwise.

Proposition 3.8 $L_K(\boldsymbol{Q};\mathbb{P}_{\mathcal{L}})$ *is a well-defined continuous function with continuous gradient (aka. derivative).*

Proof: Follow from Proposition 3.6 and Proposition 3.7.

□

Remark: One might think of using (3.30) instead of (3.29) to define Lyapunov function for its simplicity in form. However this won't result in the nice continuity property on the boundary points between priority cones. The weight function $W_K(|C_{<i}|,|C_i|)$ is introduced to technically construct a Lyapunov function with continuity.

Given parameter K, consider objective function f_K is defined as follows: For each queue state $\boldsymbol{Q}\in\mathbb{R}_+^N$, if $\boldsymbol{Q}\in D_K^{\mathcal{L}}(P)$ with $P=(C_1,C_2,...,C_{|P|})$ then

$$f_K(\boldsymbol{Q},\boldsymbol{\mu}) := \sum_{i=1}^{|P|} W_K(|C_{<i}|,|C_i|)Q_{C_i}(\mu_{C_i,out}-\mu_{C_i,in}) \tag{3.44}$$

Recall that $\mathcal{M}(\boldsymbol{H},P)$ is the set of admissible routing control actions based on priority list P under topology $\boldsymbol{H}$. The following Lemma provides an alternative way to describe K policy: It is a policy that maximizes utilities function f_K.

Lemma 3.2 *Let $\mathcal{M}(\boldsymbol{H},P)$ be the set of admissible routing control actions based on priority list P under topology $\boldsymbol{H}$. Then for any K policy $\mathcal{P}_K$, we have*

$$\mathcal{M}(\boldsymbol{H},\mathcal{P}_K(\boldsymbol{Q})) \subseteq \arg\max_{\boldsymbol{\mu}\in\mathcal{M}(\boldsymbol{H})} f_K(\boldsymbol{Q},\boldsymbol{\mu}) \tag{3.45}$$

for all $\boldsymbol{Q} \in \mathbb{R}_+^N$*, where* $\mathcal{M}(\boldsymbol{H})$ *is the set of all admissible routing control actions that satisfies* (2.3) *and* (2.4) *under topology state* $\boldsymbol{H}$*.*

Proof: For easier presentation, define U_{C_i} as follows

$$U_{C_i} := \begin{cases} W_K(|C_{<i}|, |C_i|)Q_{C_i} & \text{if } 1 \le i \le |P| \\ 0 & \text{if } i = 0 \end{cases} \tag{3.46}$$

We adopt notation $[a]$ to denote the priority class node a belongs to. i.e. $[a] := C_i$ if $a \in C_i$. Hence we have $U_{[a]} = U_{C_i}$ if $a \in C_i$.

Now we rewrite (3.45) in another form:

$$\begin{aligned} f_K(\boldsymbol{Q}, \boldsymbol{\mu}) &:= \sum_{i=1}^{|P|} W_K(|C_{<i}|, |C_i|)Q_{C_i}(\mu_{C_i,out} - \mu_{C_i,in}) \\ &= \sum_{i=0}^{|P|} U_{C_i}(\mu_{C_i,out} - \mu_{C_i,in}) \\ &= \sum_{a \in \mathcal{N} \setminus C_0} \left\{ \sum_{b \in \mathcal{N} \setminus C_0} \mu_{ab} \left[U_{[a]} - U_{[b]} \right] + \sum_{b \in C_0} \mu_{ab} U_{[a]} \right\} \\ &= \sum_{a \in \mathcal{N} \setminus C_0} \left\{ \sum_{b \in \mathcal{N}} \mu_{ab} \left[U_{[a]} - U_{[b]} \right] \right\} \end{aligned} \tag{3.47}$$

where the second equation is due to the fact that $U_{C_0} := 0$.

By definition of $D_K^{\mathcal{L}}(P)$, we have the following constraint for any i, given a priority list P:

$$W_K(|C_{<i}|, |C_i|)Q_{C_i} \le W_K(|C_{<i+1}|, |C_{i+1}|)Q_{C_{i+1}} \tag{3.48}$$

(3.48) is obtained by considering the separation hyperplane between P and all its one-step confinement lists (By Proposition 3.3, all confinement lists of P are critical)

Apply the definition of U_{C_i} in (3.46) to (3.48), we have

$$U_{C_0} \le U_{C_1} \le ... \le U_{C_{|P|}} \tag{3.49}$$

By the routing constraints defined in (2.3) and (2.4), under topology state $\boldsymbol{H}$, for each node a, there exists one and only one node b from the potential forwarder set of a (b may be equal to a) such that $\mu_{ab} = 1$. So we have

$$\begin{aligned}
&\max_{\boldsymbol{\mu}\in\mathcal{M}(\boldsymbol{H})} f_K(\boldsymbol{Q},\boldsymbol{\mu}) \\
&= \max_{\boldsymbol{\mu}\in\mathcal{M}(\boldsymbol{H})} \sum_{a\in\mathcal{N}\backslash C_0} \left\{ \sum_{b\in\mathcal{N}} \mu_{ab} \left[U_{[a]} - U_{[b]} \right] \right\} \\
&= \sum_{a\in\mathcal{N}\backslash C_0} \max_{b\in\mathcal{S}_a(t)} \left\{ U_{[a]} - U_{[b]} \right\} \\
&= \sum_{a\in\mathcal{N}\backslash C_0} \left\{ U_{[a]} - \min_{b\in\mathcal{S}_a(t)} U_{[b]} \right\}
\end{aligned} \tag{3.50}$$

From (3.50), it is clear that to maximize f_K under topology state $\boldsymbol{H}$, a should forward its packet to the node in the lowest priority class among its forwarders. And this is exactly how K policy is defined.

□

3.3.2 Proof of Stability

The following simple Lemma will be very useful. It serves as a basic building block for proving our main results

Lemma 3.3 *If Q^+, Q, μ, A are nonnegative real random variables, and there exist nonnegative real numbers v, c such that $\mu \leq c$,$A \leq c$ and the following relations hold:*

$$Q^+ \leq Q + A \tag{3.51}$$

$$Q^+ \leq Q - \mu + A, \quad \textit{if} \quad Q > v \tag{3.52}$$

then there exists a constant β such that

$$(Q^+)^2 - Q^2 \leq \beta - 2Q(\mu - A) \tag{3.53}$$

Proof: If $Q > v$, we have

$$Q^+ \leq Q - \mu + A \tag{3.54}$$

After taking the square of (3.54) at both sides and proper arrangement, we have.

$$(Q^+)^2 - Q^2 \leq (\mu - A)^2 - 2Q(\mu - A) \tag{3.55}$$

On the other hand, if $Q \leq v$, by (3.51), we still have

$$Q^+ \leq Q + A \tag{3.56}$$

After taking the square of (3.56) at both sides and proper arrangement, we have.

$$(Q^+)^2 - Q^2 \leq A^2 + 2Q\mu - 2Q(\mu - A) \tag{3.57}$$

Let $\beta := 4c^2 + 2vc$ and notice that

$$\beta \geq (\mu - A)^2 \tag{3.58}$$

and

$$\beta \geq A^2 + 2Q\mu, \quad \text{if} \quad Q < v. \tag{3.59}$$

The proof is completed by applying (3.58) and (3.59) to (3.55) and (3.57), respectively.

□

The following Lemma gives queue dynamics when the total backlog within a priority class is sufficiently large.

Lemma 3.4 *If K policy $\mathcal{P}_K$ $(K \geq 2)$ based on $\mathbb{P}_{\mathcal{L}}$ is used in a network with link set $\mathcal{L}'$ and $\mathcal{L}' \subseteq \mathcal{L}$, then there exists a constant α_K such that for any priority class C_i from any priority list $P = (C_1, C_2, ..., C_{|P|}) \in \mathbb{P}_{\mathcal{L}}$, as long as $\boldsymbol{Q}(t) \in D_K^{\mathcal{L}}(P)$ and $Q_{C_i}(t) > \alpha_K$, the queue dynamics can be written as*

$$Q_{C_i}(t+1) \leq Q_{C_i}(t) - \mu_{C_i,out}(t) + \mu_{C_i,in}(t) + A_{C_i}(t) \tag{3.60}$$

where $\mu_{C_i,in}(t)$, $\mu_{C_i,out}(t)$, and A_{C_i} are defined in (3.3) (3.4), and (3.1), respectively.

Proof: It is sufficient to prove that when Q_{C_i} is sufficiently large, all non-isolated nodes in priority class C_i under link set $\mathcal{L}'$ will become larger than μ_{max}. So these queues will not become empty (no edge effect). Then (3.60) must hold.

First notice that by Proposition 3.4, a non-isolated node in priority class C_i under link set $\mathcal{L}'$ is also non-isolated under link set $\mathcal{L}$ since $\mathcal{L}' \subseteq \mathcal{L}$. So it is sufficient to prove that when Q_{C_i} is sufficiently large, all non-isolated nodes in priority class C_i under link set $\mathcal{L}$ will become larger than μ_{max}.

For any non-isolated node $a \in C_i$ under $\mathcal{L}$, since $P \in \mathbb{P}_{\mathcal{L}}$ is critical, by Lemma 3.1 (1),

$$Q_a \geq \frac{K-1}{K^{|C_i|}-1} Q_{C_i} \tag{3.61}$$

Now choose $\alpha_K > \mu_{max} \frac{K^{|C_i|}-1}{K-1}$. Then $Q_{C_i} > \alpha_K$ implies $Q_a > \mu_{max}$. We can repeat this procedure for all priority classes of all priority lists. Since the number of priority classes and number of priority lists are finite, there exists a single constant α_K, such that for any priority class of any priority lists, if $\boldsymbol{Q}(t) \in D_K^{\mathcal{L}}(P)$ and $Q_{C_i}(t) > \alpha_K$, then all non-isolated nodes in C_i have queue length greater than μ_{max}. Thus these nodes will not go to empty from time t to time $t+1$. Therefore

$$Q_{C_i}(t+1) \leq Q_{C_i}(t) - \mu_{C_i,out}(t) + \mu_{C_i,in}(t) + A_{C_i}(t) \tag{3.62}$$

□

Lemma 3.5 *If K policy $\mathcal{P}_K$ ($K \geq 2$) based on $\mathbb{P}_{\mathcal{L}}$ is used in a network with link set $\mathcal{L}'$ and $\mathcal{L}' \subseteq \mathcal{L}$, then for any priority list $P = (C_1, C_2, ..., C_{|P|}) \in \mathbb{P}_{\mathcal{L}}$, and $\boldsymbol{Q}(t) \in D_K^{\mathcal{L}}(P)$*

$$Q_{C_i}^2(t+1) - Q_{C_i}^2(t) \leq \beta_K - 2Q_{C_i}(t)\left(\mu_{C_i,out}(t) - \mu_{C_i,in}(t) - A_{C_i}(t)\right) \tag{3.63}$$

where β_K is some constant.

Proof:

By Lemma 3.4, when K policy is used, there exists α_K such that if $\boldsymbol{Q}(t) \in D_K^{\mathcal{L}}(P)$ and $Q_{C_i}(t) > \alpha_K$, then

$$Q_{C_i}(t+1) \leq Q_{C_i}(t) - \mu_{C_i,out}(t) + \mu_{C_i,in}(t) + A_{C_i}(t) \tag{3.64}$$

On the other hand, if $Q_{C_i}(t) \leq \alpha_K$, then

$$Q_{C_i}(t+1) \leq Q_{C_i}(t) + \mu_{C_i,in}(t) + A_{C_i}(t) \tag{3.65}$$

(3.63) follows by applying Lemma 3.3 with $Q^+ = Q_{C_i}(t+1)$, $Q = Q_{C_i}(t)$, $A = \mu_{C_i,in}(t) + A_{C_i}(t)$, $\mu = \mu_{C_i,out}(t)$, $v = \alpha_K$, and $c = N \max\{\mu_{max}, A_{max}\}$.

□

Note that $\boldsymbol{Q}(t) \in D_K^{\mathcal{L}}(\mathcal{P}_K(\boldsymbol{Q}(t)))$, but it is not necessarily true that $\boldsymbol{Q}(t+1) \in D_K^{\mathcal{L}}(\mathcal{P}_K(\boldsymbol{Q}(t)))$. This is because the queues at time slot $t+1$ might cross the boundary points of the priority cone and the priority list might change. However, the following Lemma shows that due to the differentiability of the Lyapunov function, the error introduced by crossing boundary is of a higher order.

Lemma 3.6 *If K policy $\mathcal{P}_K$ $(K \geq 2)$ based on $\mathbb{P}_{\mathcal{L}}$ is used in a network with link set $\mathcal{L}'$ and $\mathcal{L}' \subseteq \mathcal{L}$, then for any priority list $P = (C_1, C_2, ..., C_{|P|}) \in \mathbb{P}_{\mathcal{L}}$, and $\boldsymbol{Q}(t) \in D_K^{\mathcal{L}}(P)$*

$$L_K(\boldsymbol{Q}(t+1); \mathbb{P}_{\mathcal{L}}) - L_K(\boldsymbol{Q}(t); \mathbb{P}_{\mathcal{L}}) \leq$$
$$\gamma_K - 2\sum_i W_K(|C_{<i}|, |C_i|) Q_{C_i}(t)(\mu_{C_i,out}(t) - \mu_{C_i,in}(t) - A_{C_i}(t)) + o(\|\boldsymbol{Q}(t+1) - \boldsymbol{Q}(t)\|)$$

where γ_K is some constant.

Proof:

It follows from Lemma 3.5 and (3.29) that

$$L_K(\boldsymbol{Q}(t+1);P) - L_K(\boldsymbol{Q}(t);P)$$
$$\leq \gamma_K - 2\sum_i W_K(|C_{<i}|,|C_i|)Q_{C_i}(t)(\mu_{C_i,out}(t) - \mu_{C_i,in}(t) - A_{C_i}(t)) \quad (3.66)$$

for $\boldsymbol{Q}(t) \in D_K^{\mathcal{L}}(P)$, where γ_K is some constant.

Since $L_K(\boldsymbol{Q}(t);P)$ is differentiable, we have

$$L_K(\boldsymbol{Q}(t+1);P) = L_K(\boldsymbol{Q}(t);P) + \nabla L_K(\boldsymbol{Q}(t);P)\cdot(\boldsymbol{Q}(t+1) - \boldsymbol{Q}(t))$$
$$+ o(\|\boldsymbol{Q}(t+1) - \boldsymbol{Q}(t)\|) \quad (3.67)$$

Likewise, $L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}})$ is differentiable by Proposition 4.2. So

$$L_K(\boldsymbol{Q}(t+1);\mathbb{P}_{\mathcal{L}}) = L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}}) + \nabla L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}})\cdot(\boldsymbol{Q}(t+1) - \boldsymbol{Q}(t))$$
$$+ o(\|\boldsymbol{Q}(t+1) - \boldsymbol{Q}(t)\|) \quad (3.68)$$

Subtracting (3.67) from (3.68) and noting that $L_K(\boldsymbol{Q}(t);P) = L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}})$ and $\nabla L_K(\boldsymbol{Q}(t);P) = \nabla L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}})$ for $\boldsymbol{Q}(t) \in D_K^{\mathcal{L}}(P)$, we have

$$L_K(\boldsymbol{Q}(t+1);\mathbb{P}_{\mathcal{L}}) - L_K(\boldsymbol{Q}(t+1);P) \leq o(\|\boldsymbol{Q}(t+1) - \boldsymbol{Q}(t)\|) \quad (3.69)$$

The proof is completed by adding (3.66) and (3.69) and substituting $L_K(\boldsymbol{Q}(t);P)$ by $L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}})$.

□

Now, we are ready to prove Theorem 3.1.

Proof of Theorem 3.1:

Suppose K policy is used, and $\boldsymbol{Q}(t) \in D_K^{\mathcal{L}}(P)$, where $P = (C_1, C_2, ..., C_{|P|})$, then by Lemma 3.6:

$$L_K(\boldsymbol{Q}(t+1);\mathbb{P}_{\mathcal{L}}) - L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}}) \leq$$
$$\gamma_K - 2\sum_i W_K(|C_{<i}|,|C_i|)Q_{C_i}(t)(\mu_{C_i,out}(t) - \mu_{C_i,in}(t) - A_{C_i}(t)) + o(\|\boldsymbol{Q}(t+1) - \boldsymbol{Q}(t)\|)$$

Taking the conditional expectation yields,

$$\begin{aligned}
&\Delta L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}}) = \mathbb{E}\{L_K(\boldsymbol{Q}(t+1);\mathbb{P}_{\mathcal{L}}) - L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}})|\boldsymbol{Q}(t)\} \\
\leq& \gamma_K - 2\sum_i W_K(|C_{<i}|,|C_i|)Q_{C_i}(t)\mathbb{E}\left\{(\mu_{C_i,out}(t) - \mu_{C_i,in}(t) - A_{C_i}(t))|\boldsymbol{Q}(t)\right\} \\
&+ o(\|\boldsymbol{Q}(t+1) - \boldsymbol{Q}(t)\|) \\
=& \gamma_K - 2\mathbb{E}[f_K(\boldsymbol{Q}(t),\boldsymbol{\mu}(t))] - \sum_i (W_K(|C_{<i}|,|C_i|)\lambda_{C_i}) + o(\|\boldsymbol{Q}(t+1) - \boldsymbol{Q}(t)\|)
\end{aligned} \tag{3.70}$$

Because the control action $\boldsymbol{\mu}(t)$ in (3.70) has to be admissive, we require $\boldsymbol{\mu}(t) \in \mathcal{M}(\boldsymbol{H}(t))$. This implies that the set of feasible values $\boldsymbol{\mu}(t)$ could take depends on the topology state $\boldsymbol{H}(t)$, which is a random variable. Hence, we conclude that $f_K(\boldsymbol{Q}(t),\boldsymbol{\mu}(t))$ is a random variable.

Since $\boldsymbol{\lambda}$ is strictly within the capacity region Λ, there exist a positive vector $\epsilon > 0$ such that $\boldsymbol{\lambda} + \boldsymbol{\epsilon} \in \Lambda$. By Theorem 2.4 there exists a stationary randomized algorithm that makes decisions based only on the current topology state (and hence independent of the current queue backlog) so that

$$\mathbb{E}\left\{(\tilde{\mu}_{C_i,out}(t) - \tilde{\mu}_{C_i,in}(t) - A_{C_i}(t))|\boldsymbol{Q}(t)\right\} \geq \epsilon \tag{3.71}$$

where $\tilde{\mu}_{C_i,out}(t)$ and $\tilde{\mu}_{C_i,out}(t)$ denote the amount of date transmitted under such randomized policy.

By the definition of f_K in (3.45) and Lemma 3.2, K policy $\mathcal{P}_K$ chooses routing control action $\boldsymbol{\mu}$ to minimize the right hand side over all polices. i.e. if $\hat{\boldsymbol{\mu}}(t) \in \mathcal{M}(\boldsymbol{H}(t),\mathcal{P}_K(\boldsymbol{Q}(t)))$, then we have

$$f_K(\boldsymbol{Q}(t),\tilde{\boldsymbol{\mu}}(t)) \leq \max_{\boldsymbol{\mu}(t)\in\mathcal{M}(\boldsymbol{H}(t))} f_K(\boldsymbol{Q}(t),\boldsymbol{\mu}(t)) = f_K(\boldsymbol{Q}(t),\hat{\boldsymbol{\mu}}(t)) \tag{3.72}$$

where $\tilde{\boldsymbol{\mu}}(t)$ denote the control action under the randomized policy. By applying (3.72) to (3.70) we have

$$\begin{aligned}
&\Delta L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}}) \\
\leq&\gamma_K - 2\mathbb{E}\left[f_K(\boldsymbol{Q}(t),\hat{\boldsymbol{\mu}}(t))\right] - \sum_i \left(W_K(|C_{<i}|,|C_i|)\lambda_{C_i}\right) + o(\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|) \\
\leq&\gamma_K - 2\mathbb{E}\left[f_K(\boldsymbol{Q}(t),\tilde{\boldsymbol{\mu}}(t))\right] - \sum_i \left(W_K(|C_{<i}|,|C_i|)\lambda_{C_i}\right) + o(\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|) \\
=&\gamma_K - 2\sum_i W_K(|C_{<i}|,|C_i|)Q_{C_i}(t)\mathbb{E}\left\{(\tilde{\mu}_{C_i,out}(t) - \tilde{\mu}_{C_i,in}(t) - A_{C_i}(t))|\boldsymbol{Q}(t)\right\} \\
&+ o(\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|)
\end{aligned} \tag{3.73}$$

By using (3.71), we have

$$\Delta L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}}) \leq \gamma_K - 2\epsilon\sum_i W_K(|C_{<i}|,|C_i|)Q_{C_i}(t) + o(\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|) \tag{3.74}$$

Note that $W_K(|C_{<i}|,|C_i|) \geq 1$ for any K, C_i, so

$$\begin{aligned}
&\mathbb{E}\{L_K(\boldsymbol{Q}(t+1);\mathbb{P}_{\mathcal{L}}) - L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}})|\boldsymbol{Q}(t)\} \\
\leq&\gamma_K - 2\epsilon\sum_i Q_{C_i}(t) + o(\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|) \\
=&\gamma_K - 2\epsilon\sum_a Q_a(t) + o(\frac{\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|}{\sum_a Q_a(t)})\sum_a Q_a(t) \\
=&\gamma_K - 2\left(\epsilon - o(\frac{\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|}{\sum_a Q_a(t)})\right)\sum_a Q_a(t)
\end{aligned} \tag{3.75}$$

Since $\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|$ is bounded, $\frac{\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|}{\sum_a Q_a(t)} \to 0$ as $\sum_a Q_a(t)$ increases. Therefore, for sufficiently large $\sum_a Q_a(t)$, $o(\frac{\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|}{\sum_a Q_a(t)}) \leq \frac{\epsilon}{2}$. Thus we conclude for sufficiently large $\sum_a Q_a(t)$, we have

$$\mathbb{E}\{L_K(\boldsymbol{Q}(t+1);\mathbb{P}_{\mathcal{L}}) - L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}})|\boldsymbol{Q}(t)\} \leq \gamma_K - \epsilon\sum_a Q_a(t) \tag{3.76}$$

The stability follows from Lemma 2.1.

□

3.4 Examples of K Policies

In this section, we give a few toy examples to get some intuition and insights of the structure of K policy.

Example 1:

Consider the network of 3 nodes as given in Figure 3.3.

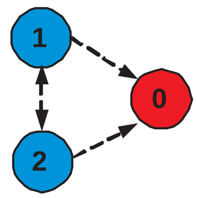

Figure 3.3: A 3 nodes network, $(N = 2)$. Node 0 is the destination node

The underlying Lyapunov function of the network (scaled by factor $K^2 - 1$) is given by

$$L(\boldsymbol{Q}) = \begin{cases} (K+1)Q_1^2 + \frac{K+1}{K}Q_2^2 & Q_2 > KQ_1 \\ \frac{K+1}{K}Q_1^2 + (K+1)Q_2^2 & Q_1 > KQ_2 \\ (Q_1 + Q_2)^2 & \text{otherwise} \end{cases} \tag{3.77}$$

Figure 3.4 shows the priority cone partition and corresponding priority list assigned for each cone, where the solid curves are the contour lines of underlying Lyapunov function. As we can see that the contour lines are smooth, due to the fact that the Lyapunov function is continuous and differentiable.

Note that in the central cone where $Q_2 \leq KQ_1$ and $Q_1 \leq KQ_2$, the two nodes are grouped into a single priority class. So the transmission between these two nodes are not specified by K policy. Compared with backpressure algorithm, K policy provides more flexibility in designing a throughput optimal policy with potentially better delay performance thanks to the existence of the central cone.

It is easy to see that the central cone becomes "narrower" when K is pushed towards 1 and completely vanishes when $K = 1$. The resulting cones are depicted in

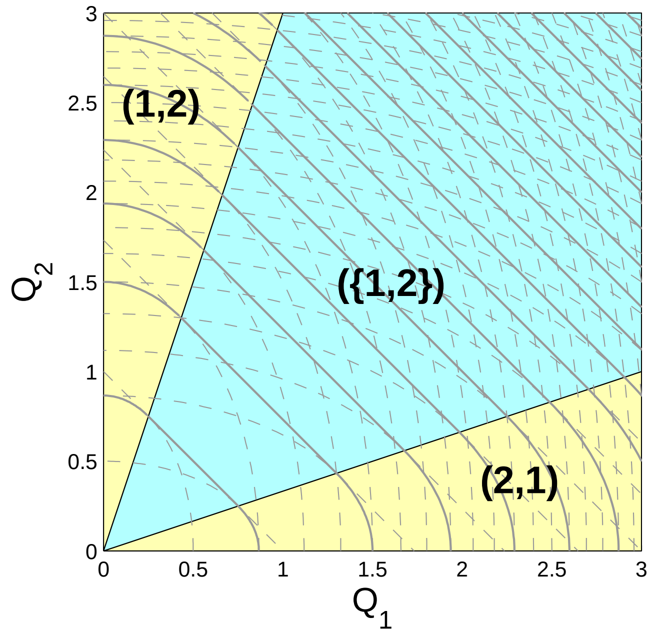

Figure 3.4: Priority cone partition and contour lines of underlying Lyapunov function of network in Figure 3.3

Figure 3.5. Now K policy is reduced to backpressure algorithm, and the underlying Lyapunov function is simply in the quadratic form: $L(\boldsymbol{Q}) = Q_1^2 + Q_2^2$. It turns out such an observation holds for networks with any size. Indeed it can be shown that backpressure algorithm is K policy based on $\mathbb{P}_{all}$ when $K = 1$. This in one way, implies that backpressure algorithm is a special case of K policy. We will provide an alternative way to show that backpressure algorithm is a special case of K policy later in this chapter.

Example 2:

Now consider the network of 3 nodes as given in Figure 3.6. Unlike Example 1, now node 2 cannot transmit to node 0 directly (i.e. $(2,0) \notin \mathcal{L}$). So priority list $(2,1)$ is non-critical.

The underlying Lyapunov function of the network (scaled by factor $K^2 - 1$)

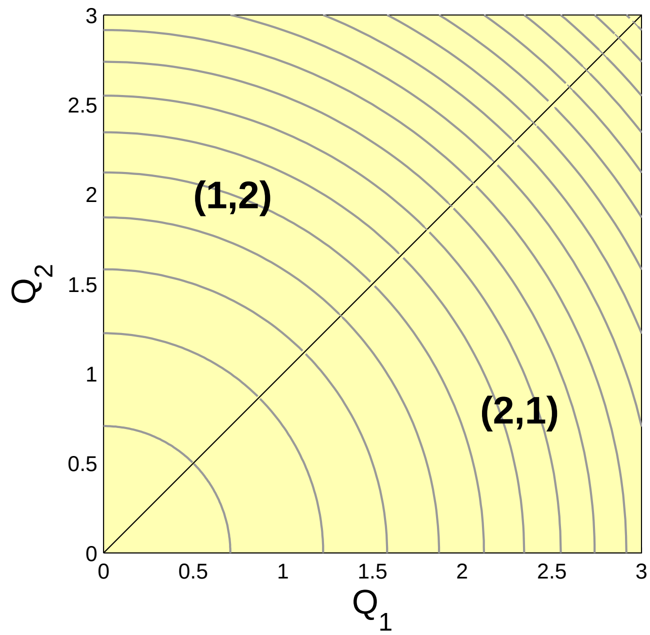

Figure 3.5: Priority cone partition and contour lines of underlying Lyapunov function of network when $K = 1$

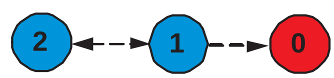

Figure 3.6: A 3 nodes network, $(N = 2)$. Node 0 is the destination node

is given by

$$L(\boldsymbol{Q}) = \begin{cases} (K+1)Q_1^2 + \frac{K+1}{K}Q_2^2 & Q_2 > KQ_1 \\ (Q_1 + Q_2)^2 & \text{otherwise} \end{cases} \quad (3.78)$$

Figure 3.7 shows the priority cone partition and contour lines of underlying Lyapunov function. It can be seen that even Q_1 is K times larger than Q_2, K policy will not necessarily route packet from node 1 to node 2 since priority list $(2, 1)$ is non-critical. This could potentially reduces redundant routing that causes poor delay performance in backpressure algorithm.

Example 3:

Consider the network of 4 nodes as given in Figure 3.8.

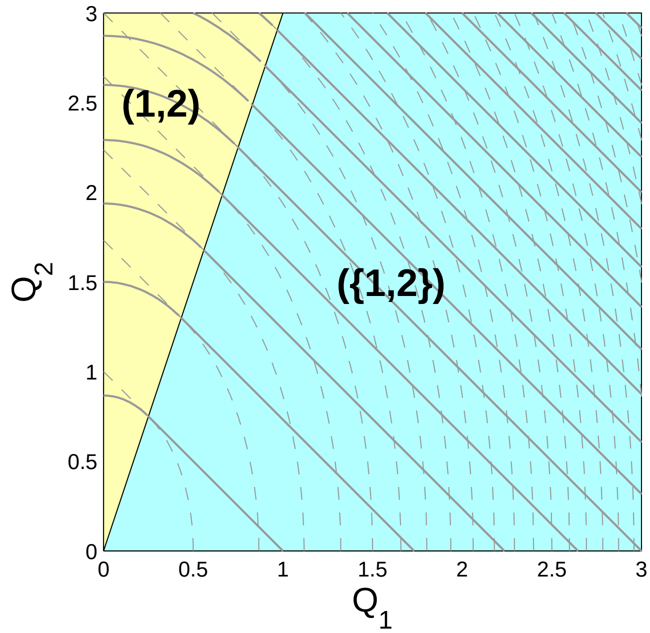

Figure 3.7: Priority cone partition and contour lines of underlying Lyapunov function of network in Figure 3.6

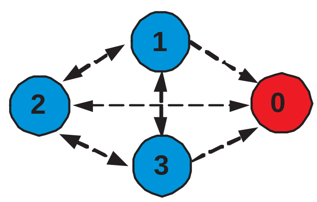

Figure 3.8: A 4 nodes network, ($N = 3$). Node 0 is the destination node

Figure 3.9 shows the priority cone partition and corresponding priority list assigned for each cone.

Example 4:

Consider the network of 4 nodes as given in Figure 3.10.

Unlike Example 3, some priority lists are not critical due to the fact that node 2 is not able to transmit to destination node 0 directly. The "tree" of priority lists is shown in Figure 3.11. Each arrow in Figure 3.11 points from a list to one of its one-step refinement lists.

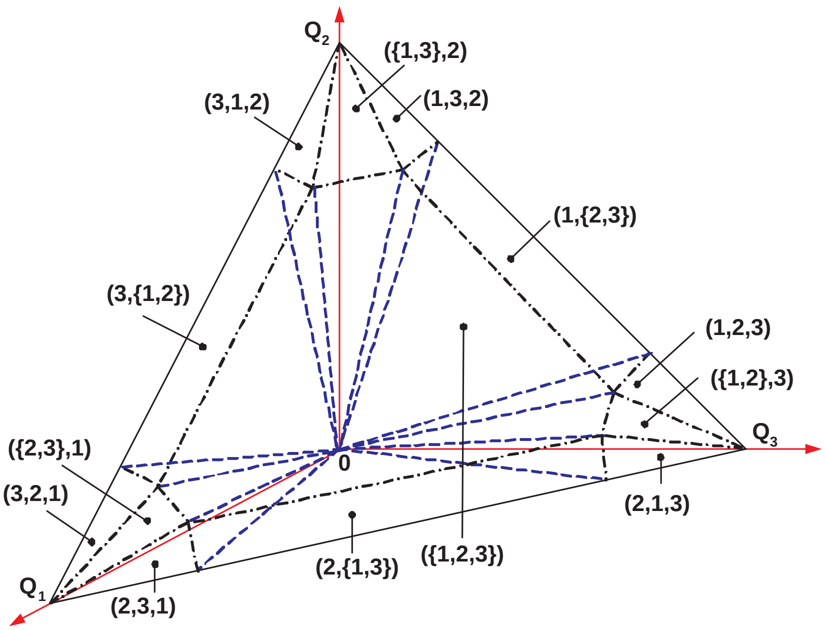

Figure 3.9: Priority cones of the 4 nodes network, $(N = 3)$

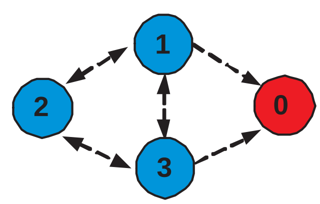

Figure 3.10: A 4 nodes network, $(N = 3)$. Node 0 is the destination node

Figure 3.12 shows the priority cone partition and corresponding priority list assigned for each cone.

As shown in Figure 3.4, Figure 3.7, Figure 3.9, and Figure 3.12, K policy groups the queues based on their backlogs. In the central cone, all nodes belong to the same priority class. The Lyapunov function in this cone is the squared sum of all queue backlogs. Since the Lyapunov drift won't be affected by any packet forwarding within the same priority class, it allows a routing policy to potentially deviate from backpressure decisions to arrive at a better delay performance, However, when one of the queues becomes relatively large in comparison to the other

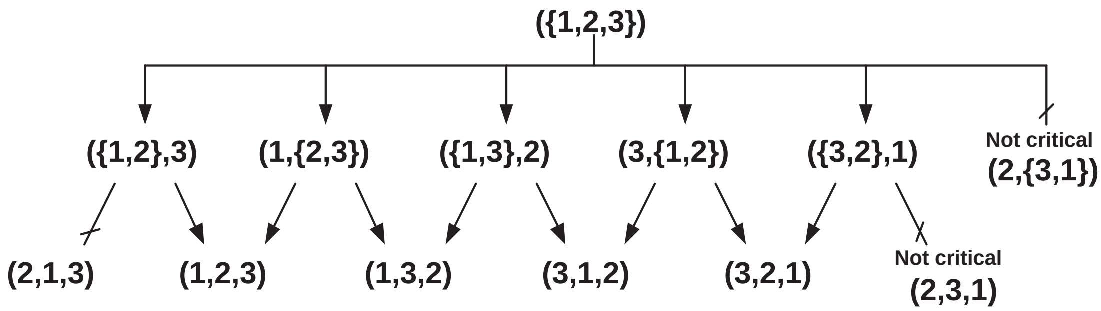

Figure 3.11: Priority lists of the 4 nodes network, $(N = 3)$

nodes's backlogs, the backlog vector falls in a cone in which the node with large backlog is in a separate priority class. The Lyapunov function for this cone, now, is the squared queue backlog of the node with large backlog plus the squared sum of other queue backlogs and its negative drift is ensured only when packets are routed from the node with disportionately large backlog to other nodes. Similarly one can analyze the behavior of the Lyapunov function in other cones. This structure allows a routing policy to potentially achieve a better delay performance, while at the same time still maintaining the stability of the network whenever possible.

3.5 Extension

3.5.1 On Choice of $W_K(n, m)$

We have previous defined $W_K(n, m)$ as

$$W_K(n,m) := \frac{1}{K^n(K^m - 1)} \tag{3.79}$$

It turns out that (3.79) is not the unique choice. From the proof of Proposition 3.6, it is not difficult to see that any positive bivariable integer function W : $\mathbb{N} \cup \{0\} \times \mathbb{N} \mapsto \mathbb{R}^+$ that satisfies

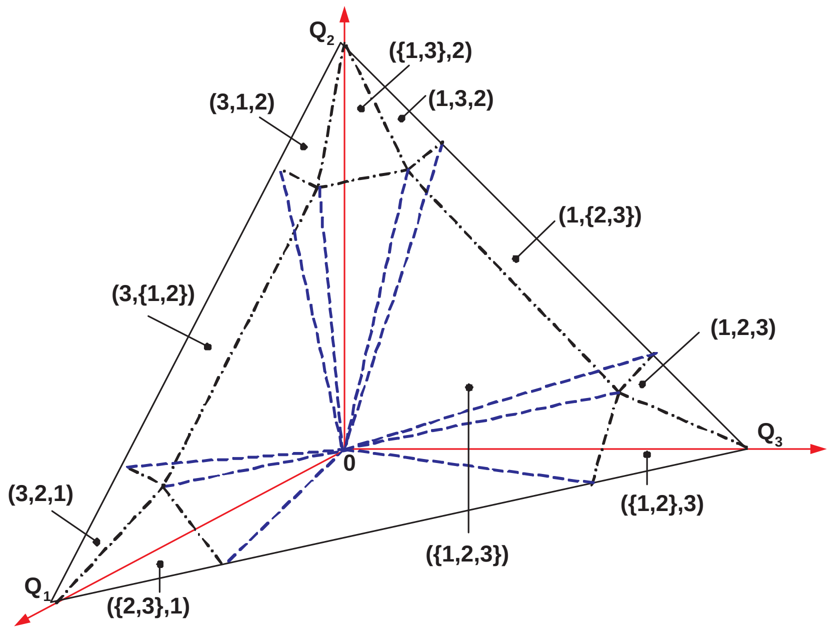

Figure 3.12: Priority cones of the 4 nodes network, $(N = 3)$

1)

$$\frac{1}{W(n, m_1 + m_2)} = \frac{1}{W(n, m_1)} + \frac{1}{W(n + m_1, m_2)} \tag{3.80}$$

2)

$$\frac{W(0, n)}{W(n, 1)} > 1 \tag{3.81}$$

will result in a complete partition of $\mathbb{R}_+^N$ and a K policy defined on the partition is throughput optimal. Note that condition 2) is sufficient but not necessary for a non-overlapped partition.

From (3.80), we have

$$\frac{1}{W(n, m)} = \frac{1}{W(0, m + n)} - \frac{1}{W(0, n)} \tag{3.82}$$

From (3.81) and (3.80), we have

$$\frac{1}{W(n, 1)} = \frac{1}{W(0, n + 1)} - \frac{1}{W(0, n)} > \frac{1}{W(0, n)} \tag{3.83}$$

i.e.

$$\frac{W(0, n)}{W(0, n + 1)} > 2 \tag{3.84}$$

From (3.82) and (3.84), it can be seen that function W can be uniquely determined by values of $W(0,n)$, $n = 1,2,3...$

The function can be constructed as follows:

1) For each $n = 1,2,3...$, assign a positive value to $W(0,n)$, such that $W(0,n)$ is monotonically decreasing in n and

$$\frac{W(0,n)}{W(0,n+1)} > 2 \tag{3.85}$$

2) For any n,m, $W(n,m)$ is defined by

$$W(n,m) := \left(\frac{1}{W(0,m+n)} - \frac{1}{W(0,n)}\right)^{-1} \tag{3.86}$$

$W(n,m)$ is ensured to be positive since $W(0,n)$ is monotonically decreasing in n.

Now we verify that such a construction indeed satisfies (3.80) and (3.81): For any n,m_1 and m_2, by construction,

$$\frac{1}{W(n,m_1+m_2)} = \frac{1}{W(0,n+m_1+m_2)} - \frac{1}{W(0,n)} \tag{3.87}$$

$$\frac{1}{W(n,m_1)} = \frac{1}{W(0,n+m_1)} - \frac{1}{W(0,n)} \tag{3.88}$$

and

$$\frac{1}{W(n+m_1,m_2)} = \frac{1}{W(0,n+m_1+m_2)} - \frac{1}{W(0,n+m_1)} \tag{3.89}$$

(3.80) follows as (3.87) = (3.88) +(3.89). And by construction, (3.81) is trivially satisfied.

In this paper, we have used $W(0,n) = \frac{1}{K^n-1}$. To satisfy (3.81), we require

$$\frac{K^{n+1}-1}{K^n-1} > 2 \tag{3.90}$$

which holds when $K \geq 2$ (Sufficient but not necessary condition).

3.5.2 Shifted K policy

Consider a vector variable $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_N] \in \mathbb{R}_N^+$. Given a link set $\mathcal{L}$, the shifted K policy $\mathcal{P}_K$ based on critical list set $\mathbb{P}_{\mathcal{L}}$ is defined as

$$\mathcal{P}_K(\boldsymbol{Q}) = P \quad \text{if} \quad \boldsymbol{Q} + \boldsymbol{\theta} \in D_K^{\mathcal{L}}(P) \tag{3.91}$$

In other words, the policy $\mathcal{P}_K$ routes the packet according to priority list P when the shifted queue state $\boldsymbol{Q}(t) + \boldsymbol{\theta}$ belongs to $D_K^{\mathcal{L}}(P)$ at time slot t.

Proposition 3.9 *Shifted K policy with $\boldsymbol{\theta} \geq 0$ is throughput optimal.*

Proof:

Consider the shifted queues $\boldsymbol{Q}'(t) := \boldsymbol{Q}(t) + \boldsymbol{\theta}$. The one-step dynamics of $\boldsymbol{Q}'(t)$ is given by

$$Q'_a(t+1) \leq \max\{Q'_a(t) - \mu_{a,out}(t), \theta_a\} + \mu_{a,in}(t) + A_a(t) \tag{3.92}$$

Note that $Q'_a(t) \geq \theta_a$, so we have

$$Q'_a(t+1) \leq Q'_a(t) + \mu_{a,in}(t) + A_a(t) \tag{3.93}$$

Hence

$$Q'_{C_i}(t+1) \leq Q'_{C_i}(t) + \mu_{C_i,in}(t) + A_{C_i}(t) \tag{3.94}$$

By Lemma 3.4, if $\boldsymbol{Q}'(t) \in D_K^{\mathcal{L}}(P)$, there exists α_K such that when $Q'_{C_i}(t) > \alpha_K$, all non-isolated nodes in priority class C_i will be larger than $\mu_{max} + \theta_{max}$. So the queue dynamic of Q'_a is given by

$$Q'_a(t+1) \leq Q'_a(t) - \mu_{a,out}(t) + \mu_{a,in}(t) + A_a(t) \tag{3.95}$$

and hence when $Q'_{C_i}(t) > \alpha_K$, the queue dynamics of $Q'_{C_i}(t)$ can be written as

$$Q'_{C_i}(t+1) \leq Q'_{C_i}(t) - \mu_{C_i,out}(t) + \mu_{C_i,in}(t) + A_{C_i}(t) \tag{3.96}$$

Apply Lemma 3.3 to (3.94) and (3.96), we have a result similar to Lemma 3.5:

$$Q'^2_{C_i}(t+1) - Q'^2_{C_i}(t) \leq \beta_K - 2Q'_{C_i}(t)\left\{\mu_{C_i,out}(t) - \mu_{C_i,in}(t) - A_{C_i}(t)\right\} \tag{3.97}$$

The proof of throughput optimality of shifted K policy then can be completed by following the same proof as that of Theorem 3.1, except that $\boldsymbol{Q}(t)$ is replaced with $\boldsymbol{Q}'(t)$.

□

3.6 Stability of General Priority Policy

In previous sections, we have showed that a wide class of priority policies called K policy is throughput optimal. Now we utilize K policy to show the stability of other priority policies. Particularly, we will specialize our result to recover the throughput optimality of two known routing policies, backpressure (already known to be throughput optimal) and ORCD (whose throughput optimality only was conjectured in [21]).

By Proposition 3.2, given a priority policy $\mathcal{P}$, if there exists a K policy $\mathcal{P}_K$ for sufficiently large K, such that $\mathcal{P}$ is a refined policy of $\mathcal{P}_K$, then priority policy $\mathcal{P}$ is throughput optimal. Since the domain of $\boldsymbol{Q}$ is divided into *finite* number of cones by K policy, each corresponding to one critical priority list, we have the following Theorem that gives the condition under which a general routing policy is throughput optimal

Theorem 3.2 *For a general priority policy $\mathcal{P}$, if there exists a priority list set $\mathbb{P}_{\mathcal{L}}$ and for each $P \in \mathbb{P}$, there exist $\boldsymbol{\theta} \geq 0$ and sufficiently large K such that $\mathcal{P}(\boldsymbol{Q})$ is a refinement of P as long as $\boldsymbol{Q} + \boldsymbol{\theta} \in D^{\mathcal{L}}_K(P)$, then priority policy $\mathcal{P}$ is throughput optimal.*

As a simple (and maybe convoluted) application of Theorem 3.2, we show that backpressure policy $\mathcal{P}_{BP}$ is throughput optimal.

Recall that $\mathcal{L}_{all}$ denote the set of all possible links in a network. It is easy to see that $\mathbb{P}_{all} = \mathbb{P}_{\mathcal{L}_{all}}$. We will show that backpressure policy $\mathcal{P}_{BP}$ is a refined policy of K policy based on $\mathbb{P}_{all}$. And therefore by Theorem 3.2, it is throughput optimal for any network. (Since any network can be viewed as a sub-network of $\mathcal{L}_{all}$.)

To see why backpressure policy is a refined policy of K policy based on $\mathbb{P}_{all}$, first notice that $\mathbb{P}_{all}$ contains all possible priority lists and all nodes in a priority list are non-isolated, since $\mathcal{L}_{all}$ contains all possible links. So by Lemma 3.1, for $K \geq 2$, $Q_a \in C_i$ and $Q_b \in C_{<i}$, we have $Q_a \geq (K-1)Q_{C_{<i}} \geq Q_b$. It is well known that backpressure policy is a priority policy that ranks the nodes according to their backlogs. So backpressure policy will assign all nodes in C_i higher priority than nodes in $C_{<i}$. This implies that $\mathcal{P}_{BP}(\boldsymbol{Q})$ is a refinement of P as long as $\boldsymbol{Q} \in D_K^{\mathcal{L}}(P)$ and $K \geq 2$. i.e. backpressure policy $\mathcal{P}_{BP}$ is a refined policy of K policy based on $\mathbb{P}_{all}$.

To get some more intuitions, let us consider a simple 3 nodes network in Figure 3.13. We can graphically see that both backpressure and ORCD algorithm (which will be formally discussed in next section) are throughput optimal according to Theorem 3.2.

3.7 Review of ORCD

In [21], a stationary priority routing policy called Opportunistic Routing with Congestion Diversity (ORCD) was proposed to improve the delay performance of existing backpressure type schemes. In this section, we briefly review ORCD.

Let $p_{ab} := P(h_{ab}(t) = 1)$ denote the probability that node a is able transmit

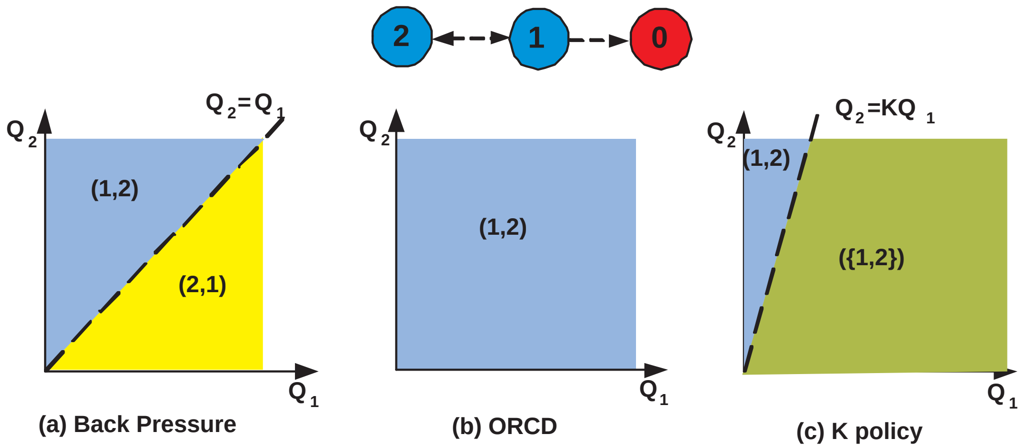

Figure 3.13: A toy example of a three nodes network ($N = 2$)

a packet to node b. Generally, p_{ab} might not be equal to p_{ba}. Assume each node a is able to estimate p_{ab}. And assume $\mathcal{L}$ contains the set of node pair (a, b) with $p_{ab} > 0$, i.e. $\mathcal{L} := \{(a, b) : p_{ab} > 0\}$.

In ORCD policy, nodes are prioritized according to cost function $V_a(t)$ ($a \in \mathcal{N} \setminus C_0$), with higher cost meaning higher priority. $V_a(t)$ is roughly interpreted as the current estimated waiting time of node a to destination C_0. $V_a(t)$ can be calculated for each slot t in a recursive manner. The recursive procedure results in a vector $(V_1(t), .., V_N(t))$ that satisfies the following equation for all $a \in \mathcal{N} \setminus C_0$:

$$V_a(t) = \frac{Q_a(t)}{p(a,t)} + \sum_{b \in \mathcal{U}_a^+(t)} \frac{p(a,b,t)}{p(a,t)} V_b(t) \tag{3.98}$$

where $\mathcal{U}_a^+(t) := \{k | V_b(t) < V_a(t), (a, b) \in \mathcal{L}\}$ is the set of nodes of priority strictly lower than node a at time t. Note that $\mathcal{U}_a^+(t)$ is different from the set of potential forwarder nodes $\mathcal{S}_a(t)$ in that it also includes nodes that are not able to transmit to at time t as long as its cost function is strictly less than $V_a(t)$.

$$p(a,t) := P(\bigcup_{b \in \mathcal{U}_a^+(t)} [h_{ab}(t) = 1]) \tag{3.99}$$

is the probability that at least one node of higher priority that hear a.

$$p(a,b,t) := P([h_{ab}(t) = 1] \cap (\bigcap_{c \in \mathcal{U}_a^+(t), V_c(t) < V_b(t)} [h_{ac}(t) = 0])) \tag{3.100}$$

is the probability that b is the lowest priority node to hear a. It is easy to see that

$$p(a,t) = \sum_{b \in \mathcal{U}_a^+(t)} p(a,b,t) \tag{3.101}$$

ORCD policy requires centralized computation of the costs, making it unsuitable for distributed implementation in an ad-hoc network. The authors in [21] then proposed a decentralized algorithm called distributed ORCD (D-ORCD) to compute the costs. In D-ORCD, each node uses the costs at time slot $t-1$ to update its own cost at time slot t. The following formula is used to update the cost of each node:

$$V_a(t) = \frac{Q_a(t)}{p(a,t-1)} + \sum_{b \in \mathcal{U}_a^+(t-1)} \frac{p(a,b,t-1)}{p(a,t-1)} V_b(t-1) \tag{3.102}$$

3.8 Throughput Optimality of ORCD

In [21], the authors showed that ORCD has better delay performance than backpressure algorithm, as ORCD combines the congestion information with the shortest path calculations inherent in opportunistic routing. The throughput optimality of ORCD was conjectured in [21] but was left unproven. However, with the help of Theorem 3.2 and thanks to the special structure of K policy, we will show in this section that ORCD is indeed throughput optimal. More specially, we will show that ORCD is a refined policy of K-policy.

We begin with a useful lemma before proving our main result.

Lemma 3.7 *For any two nodes, a, b, if $p_{ab} > 0$, then under ORCD,*

$$V_a(t) \leq \frac{Q_a(t)}{p_{ab}} + V_b(t) \tag{3.103}$$

Proof:

If $V_a(t) \leq V_b(t)$ then clearly (3.103) holds. So in what follows, we assume $V_a(t) > V_b(t)$. Since $p_{ab} > 0$, by definition, $b \in \mathcal{U}_a^+(t)$. So the problem becomes to prove that

$$V_a(t) \leq \frac{Q_a(t)}{p_{ab}} + V_b(t) \tag{3.104}$$

for any $b \in \mathcal{U}_a^+(t)$.

Assume $\mathcal{U}_a^+(t) = \{n_1, ..., n_K\}$, $(K \geq 1)$. And without loss of generality, assume $V_{n_1}(t) \leq V_{n_2}(t) \leq, ..., \leq V_{n_K}(t)$. (3.98) can now be rewritten as

$$V_a(t) = \frac{Q_a(t)}{p(a,t)} + \sum_{k=1}^{K} \frac{p(a,n_k,t)}{p(a,t)} V_{n_k}(t) \tag{3.105}$$

where

$$p(a,t) = P(\bigcup_{k=1}^{K} [h_{an_k}(t) = 1]) \tag{3.106}$$

$$p(a,n_k,t) = P([h_{an_k}(t) = 1] \cap (\bigcap_{i=1}^{k-1} [h_{an_i}(t) = 0])) \tag{3.107}$$

Now we focus on a node in $\mathcal{U}_a^+(t)$, say n_j, and we want to show that

$$V_a(t) \leq \frac{Q_a(t)}{p_{an_j}} + V_{n_j}(t) \tag{3.108}$$

Since $V_{n_1}(t) \leq ... \leq V_{n_{j-1}}(t) \leq V_{n_j}(t) \leq V_{n_{j+1}}(t) \leq, ..., \leq V_{n_K}(t)$, (3.105) can be bounded by

$$V_a(t) \leq \frac{Q_a(t)}{p(a,t)} + \frac{\sum_{k=1}^{j} p(a,n_k,t)}{p(a,t)} V_{n_j}(t) + \frac{\sum_{k=j+1}^{K} p(a,n_k,t)}{p(a,t)} V_{n_K}(t) \tag{3.109}$$

For notation concision, we define

$$p_1 := \sum_{k=1}^{j} p(a,n_k,t) \tag{3.110}$$

and

$$p_2 := \sum_{k=j+1}^{K} p(a,n_k,t) \tag{3.111}$$

So we have

$$p(a,t) = p_1 + p_2 \tag{3.112}$$

(3.109) now becomes

$$\begin{aligned} V_a(t) \leq & \frac{Q_a(t)}{p(a,t)} + \frac{p_1}{p(a,t)} V_{n_j}(t) + \frac{p_2}{p(a,t)} V_{n_K}(t) \\ = & \frac{p_1}{p(a,t)} \left(\frac{Q_a(t)}{p_1} + V_{n_j}(t) \right) + \frac{p_2}{p(a,t)} V_{n_K}(t) \end{aligned} \tag{3.113}$$

Note that

$$\frac{p_1}{p(a,t)} + \frac{p_2}{p(a,t)} = 1 \tag{3.114}$$

i.e. $V_a(t)$ can be bounded by a linear combination of $\frac{Q_a(t)}{p_1} + V_{n_j}(t)$ and $V_{n_K}(t)$. But since $V_{n_K}(t) \leq V_a(t)$, we must have

$$V_a(t) \leq \frac{Q_a(t)}{p_1} + V_{n_j}(t) \tag{3.115}$$

Finally, notice that $[h_{an_j}(t) = 1] \subseteq \bigcup_{i=1}^{j} [h_{an_i}(t) = 1]$, so $p_{an_j} = P(h_{an_j}(t) = 1) \leq p_1$. Substitute into (3.115), we conclude

$$V_a(t) \leq \frac{Q_a(t)}{p_{an_j}} + V_{n_j}(t) \tag{3.116}$$

$\square$

Now we are ready to prove the main result of this section:

Theorem 3.3 *ORCD algorithm is throughput optimal.*

Proof:

Let $p_{min} := \min\{p_{ab} | a, b, p_{ab} > 0\}$ and $K = \frac{1}{p_{min}} + 1$. We prove it by showing that ORCD is a refined policy of K policy. More specifically, given ORCD policy $\mathcal{P}_{ORCD}$, for each $P = (C_1, ..., C_{|P|}) \in \mathbb{P}_{\mathcal{L}}$, we are going to show that $\mathcal{P}_{ORCD}(\boldsymbol{Q}(t))$ is a refinement of P as long as $\boldsymbol{Q}(t) \in D_K^{\mathcal{L}}(P)$.

Let us focus on the priority class C_i. Let $C_{<i}$ denote the set of nodes that is in lower priority classes than C_i. i.e. $C_{<i} = \bigcup_{j=1}^{i-1} C_j$.

For any non-isolated node $a \in C_i$, by Lemma 3.1 (2), we have

$$Q_a(t) \geq (K-1)Q_{C_{<i}}(t) \tag{3.117}$$

And it is easy to see the following trivial lower bound of $V_a(t)$ for $a \in C_i$:

$$V_a(t) \geq Q_a(t) \tag{3.118}$$

One the other hand, since P is critical, for each node $b \in C_{<i}$, there exist intermediate nodes $b_1, b_2, ..., b_k \in C_{<i}$ such that $p_{bb_1} > 0$,$p_{b_1b_2} > 0$...,$p_{b_k0} > 0$. By a simple induction using Lemma 3.7, we have the following upper bound of $V_b(t)$:

$$V_b(t) \leq \sum_{m \in C_{<i}} \frac{Q_m(t)}{p_{min}} = \frac{Q_{C_{<i}}(t)}{p_{min}} = (K-1)Q_{C_{<i}}(t) \tag{3.119}$$

Combining (3.117), (3.118) and (3.119), we have

$$V_a(t) \geq Q_a(t) \geq (K-1)Q_{C_{<i}}(t) \geq V_b(t), \quad b \in C_i \tag{3.120}$$

So we conclude that ORCD algorithm will assign all non-isolated nodes in C_i higher priority than nodes in $C_{<i}$.

Now let us focus on the semi-isolated nodes in C_i. Since the semi-isolated nodes in C_i cannot transmit directly to nodes in $C_0, ..., C_{i-1}$, it can only transmit to the non-isolated nodes in C_i or nodes in higher priority class. By (3.98), these nodes will be assigned higher priorities than nodes in $C_{<i}$.

Therefore we conclude that ORCD algorithm will assign all nodes (both non-isolated and semi-isolated) in C_i higher priority than nodes in $C_{<i}$.

By repeating this procedure for C_i, $i = |P|, |P|-1, ..., 2$ in order, we conclude that $\mathcal{P}_{ORCD}(\boldsymbol{Q}(t))$ is a refinement of P as long as $\boldsymbol{Q} \in D_K^{\mathcal{L}}(P)$ and $K = \frac{1}{p_{min}} + 1$. By Theorem 3.2, priority policy $\mathcal{P}_{ORCD}$ is throughput optimal.

□

3.9 Acknowledgement

Chapter 3 is in part a reprint of the material in paper: M. Naghshvar, H. Zhuang, and T. Javidi, "A General Class of Throughput Optimal Routing Policies in Multi-hop Wireless Networks,", *submitted for IEEE Transactions on Information Theory*. The dissertation author was the primary investigator and author of the paper.

Chapter 4

Multi-commodity Case: Extension of K Policy

In Chapter 3, a general routing policy, called K policy, is proposed, which is an extension of backpressure (BP) algorithm and is shown to be throughput optimal. As a major application of K policy, ORCD, which was shown to exhibit better delay performance than BP algorithm in [21], is proved to be a special case of K policy and thus it is throughput optimal. The routing problem considered in Chapter 3 is restricted to single commodity case, where data arriving at one or more nodes are routed to a single destination node or a single set of nodes. However, more common in practice, multiple incoming data flows carrying different commodities sharing the same ad hoc network are routed to different destination nodes. Therefore, it is of interests to extend the single commodity routing algorithm such as K policy to solve the routing problem in a multi-commodity network.

It is well known that backpressure algorithms can be trivially extended to multi-commodity case and preserves its throughput optimality without scarifying it distributed nature [22][33]. Unfortunately, as we will see, a similar extension of K policy based algorithm (such as ORCD) to multi-commodity case would require the a centralized controller with global network knowledge.

Classical backpressure routing algorithm is designed to select the commodity that results in the largest backlog difference. Since single commodity backpressure algorithm can be viewed as a special case of K policy, one might conjecture to use the same commodity selection criterion to extend K policy. However, we will show via an example later that, contrary to the intuition, such a commodity selection policy generally doesn't preserve the throughput optimality when combined with a K policy. Therefore, different commodity selection criteria need to be considered.

Motivated by these difficulties, our goal in the rest of the dissertation is to find multi-commodity routing algorithms with the following properties:

1. It preserves throughput optimality;

2. It can be implemented in a decentralized manner, i.e. it only requires local network information such as the backlog of the neighboring codes;

3. It has good delay performance over existing multi-commodity routing algorithms such as classical backpressure algorithm.

When there is only a single commodity in the network, the routing algorithm can be naturally implemented as an opportunistic routing algorithm. This is because each node doesn't have to choose a commodity before broadcasting its packet. However, in a multi-commodity network, to take advantage of opportunistic routing, each node has to decide which commodity to transmit before its broadcast transmission. Since under the opportunistic routing assumption, the outcome of transmission (i.e. current topology state) is unknown before transmission occurs, the commodity selection criterion cannot depend on current topology state. This further complicates the routing algorithm design.

Nevertheless, in this chapter, we will assume that the current topology state is fully known when making commodity selection and routing decision. We concisely refer to this case as CSI-Tx (channel state information at transmitters). CSI-Tx assumption would allow for a clear presentation of the routing problem and illuminate the main concepts in a simpler form. The resulting algorithm is not opportunistic in nature. Though CSI-Tx assumption results in a larger network capacity (i.e. throughput), it is at the expense of requiring accurate chancel measurement and fast reliable feedback, which would be very demanding in a resource-limited network. In Chapter 5, we will relax the condition to CDI-Tx (channel distribution information at transmitters), where current topology state is unknown before each transmission and the commodity selection decision is made based only on the queue state and the statistics of topology state. The algorithm proposed under CSI-Tx assumption can be naturally modified and implemented as an opportunistic routing algorithm under CDI-Tx assumption. The technique used here is very similar to [41][25], in which the so called DIVBAR algorithm was proposed as an opportunistic version of original backpressure algorithm under CDI-Tx assumption.

By assuming CSI-Tx, we will focus on multi-commodity routing policies with a structure denoted by A-B, where A is a single commodity routing policy such as BP or ORCD, and B is a commodity selection criterion. Suppose there are multiple commodities and the set of commodities is denoted by $\mathcal{K}$, at each time slot t, routing policy A and commodity selection criterion B are integrated by the following steps:

Step 1: For each commodity $c \in \mathcal{K}$, each node a selects the optimal forwarder node according to A routing policy as if c is the only commodity in the network. The selection is denoted by $b_a(c)$.

Step 2: From the outcome of each commodity in step 1), the commodity selection criterion B selects the optimal commodity c_a^* based on certain criterion.

Step 3: Node a forwards a commodity c_a^* packet to node $b_a(c_a^*)$.

We called a multi-commodity algorithm with the above structure as an A-B policy. Note that not all multi-commodity routing policies can be classified as A-B policies. Therefore, A-B policies forms a strict subset among all multi-commodity routing policies. Several reasons for considering only A-B type policy include

- A-B type policy decomposes the problem into two subproblems: single commodity routing and commodity selection, which can potentially reduce the complexity of the problem and makes the derivation more trackable.
- It can take advantage of the existing results on single commodity routing algorithms such as ORCD, which is shown to be throughput optimal and delay efficient.
- As we will see soon, A-B type policy allows for a simple commodity selection criterion, which is usually easy to be implemented in a distributed manner.

Two major commodity selection criteria we will consider in this work are D criterion and R criterion. D criterion is designed to select the commodity with the maximum backlog difference. In contrast, R criterion is designed to select the commodity with the maximum backlog ratio. By the naming rules of A-B policy, classical backpressure algorithm in a multi-commodity network is indeed BP-D policy, i.e. it can be viewed as single commodity backpressure algorithm combined with maximum difference commodity selection criterion.

In this chapter, we first consider a generalized extension of backpressure (BP) algorithm in multi-commodity scenario. To allow for a generalized commodity selection criterion, a non-quadric Lyapunov function construction is introduced. Particularly, we propose BP-R policy as an alterative to the classical backpressure algorithm (aka. BP-D policy).

We then consider the problem of applying similar commodity selection criterion to K policy in a multi-commodity network. We extend K policy by constructing a piecewise non-quadric Lyapunov function. K-R policy is shown to be throughout optimal under mild conditions.

4.1 Generalized Backpressure Algorithm

Before discussing the generalized backpressure algorithm, let us first briefly review the original classical multi-commodity backpressure algorithm, which is well known for its throughput optimality in multi-commodity networks [22][33][36].

4.1.1 Review of Classical Multi-commodity Backpressure Algorithm

Consider the on/off network introduced in Chapter 2. Suppose there are multiple commodities and the set of commodities is denoted by $\mathcal{K}$. In classical multi-commodity backpressure algorithm, the optimal commodity and the optimal forwarder node for node a at time t are determined by the following joint optimization:

$$b_a^*, c_a^* = \arg \max_{b \in \mathcal{S}_a(t), c \in \mathcal{K}} (Q_a^c(t) - Q_b^c(t)) \tag{4.1}$$

where c_a^* and b_a^* are the optimal commodity and optimal forwarder node for node a, respectively. A tie can be broken in an arbitrary manner. Note that if $b_a^* = a$, node a retains the packet without any forwarding. Rewrite (4.1) as a two-level optimizations:

$$b_a^*, c_a^* = \arg \max_{c \in \mathcal{K}} \max_{b \in \mathcal{S}_a(t)} (g(Q_a^c(t)) - g(Q_b^c(t))) \tag{4.2}$$

Notice that the inner optimization is to choose a node in $\mathcal{S}_a(t)$ such that $Q_b^c(t)$ is minimized, i.e.:

$$b_a(c) = \arg \min_{b \in \mathcal{S}_a(t)} (Q_b^c(t)) \tag{4.3}$$

where $b_a(c)$ is the optimal forwarder node to be chosen by the single commodity backpressure algorithm if commodity c is transmitted at time t. The outer optimization of (4.2) becomes

$$c_a^* = \arg \max_{c \in \mathcal{K}} (Q_a^c(t) - Q_{b_a(c)}^c(t)) \tag{4.4}$$

That is, the commodity with the largest backlog difference is selected. We refer to the commodity selection criterion in (4.4) as *D criterion* where D stands for "difference". So classical backpressure algorithm can be viewed as single commodity backpressure (BP) algorithm combined with D criterion. By the naming rule of A-B policy, we refer to classical backpressure algorithm as BP-D policy.

4.1.2 Generalization

Now we generalize (4.1) by introducing a utility function $g(\cdot)$. At time t, consider the following routing criterion for each $a \in \mathcal{N}$

$$b_a^*, c_a^* = \arg \max_{b \in \mathcal{S}_a(t), c \in \mathcal{K}} g(Q_a^c(t)) - g(Q_b^c(t)) \tag{4.5}$$

where utility function $g : \mathbb{R}^+ \mapsto \mathbb{R}^+$ is a function satisfying the follow properties

1) $g(x)$ is non-decreasing;

2) $g(x) \to +\infty$ as $x \to +\infty$;

3) $g(x)$ has bounded derivative (i.e. there exists $g_{sup} > 0$ such that $g'(x) \leq g_{sup}$).

Rewrite (4.5) as a two-level optimizations

$$b_a^*, c_a^* = \arg \max_{c \in \mathcal{K}} \max_{b \in \mathcal{S}_a(t)} (Q_a^c(t) - Q_b^c(t)) \tag{4.6}$$

where the inner optimization is to choose a node in $\mathcal{S}_a(t)$ such that $Q_b^c(t)$ is minimized:

$$b_a(c) = \arg\min_{b\in\mathcal{S}_a(t)} g(Q_b^c(t)) = \arg\min_{b\in\mathcal{S}_a(t)} Q_b^c(t) \tag{4.7}$$

Here $b_a(c)$ is the optimal forwarder node if commodity c is transmitted at time t, which is the same as (4.3) in BP-D due to the non-decreasing property of $g(\cdot)$. In other words, the choice of $g(\cdot)$ won't affect the routing decision made by the single commodity backpressure algorithm. Priorities of nodes are always ranked by their queue lengths regardless of the choice of $g(\cdot)$. Now the outer optimization of (4.6) becomes

$$c_a^* = \arg\max_{c\in\mathcal{K}}(g(Q_a^c(t)) - g(Q_{b_a(c)}^c(t))) \tag{4.8}$$

Hence the generalized backpressure algorithm can be viewed as the single commodity backpressure algorithm combined with commodity selection criterion defined in (4.8). Unlike in the single commodity case where the priority rank is independent of the choice of $g(\cdot)$, when there are multiple commodities, the commodity selection criterion depends on the choice of $g(\cdot)$. Let us look at two examples of $g(\cdot)$:

Special case 1: $g(x) = x$

If $g(x) = x$, the commodity selection criterion becomes

$$c_a^* = \arg\max_{c\in\mathcal{K}}(Q_a^c(t) - Q_{b_a(c)}^c(t)) \tag{4.9}$$

which is exactly D criterion we have just defined. Generalized backpressure becomes classical backpressure algorithm (BP-D).

Special case 2: $g(x) = \log(x + d)$

If $g(x) = \log(x + d)$ where d is a positive constant, then generalized backpressure algorithm becomes:

$$b_a^*, c_a^* = \arg\max_{b,c} \{\log(Q_a^c(t) + d) - \log(Q_b^c(t) + d)\} = \arg\max_{b,c} \frac{Q_a^c(t) + d}{Q_b^c(t) + d} \tag{4.10}$$

Here, positive constant d is applied to avoid infinity when the queue is empty. The resulting algorithm can be viewed as the single commodity backpressure algorithm combined the following commodity selection criterion:

$$c_a^*(t) = \arg\max_c\{\log(Q_a(t)+d) - \log(Q_{b_a(c)}(t)+d)\} = \arg\max_c\{\frac{Q_a^c(t)+d}{Q_{b_a(c)}^c(t)+d}\} \tag{4.11}$$

Roughly speaking, the (4.11) selects the commodity with the largest ratio of backlogs. We refer to the commodity selection criterion in (4.11) as *R criterion*, where R stands for "ratio". And we refer to the policy in (4.10) as BP-R policy.

4.1.3 Throughout Optimality

Theorem 4.1 *Generalized backpressure algorithm is throughput optimal.*

To prove Theorem 4.1, we first need an extension of Lemma 2.1 on Lyapunov stability.

Lemma 4.1 *(Lyapunov Stability) If there exist constants $B > 0$, $\epsilon > 0$, such that for all time slot t we have:*

$$\Delta L(\boldsymbol{Q}(t)) \leq B - \epsilon \sum_c \sum_a g(Q_a^c(t)) \tag{4.12}$$

then the network is stable.

Proof:

Define set C as

$$C := \{\boldsymbol{Q} \in \mathcal{Q} : \sum_c \sum_i g(Q_a^c(t)) \leq \frac{B+1}{\epsilon}, \} \tag{4.13}$$

It is easy to see that C is a finite set. This is because we can rewrite (4.13) as $C = \mathcal{Q} \cap \hat{C}$ where $\hat{C} := \{\boldsymbol{Q} \in \mathbb{R}_+^{N\times|\mathcal{K}|} : \sum_c \sum_i g(Q_a^c(t)) \leq \frac{B+1}{\epsilon}\}$. Since $g(x) \to +\infty$ as $x \to +\infty$, there exists constant K such that $g(x) > \frac{B+1}{\epsilon}$ as long as $x > M$.

Therefore each component of $\boldsymbol{Q}$ restricted in set $\hat{C}$ is bounded, which implies $\hat{C}$ is a compact set. This, together with the fact that $\mathcal{Q}$ is discrete, implies that C is a finite set. Note that for $\boldsymbol{Q}(t) \in \mathcal{Q} \setminus C$, we have

$$\Delta L(\boldsymbol{Q}(t)) \leq B - \epsilon \sum_c \sum_a g(Q_a^c(t)) \leq -1, \tag{4.14}$$

and for $\boldsymbol{Q}(t) \in C$, we have

$$\Delta L(\boldsymbol{Q}(t)) \leq B, \quad x \in C \tag{4.15}$$

So by Lemma 2.3, the Markov process $\{\boldsymbol{Q}(t)\}$ is stable.

□

Let $G(x) := \int_0^x g(v)dv$ denote the integration of g. The following lemma is a extension of Lemma 3.3. It is a direct consequence Taylor's theorem and serves as a basic building block for proving our main results

Lemma 4.2 *If Q^+, Q, μ, A are nonnegative real random variables, and there exist nonnegative real numbers v, c such that $\mu \leq c$,$A \leq c$ and the following relations hold:*

$$Q^+ \leq Q + A \tag{4.16}$$

$$Q^+ \leq Q - \mu + A, \quad \textit{if} \quad Q > v \tag{4.17}$$

then there exists a constant β such that

$$G(Q^+) - G(Q) \leq \beta - g(Q)(\mu - A) \tag{4.18}$$

Proof: If $Q > v$, since g is non-decreasing, from (4.17)

$$G(Q^+) \leq G(Q - \mu + A) \tag{4.19}$$

By Taylor's theorem, we have

$$G(Q - \mu + A) = G(Q) - g(Q)(\mu - A) + \frac{g'(\xi)}{2}(\mu - A)^2 \tag{4.20}$$

where ξ is some real number between Q and $Q - \mu + A$. From (4.19) and (4.20),

$$G(Q^+) - G(Q) \leq \frac{g'(\xi)}{2}(\mu - A)^2 - g(Q)(\mu - A) \tag{4.21}$$

On the other hand, if $Q \leq v$, from (4.16), we have

$$G(Q^+) \leq G(Q + A) \tag{4.22}$$

By Taylor's theorem, we have

$$G(Q + A) = G(Q) + g(Q)A + \frac{g'(\xi)}{2}A^2 \tag{4.23}$$

where ξ is some real number between Q and $Q + A$. From (4.22) and (4.23),

$$G(Q^+) - G(Q) \leq \frac{g'(\xi)}{2}A^2 + g(Q)A = \frac{g'(\xi)}{2}A^2 + g(Q)\mu - g(Q)(\mu - A) \tag{4.24}$$

Now choose $\beta := cg(v) + \frac{c^2 g_{sup}}{2}$, which is a constant and notice that

$$\beta \geq \frac{g'(\xi)}{2}(\mu - A)^2 \tag{4.25}$$

$$\beta \geq \frac{g'(\xi)}{2}A^2 + g(Q)\mu \tag{4.26}$$

The proof is completed by applying (4.25) and (4.26) to (4.21) and (4.24), respectively

□

Proof of Theorem 4.1:

Let $\boldsymbol{Q}(t)$ represent the matrix of queue backlogs, and define the following Lyapunov function:

$$L(\boldsymbol{Q}(t)) = \sum_i \sum_c G_{ic}(Q_i^c(t)) \tag{4.27}$$

Now we are going to calculate the drift $\Delta L(\boldsymbol{Q}(t))$. It is easy to see that the queueing dynamics in (2.2) satisfies the conditions in Lemma 4.2. So there exists a constant β such that for any (i, c),

$$G(Q_i^c(t+1)) - G(Q_i^c(t)) \leq \beta - g(Q_i^c(t))(\mu_{i,out}^c(t) - A_i^c(t) - \mu_{i,in}^c(t)) \tag{4.28}$$

Summing over all (i, c), we have

$$L(\boldsymbol{Q}(t+1)) - L(\boldsymbol{Q}(t)) \leq \beta NM - \sum_{ic} g(Q_i^c(t))(\mu_{i,out}^c(t) - A_i^c(t) - \mu_{i,in}^c(t)) \tag{4.29}$$

Taking the conditional expectations yields the following bounds for Lyapunov drift:

$$\begin{aligned}\Delta L(\boldsymbol{Q}(t)) \leq \beta NM + \sum_{ic} g(Q_i^c(t))\mathbb{E}\{A_i^c(t)|\boldsymbol{Q}(t)\} \\ - \mathbb{E}\left\{\sum_{ic} g(Q_i^c(t))\left(\mu_{i,out}^c(t) - \mu_{i,in}^c(t)\right)|\boldsymbol{Q}(t)\right\}\end{aligned} \tag{4.30}$$

Since arrivals $\boldsymbol{A}(t)$ are i.i.d. over time slots, we have $\mathbb{E}\{A_i^c(t)|\boldsymbol{Q}(t)\} = \lambda_i^c$ for all (i, c). Hence

$$\Delta L(\boldsymbol{Q}(t)) \leq \beta NM + \sum_{ic} g(Q_i^c(t))\lambda_i^c - \mathbb{E}\left\{\sum_{ic} g(Q_i^c(t))\left(\mu_{i,out}^c(t) - \mu_{i,in}^c(t)\right)|\boldsymbol{Q}(t)\right\} \tag{4.31}$$

Note that

$$\begin{aligned}&\sum_{ic} g(Q_i^c(t))\left(\mu_{i,out}^c(t) - \mu_{i,in}^c(t)\right) \\ =&\sum_{ic} g(Q_i^c(t))\left[\sum_b \mu_{ib}^c(t) - \sum_a \mu_{ai}^c(t)\right] \\ =&\sum_{ab}\sum_c \mu_{ab}^c(t)\left[g(Q_a^c(t)) - g(Q_b^c(t))\right]\end{aligned} \tag{4.32}$$

We conclude from (4.32) that general backpressure algorithm is designed to minimize the bound in the right hand side of (4.30) at each time slot over all admissible policies.

However, because $\boldsymbol{\lambda} + \boldsymbol{\epsilon} \in \Lambda$, we know by Theorem 2.4 that there exists a stationary randomized policy that makes decisions based only on the current topology state and independent of the current queue backlog so that all (i, c), we have

$$\mathbb{E}\left\{\sum_{ic} g(Q_i^c(t))\left(\tilde{\mu}_{i,out}^c(t) - \tilde{\mu}_{i,in}^c(t)\right)|\boldsymbol{Q}(t)\right\} = \epsilon + \lambda_i^c \tag{4.33}$$

Applying (4.33) to (4.31), we conclude

$$\Delta L(\boldsymbol{Q}(t)) \leq \beta NM - \epsilon \sum_{ic} g(Q_i^c(t)) \tag{4.34}$$

This drift inequality is in the exact form for application of Lemma 4.1, completing the proof.

□

4.2 K Policy with Multi-commodity

Now we turn to the central question of this chapter: How K policy can be extended to apply in a multi-commodity network? More specifically, we would like to find the answers to these questions:

1) How the Lyapunov function is defined in multi-commodity scenario?

2) What is the commodity selection criteria to adopt with K policy?

From the generalized backpressure algorithm, we have seen that by generalizing the quadratic Lyapunov function to non-quadratic, the routing policy within the same commodity remains the same, but the commodity selection criterion is generalized. We have shown in Chapter 3 that K policy is based on piece-wise quadratic Lyapunov function, which is a quadratic function in each priority cone. These priority cones forms a complete partition of the queue space. It is then natural to extend K policy to multi-commodity scenario by considering piece-wise non-quadratic Lyapunov function.

4.2.1 Construction of the Lyapunov Function

We first consider a single commodity c.

Given a priority list $P^c = (C_1^c, C_2^c, ..., C_{|P^c|}^c)$, the Lyapunov function under list P^c and parameter K is defined as

$$\begin{aligned} L_K(\boldsymbol{Q}^c; P^c) :=& \sum_{i=1}^{|P^c|} \int_0^{Q^c_{C_i^c}} g\left(W_K(|C^c_{<i}|, |C_i^c|)x\right) dx \\ =& \sum_{i=1}^{|P^c|} \frac{1}{W_K(|C^c_{<i}|, |C_i^c|)} G\left(W_K(|C^c_{<i}|, |C_i^c|) Q^c_{C_i^c}\right) \end{aligned} \tag{4.35}$$

where $W_K(n,m)$ is defined in (3.8) and $C^c_{<i}$ denote the set of nodes that has lower priority than nodes in the i-th priority class, i.e.

$$C^c_{<i} := \bigcup_{j=1}^{i-1} C_j^c \tag{4.36}$$

Proposition 4.1 *For any adjacent list pair P and P', $L_K(\boldsymbol{Q}; P)$ and $L_K(\boldsymbol{Q}; P')$ are equal and have equal gradient on separation hyperplane $H_K(P; P')$ defined in* (3.12)

Proof:

Given an adjacent list pair, $P = \{C_1, C_2, ..., C_{|P|}\}$ and $P' = (C_1', C_2', ..., C'_{|P'|})$, without loss of generality, assume P' is a one-step refinement of P with regard to priority class $C_i (1 \le i \le |P|)$, then $|P'| = |P| + 1$ and

$$\begin{cases} C_k = C_k' & \text{if } 1 \le k \le i-1 \\ C_i = C_i' \cup C_{i+1}' & \\ C_k = C'_{k+1} & \text{if } i+1 \le k \le |P| \end{cases} \tag{4.37}$$

On the hyperplane that separates the adjacent list pair P and P', we have by definition

$$W_K(|C'_{<i}|, |C_i'|) Q_{C_i'} = W_K(|C'_{<i+1}|, |C'_{i+1}|) Q_{C'_{i+1}} \tag{4.38}$$

Rewrite (4.38) as

$$\frac{1}{W_K(|C'_{<i+1}|,|C'_{i+1}|)}(Q_{C'_i}+Q_{C'_{i+1}}) = \frac{1}{W_K(|C'_{<i}|,|C'_i|)}Q_{C'_{i+1}}+\frac{1}{W_K(|C'_{<i+1}|,|C'_{i+1}|)}Q_{C'_{i+1}} \tag{4.39}$$

By using Proposition (3.5), we have

$$\frac{1}{W_K(|C'_{<i+1}|,|C'_{i+1}|)}(Q_{C'_i}+Q_{C'_{i+1}}) = \frac{1}{W_K(|C'_{<i}|,|C'_i|+|C'_{i+1}|)}Q_{C'_{i+1}} \tag{4.40}$$

By using (4.37) and after proper arrangement, we have

$$W_K(|C_{<i}|,|C_i|)Q_{C_i} = W_K(|C'_{<i}|+|C'_i|,|C'_{i+1}|)Q_{C'_{i+1}} \tag{4.41}$$

This, together with (4.38), gives

$$W_K(|C_{<i}|,|C_i|)Q_{C_i} = W_K(|C'_{<i+1}|,|C'_{i+1}|)Q_{C'_{i+1}} = W_K(|C'_{<i}|,|C'_i|)Q_{C'_i} \tag{4.42}$$

So on the hyperplane $H_K(P;P')$ that separates the adjacent list pair P and P', we have

$$g(W_K(|C_{<i}|,|C_i|)Q_{C_i}) = g(W_K(|C'_{<i+1}|,|C'_{i+1}|)Q_{C'_{i+1}}) = g(W_K(|C'_{<i}|,|C'_i|)Q_{C'_i}) \tag{4.43}$$

For $a \in C_i$, we have

$$\frac{\partial L_K(\boldsymbol{Q};P)}{\partial Q_a} = g\left(W_K(|C_{<i}|,|C_i|)Q_{C_i}\right) \tag{4.44}$$

By (4.44), for $a \in C'_i$, on $H_K(P;P')$, we have

$$\frac{\partial L_K(\boldsymbol{Q};P')}{\partial Q_a} = g(W_K(|C'_{<i}|,|C'_i|)Q'_{C_i}) = g(W_K(|C_{<i}|,|C_i|)Q_{C_i}) = \frac{\partial L_K(\boldsymbol{Q};P)}{\partial Q_a} \tag{4.45}$$

Similarly, for $a \in C'_{i+1}$, on $H_K(P;P')$, we have

$$\frac{\partial L_K(\boldsymbol{Q};P')}{\partial Q_a} = g(W_K(|C'_{<i+1}|,|C'_{i+1}|)Q_{C'_{i+1}}) = g(W_K(|C_{<i}|,|C_i|)Q_{C_i}) = \frac{\partial L_K(\boldsymbol{Q};P)}{\partial Q_a} \tag{4.46}$$

This proves that they have equal gradient on separation hyperplane $H_K(P;P')$.

Now we are going to show that on $H_K(P;P')$,

$$L_K(\boldsymbol{Q};P') = L_K(\boldsymbol{Q};P) \tag{4.47}$$

That is, we need to show that

$$\begin{aligned}\int_0^{Q_{C'_i}} g\left(W_K(|C'_{<i}|,|C'_i|)x\right)dx + \int_0^{Q_{C'_{i+1}}} g\left(W_K(|C'_{<i+1}|,|C'_{i+1}|)x\right)dx \\ = \int_0^{Q_{C_i}} g\left(W_K(|C_{<i}|,|C_i|)x\right)dx\end{aligned} \tag{4.48}$$

Notice that

$$\begin{aligned} LHS =& \frac{1}{W_K(|C'_{<i}|,|C'_i|)}\int_0^{W_K(|C'_{<i}|,|C'_i|)Q_{C'_i}} g(x)dx \\ &+ \frac{1}{W_K(|C'_{<i+1}|,|C'_{i+1}|)}\int_0^{W_K(|C'_{<i+1}|,|C'_{i+1}|)Q_{C'_{i+1}}} g(x)dx \\ =& \left(\frac{1}{W_K(|C'_{<i}|,|C'_i|)} + \frac{1}{W_K(|C'_{<i+1}|,|C'_{i+1}|)}\right)\int_0^{W_K(|C_{<i}|,|C_i|)Q_{C_i}} g(x)dx \\ =& \frac{1}{W_K(|C_{<i}|,|C_i|)}\int_0^{W_K(|C_{<i}|,|C_i|)Q_{C_i}} g(x)dx \\ =& RHS \end{aligned} \tag{4.49}$$

This completes the proof.

□

For multi-commodity case, the joint queue state $\boldsymbol{Q}(t)$ is a collection of queue state vectors, with $\boldsymbol{Q}^c(t)$ corresponds to the queue state of commodity c, i.e. $\boldsymbol{Q}(t) := \{\boldsymbol{Q}^c(t)\}_{c\in\mathcal{K}}$. Let $\mathbb{P}^c_{\mathcal{L}}$ denote the set of critical priority lists of commodity c. Then the joint critical priority list set is defined as $\mathbb{P}_{\mathcal{L}} := \{\mathbb{P}^c_{\mathcal{L}}\}_{c\in\mathcal{K}}$.

Now, we are ready to write the Lyapunov function given a priority list set $\mathbb{P}_{\mathcal{L}}$:

$$L_K(\boldsymbol{Q};\mathbb{P}_{\mathcal{L}}) := \sum_{c\in\mathcal{K}}\sum_{P\in\mathbb{P}^c_{\mathcal{L}}} L_K(\boldsymbol{Q}^c;P)\mathbb{I}\{\boldsymbol{Q}^c \in D^{\mathcal{L}}_K(P)\} \tag{4.50}$$

where $\mathbb{I}\{\boldsymbol{Q}^c \in D_K^{\mathcal{L}}(P)\}$ is the indicator function which equals 1 when $\boldsymbol{Q}^c \in D_K^{\mathcal{L}}(P)$ and 0 otherwise.

Proposition 4.2 $L_K(\boldsymbol{Q}; \mathbb{P}_{\mathcal{L}})$ *is a continuous function with continuous gradient (aka. derivative).*

Proof: Follow from Proposition 4.1.

□

4.2.2 K-R Policy and Its Throughput Optimality

Similar to the single commodity case, given parameter K, we define the multi-commodity version of f_K as follows: Given queue state $\boldsymbol{Q}$, if for each commodity c, $\boldsymbol{Q}^c \in D_K^{\mathcal{L}}(P^c)$, where $P^c = (C_1^c, C_2^c, ..., C_{|P|}^c)$, then

$$f_K(\boldsymbol{Q}, \boldsymbol{\mu}) := \sum_{c=1} \sum_{i=1}^{|P^c|} g(W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c})(\mu_{C_i^c, out} - \mu_{C_i^c, in}) \tag{4.51}$$

Like in the single commodity case, we define $U_{C_i}^c$ as follows

$$U_{C_i}^c := \begin{cases} W_K(|C_{<i}|, |C_i|)Q_{C_i}^c & \text{if } 1 \leq i \leq |P| \\ 0 & \text{if } i = 0 \end{cases} \tag{4.52}$$

We also use notation $[a]$ to denote the priority class node a belongs to. i.e. we have $U_{[a]}^c = U_{C_i}^c$ if $a \in C_i$.

From the discussion of single commodity case, it is easy to see that any policy that maximize $f_K(\boldsymbol{Q}, \boldsymbol{\mu})$ in (4.51) would be throughput optimal. However, unlike the single commodity case, maximizing $f_K(\boldsymbol{Q}, \boldsymbol{\mu})$ under multi-commodity case requires global network information, more specifically, it requires the knowledge of current classes of current priority list.

To see why this is the case, we have

$$\begin{aligned} & f_K(\boldsymbol{Q}, \boldsymbol{\mu}) \\ &= \sum_{c=1} \sum_{i=1}^{|P^c|} g(W_K(|C^c_{<i}|, |C^c_i|) Q_{C^c_i})(\mu_{C^c_i,out} - \mu_{C^c_i,in}) \\ &= \sum_{a\in\mathcal{N}} \sum_{b\in\mathcal{N}} \sum_{c\in\mathcal{K}} \mu^c_{ab} \left(U^c_{[a]} - U^c_{[b]}\right) \end{aligned} \tag{4.53}$$

So

$$\begin{aligned} & \max_{\mu\in\mathcal{M}(\boldsymbol{H})} f_K(\boldsymbol{Q}, \boldsymbol{\mu}) \\ &= \max_{\mu\in\mathcal{M}(\boldsymbol{H})} \sum_{a\in\mathcal{N}} \sum_{b\in\mathcal{N}} \sum_{c\in\mathcal{K}} \mu^c_{ab} \left(U^c_{[a]} - U^c_{[b]}\right) \\ &= \sum_{a\in\mathcal{N}} \max_{b\in\mathcal{S}_a(t), c\in\mathcal{K}} \left(U^c_{[a]} - U^c_{[b]}\right) \\ &= \sum_{a\in\mathcal{N}} \left\{ \max_{c\in\mathcal{K}} \max_{b\in\mathcal{S}_a(t)} \left(U^c_{[a]} - U^c_{[b]}\right) \right\} \\ &= \sum_{a\in\mathcal{N}} \max_{c\in\mathcal{K}} \left\{ U^c_{[a]} - \min_{b\in\mathcal{S}_a(t)} U^c_{[b]} \right\} \end{aligned} \tag{4.54}$$

From (4.54), it can be seen that to maximize $f_K(\boldsymbol{Q}, \boldsymbol{\mu})$, a joint optimization over commodity $c \in \mathcal{K}$ and destination $b \in \mathcal{S}_a(t)$ is required for each node a. Such a joint optimization can be divided into two-level of optimizations. The inner optimization is to choose destination node b such that $U^c_{[b]}$ is minimized for given c. This is exactly, what K policy (such as ORCD) would do. Let $b^*_a(c)$ denote the optimal node K policy would choose if commodity c is chosen to transmit from node a (If there are multiple optimal nodes, $b^*_a(c)$ can be arbitrarily chosen). By substituting b by $b^*_a(c)$, we removed the inner optimization. Now (4.54) becomes

$$\begin{aligned} & \max_{\boldsymbol{\mu}\in\mathcal{M}(\boldsymbol{H})} f_K(\boldsymbol{Q}, \boldsymbol{\mu}) \\ &= \sum_{a\in\mathcal{N}} \max_{c\in\mathcal{K}} \left\{ U^c_{[a]} - U^c_{[b^*_a(c)]} \right\} \end{aligned} \tag{4.55}$$

That is, each node chooses an optimal commodity c such that $U^c_{[a]} - U^c_{[b^*_a(c)]}$ is maximized. However, note that the value of $U^c_{[a]}$ is generally different for different commodity c and the value requires the knowledge of the priority class that contains a, which is not known. Though technically, it is possible to calculate the value given sufficient global information of the network, it is difficult to realize in a practical wireless network.

Therefore, alternative approaches need to be considered. First of all, it is not difficult to see that maximizing $f_K(\boldsymbol{Q}, \boldsymbol{\mu})$ is not a necessary condition for throughput optimality. The following Lemma provides a relaxed condition for the throughput optimality.

Lemma 4.3 *If the routing policy $\pi^* \in \Pi$ is chosen such that for each $\boldsymbol{Q}$*

$$\mathbb{E}_{\boldsymbol{H}}\left\{f_K(\boldsymbol{Q}, \pi^*(\boldsymbol{H}, \boldsymbol{Q}))\right\} \geq \gamma \max_{\pi \in \Pi} \mathbb{E}_{\boldsymbol{H}}\left\{f_K(\boldsymbol{Q}, \pi(\boldsymbol{H}, \boldsymbol{Q}))\right\} - D \tag{4.56}$$

for some constant γ $(0 \leq \gamma \leq 1$), D $(0 \leq D < \infty)$, where the expectation is taken over the statistics of $\boldsymbol{H}(t)$, then the network is stable provided that the arrival rates are interior to $\gamma\Lambda$, which is a γ-scaled version of the capacity region Λ.

Proof: See Appendix A.3.

□

With Lemma 4.3 in mind, we are trying to find a simple commodity selection criterion that might not maximize $f_K(\boldsymbol{Q}(t), \boldsymbol{\mu}(t))$, but guarantees its gap from the maximum is bounded.

The following lemma shows that if we choose $g(x) := \log(x + d)$, then the error introduced by using $g(Q_a^c)$ as an estimate of $U_{[a]}^c$ is bounded.

Lemma 4.4 *If $g(x) := \log(x+d)$, and node $a \in \mathcal{N}$ is non-isolated under P^c, then*

$$|g(Q_a^c) - U_{[a]}^c| \leq D \tag{4.57}$$

for some positive constant D.

Proof: Assume $a \in C_i^c$, and recall that

$$U_{C_i}^c := \begin{cases} g(W_K(|C_{<i}|, |C_i|)Q_{C_i}^c) & \text{if } 1 \le i \le |P| \\ 0 & \text{if } i = 0 \end{cases} \tag{4.58}$$

Then, (4.57) can be written as

$$|\log(Q_a^c + d) - \log(W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c}^c + d)| \le D \tag{4.59}$$

Since $a \in C_i^c$ is non-isolated, by Lemma 3.1,

$$Q_a^c \ge \frac{K-1}{K^{|C_i^c|} - 1} Q_{C_i^c}^c \tag{4.60}$$

Then

$$\begin{aligned} Q_a^c &\ge \frac{K-1}{(K^{|C_i^c|} - 1)W_K(|C_{<i}^c|, |C_i^c|)} W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c}^c \\ &= \frac{K-1}{K^{|C_{<i}^c|}} W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c}^c \\ &\ge \frac{K-1}{K^N} W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c}^c \\ &\ge \frac{1}{K^N} W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c}^c \end{aligned} \tag{4.61}$$

So

$$Q_a^c + d \ge \frac{1}{K^N} W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c}^c + d \ge \frac{1}{K^N}\left(W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c}^c + d\right) \tag{4.62}$$

Taking logarithm on both sides, we have

$$\log(Q_a^c + d) \ge -N\log K + \log(W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c}^c + d) \tag{4.63}$$

On the other hand,

$$Q_a^c \le Q_{C_i^c}^c \tag{4.64}$$

Therefore

$$\begin{aligned} Q_a^c &\le \frac{1}{W_K(|C_{<i}^c|, |C_i^c|} W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c}^c \\ &= K^{|C_{<i}^c|}(K^{|C_i^c|} - 1)W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c}^c \\ &\le K^N W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c}^c \end{aligned} \tag{4.65}$$

So

$$Q_a^c + d \le K^N W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c}^c + d \le K^N \left(W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c}^c + d \right) \quad (4.66)$$

Taking logarithm on both side, we have

$$\log(Q_a^c + d) \le N \log K + \log(W_K(|C_{<i}^c|, |C_i^c|)Q_{C_i^c}^c + d) \quad (4.67)$$

Combine (4.63) and (4.67), and choose $D = N \log K$, we conclude

$$|g(Q_a^c) - U_{[a]}^c| \le D \quad (4.68)$$

This completes the proof.

□

Lemma 4.4 implies that using $\log(Q_a^c + d)$ as an estimate of $U_{[a]}^c$ can guarantee a bounded error, which could in turn suggest throughout optimality according to Lemma 4.3. The commodity selection criterion in form of (4.8) with $g(x) := \log(x + d)$ is exactly R criterion. We refer to K policy combined with R criterion as K-R policy. Note that Lemma 4.4 requires the node to be non-isolated, which is always the case when K policy is designed based on $\mathbb{P}_{all}$. Therefore, we have the following main result of this chapter.

Theorem 4.2 *K-R policy based on $\mathbb{P}_{all}$ is throughput optimal.*

Proof:

By Lemma 4.3, it is sufficient to show

$$\left\{ U_{[a]}^{c_a^*} - U_{[b_a^*(c_a^*)]}^{c_a^*} \right\} \ge \max_{c \in \mathcal{K}} \left\{ U_{[a]}^c - U_{[b_a^*(c)]}^c \right\} - D \quad (4.69)$$

for some constant D $(0 \le D < \infty)$, where c_a^* is the commodity chosen under R criterion, that is

$$c_a^* = \arg\max_c \frac{Q_a^c + d}{Q_{b_a^*(c)+d}^c} \quad (4.70)$$

Let c'_a denote the commodity that maximizes $\left\{U^c_{[a]} - U^c_{[b^*_a(c)]}\right\}$. Then we are going to prove

$$\left\{U^{c^*_a}_{[a]} - U^{c^*_a}_{[b^*_a(c^*_a)]}\right\} \geq \left\{U^{c'_a}_{[a]} - U^{c'_a}_{[b^*_a(c'_a)]}\right\} - D \tag{4.71}$$

Since $g(x) := \log(x + d)$, it is easy to rewrite (4.70) as

$$c^*_a = \arg\max_c \{g(Q^c_a) - g(Q^c_{b^*_a(c)})\} \tag{4.72}$$

Then by definition we have

$$g(Q^{c^*_a}_a) - g(Q^{c^*_a}_{b^*_a(c^*_a)}) \geq g(Q^{c'_a}_a) - g(Q^{c'_a}_{b'_a(c'_a)}) \tag{4.73}$$

By comparing (4.71) with (4.73), it is sufficient to prove that for any node a and commodity c

$$|g(Q^c_a) - U^c_{[a]}| \leq D/4 \tag{4.74}$$

which follows directly from Lemma 4.4, since all nodes are non-isolated when the set of priority lists is given by $\mathbb{P}_{all}$.

□

4.3 Acknowledgement

Chapter 4 is in part a reprint of the material in paper: H. Zhuang and R. Cruz, "Throughput Optimal Routing in Multi-commodity Wireless Ad-hoc Networks", *in preparation*. The dissertation author was the primary investigator and author of the paper.

Chapter 5

Application of K-R Policy and Performance Comparison

5.1 ORCD-R

In Chapter 3, we have proved the throughput optimality of ORCD as an application of K policy. In Chapter 4, the extension of K policy called K-R policy is proved to be throughput optimal in a multi-commodity network under CSI-Tx assumption. Since ORCD is a special case of K policy, it is naturally to extend ORCD to multi-commodity scenario by adding R criterion as commodity selection criterion.

Recall that we refer to the backpressure algorithm with R criterion as BP-R policy. In a similar way, we refer to ORCD algorithm with R criterion as ORCD-R policy. Since both BP and ORCD are special cases of K policy, it is clear by Theorem 4.2, both BP-R and ORCD-R are throughput optimal under CSI-Tx assumption.

We have so far discussed BP-D, BP-R, and ORCD-R, it is then naturally to have ORCD-D policy, which is ORCD algorithm combined with D criterion.

Unlike the other policies, there is no theoretical results to ensure the throughput optimality of ORCD-D. Indeed, in this chapter, we will see via simulations, that under certain scenario, ORCD-D is not stable even the input rate is within the capacity region of the network.

The following table is a summary of the routing schemes under CSI-Tx assumption:

Routing scheme	Throughput optimality	Underlying Lyapunov function
BP-D (Classical back-pressure)	Yes	Quadric function
BP-R	Yes	Log-quadric function
ORCD (Single commodity)	Yes	Piecewise quadric function
ORCD-R	Yes	Piecewise Log-quadric function
ORCD-D	No	–

5.2 Opportunistic Routing under CDI-Tx Assumption

So far we have assumed CSI-Tx and the routing control decision is made based on current topology state and queue state. To obtain CSI (i.e. topology state), a channel measurement is required for each node. More specifically, before forwarding the actual data packet at each time slot, some probing data (e.g. pilot symbols) are first broadcasted to its neighboring nodes and these nodes measure the pilots symbols to see if the link is reliable and send the information back to the sender via a feedback channel. There are several major drawbacks of this mechanism. First of all, overhead needs to be added for channel measurement, which effectively reduces the actual data throughput. Secondly, since the actual transmission takes place after the channel measurement, measurement error is inevitable due to the time varying nature of wireless channels. Furthermore, the feedback chancel could also introduce errors. These errors would further reduce the effective throughput. Lastly, the complexity of implementing such routing mechanism is very high.

A more relaxed and practical assumption is to assume CDI-Tx, in which case the transmitters don't know the chancel state information but have certain knowledge of the statistics of the topology state (i.e. channel). In recent years, the concept of opportunistic routing has attracted great research interests for its ability to mitigate the impact of poor wireless links by exploiting the broadcast nature wireless transmissions and the path diversity. In an opportunistic routing algorithm, the sender of each node identifies a set of potential forwarders and multicasts the message to them. The successful recipients respond with ACKs. The sender then identifies the best among these receivers, according to some predefined criteria, and sends a forwarding order to it. Hence the actual routing decision is made after each transmission. As opposed to wired network where connectivity is limited by physical wiring, wireless networks have an inherently broadcast nature. Overhearing a message by those other than the intended recipient was considered as interference and detrimental to the network performance. Opportunistic routing turns this belief around and uses this receiver diversity to boost the network throughput. As we can see, one major advantage of opportunistic routing is that CSI is not required at transmitters. Therefore, opportunistic routing is a very suitable choice for networks under CDI-Tx assumption.

In [41][25], diversity backpressure routing (DIVBAR) is proposed as an opportunistic version of backpressure algorithm under CDI-Tx assumption, which incorporates the wireless local transmission diversity. Like backpressure algorithm, DIVBAR uses backpressure to learn efficient routes, where incoming data "pushes" old data in directions of least resistance. It is known that DIVBAR ensures bounded expected total backlog for all stabilizable arrival. However, DIVBAR suffers from poor delay performance in a lightly loaded network as the packets may be routed in in appropriate directions before enough backlog builds up to suggest alternative routs. In [20], a follow-up work to DIVBAR proposes a heuristic enhancement , known as E-DIVBAR, which uses a sum of the queues and expected number of transmissions as the new differential backlog metric.

ORCD in a single commodity network can be naturally implemented in an opportunistic manner under CDI-Tx assumption. The theoretical analysis and capacity region of single commodity ORCD in CDI-Tx case are exactly the same as those in CSI-Tx case since there is no commodity selection procedure before each transmission. It is shown in [21] that ORCD outperforms both DIVBAR and E-DIVBAR in terms of delay performance in a single commodity network under CDI-Tx assumption.

When there are multiple commodities, an opportunistic routing algorithm has to select commodity before each transmission. Since under CDI-Tx assumption, current topology state is not known at transmitters, the commodity can only be selected based on the queue state and the statistic of the network topology. Let $\mathcal{C}(\boldsymbol{Q})$ denote the commodity selection policy, which is a (potentially randomized) function of queue state only and taking values as the commodity selection for each node. Given a commodity selection for each node $\boldsymbol{c} = (c_a)_{a\in\mathcal{N}}$, where c_a is the commodity selection of node a, let $\mathcal{M}(\boldsymbol{H}, \boldsymbol{c})$ denote the set of all admissible routing control actions when the commodity selection for each node is given by $\boldsymbol{c}$. i.e. $\mathcal{M}(\boldsymbol{H}, \boldsymbol{c})$ is the set of routing control actions $\{\boldsymbol{\mu}\}$ satisfying

$$\mu_{ab}^{c_a} \leq h_{ab} \tag{5.1}$$

$$\mu_{ab}^{c} = 0, \quad c \neq c_a \tag{5.2}$$

and

$$\sum_{b} \mu_{ab}^{c_a} = 1 \tag{5.3}$$

Then a routing policy (potentially randomized) is called admissible under CDI-Tx assumption if there exists a commodity selection policy $\mathcal{C}$ such that for all $\boldsymbol{H} \in \mathcal{H}$, $\boldsymbol{Q} \in \mathcal{Q}$

$$\pi(\boldsymbol{H}, \boldsymbol{Q}) \in \mathcal{M}(\boldsymbol{H}, \mathcal{C}(\boldsymbol{Q})) \tag{5.4}$$

Let Π_{CDI} denote the set of all admissible routing policies under CDI-Tx assumption. Since Π_{CDI} is a strict subset of Π , the capacity region under CDI-Tx assumption is generally a subset of that under CSI-Tx assumption.

In previous chapters, we have proposed BP-R and ORCD-R as improved routing schemes over original backpressure algorithm (BP-D) for multi-commodity network under CSI-Tx assumption. BP-R and ORCD-R cannot be implemented under CDI-Tx assumption since they select commodity based on current topology. However, by adopting a similar technique used for DIVBAR in [41][25], we can modify BP-R and ORCD-R to opportunistic routing algorithms under CDI-Tx assumption.

We refer to the modified BP-R under CDI-Tx assumption as DIVBAR-R [1]. The algorithm is described as follows:

DIVBAR-R

Define $\mathcal{Z}_a^c(t) := \{b | Q_b^c(t) < Q_a^c(t), (a,b) \in \mathcal{L}\}$

The following algorithm is applied for each node a:

1) For each commodity c, the receivers $b \in \mathcal{Z}_a^c(t)$ are priority ranked according to $Q_b^c(t)$, so that the receivers with larger values are ordered with higher priority (breaking ties arbitrarily). We define $b(a,c,t,k)$ as the node $b \in \mathcal{S}_a(t)$ with k-th largest value $V_b^c(t)$ for commodity c. Thus by definition we have:

$$Q^c_{a,b(a,c,t,1)} \geq Q^c_{a,b(a,c,t,2)} \geq Q^c_{a,b(a,c,t,3)} \cdots \tag{5.5}$$

2) Define $\phi_{ab}^c(t)$ as the probability that a packet transmission from node a

[1]Note that despite of its name, DIVBAR-R is not an A-B type policy by definition

is successfully received by node b, but is not received by any other nodes in $\mathcal{Z}_a^c(t)$ that are ranked with lower priority than node b according to the commodity c rank ordering of the previous step.

3) Select commodity $c_a^*(t)$ as the commodity $c \in \mathcal{Z}^c(t)$ that maximizes (breaking ties arbitrarily):

$$\sum_{k=1}^{|\mathcal{Z}_a(t)|} \log\left(\frac{Q_a^c(t)+d}{Q_{b(a,c,t,k)}^c(t)+d}\right)\phi_{a,b(a,c,t,k)}^c(t) \tag{5.6}$$

where $|\mathcal{Z}_a^c(t)|$ denote the number of nodes in the set $\mathcal{Z}_a^c(t)$. Node a transmits a packet of commodity $c_a^*(t)$ for time slot t.

4) After receiving ACK/NACK feedback about the successful recipients of the transmission, node a shifts responsibility of the packet forwarding to the successful receiver b with the smallest $Q_b^c(t)$. If the node with smallest $Q_b^c(t)$ is a itself, node a retains the responsibility of the packet.

□

It is easy to show that DIVBAR-R maximizes $\mathbb{E}_{\boldsymbol{H}}\left\{f_K(\boldsymbol{Q}, \pi(\boldsymbol{H}, \boldsymbol{Q}))\right\}$ over all admissible policies in Π_{CDI} and thus it is throughput optimal under CDI-Tx assumption by Lemma 4.3.

The modified ORCD-R under CDI-Tx assumption (called Opportunistic ORCD-R) [2]) is described as follows:

Opportunistic ORCD-R

Recall that $\mathcal{U}_a^c(t) := \{b | V_b^c(t) < V_a^c(t), (a,b) \in \mathcal{L}\}$

The following algorithm is applied for each node a:

1) For each commodity c, the receivers $b \in \mathcal{U}_a^c(t)$ are priority ranked according to $V_b^c(t)$, so that the receivers with larger values are ordered with higher

[2]Note that thought letter "O" in ORCD-R stands for opportunistic, ORCD-R described in Chapter 5 is not an opportunistic routing algorithm in nature.

priority (breaking ties arbitrarily). We define $b(a, c, t, k)$ as the node $b \in \mathcal{U}_a^c(t)$ with k-th largest value $V_b^c(t)$ for commodity c. Thus by definition we have:

$$V^c_{a,b(a,c,t,1)} \geq V^c_{a,b(a,c,t,2)} \geq V^c_{a,b(a,c,t,3)} \cdots \tag{5.7}$$

2) Define $\phi^c_{ab}(t)$ as the probability that a packet transmission from node a is successfully received by node b, but is not received by any other nodes in $\mathcal{U}_a^c(t)$ that are ranked with lower priority than node b according to the commodity c rank ordering of the previous step.

3) Select commodity $c_a^*(t)$ as the commodity $c \in \mathcal{K}$ that maximizes (breaking ties arbitrarily):

$$\sum_{k=1}^{|\mathcal{U}_a^c(t)|} \left(g(Q_a^c(t)) - g(Q^c_{b(a,c,t,k)}(t))\right) \phi^c_{a,b(a,c,t,k)}(t) \tag{5.8}$$

where $|\mathcal{U}_a^c(t)|$ denote the number of nodes in the set $\mathcal{U}_a^c(t)$, and $g(x) = \log(x + d)$ for some positive constant d. Node a transmits a packet of commodity $c_a^*(t)$ for time slot t.

4) After receiving ACK/NACK feedback about the successful recipients of the transmission, node a shifts responsibility of the packet forwarding to the successful receiver b with the smallest $V_b^c(t)$. If the node with smallest $V_b^c(t)$ is a itself, node a retains the responsibility of the packet.

□

Similar to DIVBAR which inherits its throughput optimality from backpressure algorithm, the opportunistic ORCD-R inherits its throughput optimality from ORCD-R, which is stated in the following theorem:

Theorem 5.1 *Opportunistic ORCD-R is throughput optimal among all routing policies under CDI-Tx assumption.*

Proof:

Let $\pi^* \in \Pi_{CDI}$ denote the opportunistic ORCD-R policy. By Lemma 4.3, it is sufficient to show that

$$\mathbb{E}_{\boldsymbol{H}(t)}\left\{f_K(\boldsymbol{Q}(t), \pi^*(\boldsymbol{H}(t), \boldsymbol{Q}(t)))\right\} \geq \max_{\pi \in \Pi_{CDI}} \mathbb{E}_{\boldsymbol{H}(t)}\left\{f_K(\boldsymbol{Q}(t), \pi(\boldsymbol{H}(t), \boldsymbol{Q}(t)))\right\} - D \tag{5.9}$$

for some constant D. Note that the optimization on the right hand side is over the set of all admissible policies Π_{CDI} defined in (5.4).

It is then sufficient to show that for each node a

$$\begin{aligned} &\sum_{k=1}^{|\mathcal{U}_a^c(t)|} \left(g(Q_a^c(t)) - g(Q_{b(a,c,t,k)}^c(t))\right) \phi_{a,b(a,c,t,k)}^c(t) \\ \geq &\sum_{k=1}^{|\mathcal{U}_a^c(t)|} \left(U_{[a]}^c(t) - U_{[b(a,c,t,k)]}^c(t)\right) \phi_{a,b(a,c,t,k)}^c(t) - D_2 \end{aligned} \tag{5.10}$$

for some constant D_2, where $g(x) = \log(x + d)$.

(5.10) holds due to Lemma 4.4, i.e. there exists D_3 such that for any a,

$$|g(Q_a^c) - U_{[a]}^c| \leq D_3 \tag{5.11}$$

□

5.3 Numerical Results: CSI-Tx Case

Consider the 2-commodity network shown in Figure 5.1. All links are independent with probability of successful reception denoted on the graph. Commodity data 1 with arrival rate λ_1 is input from node 3 with destination node 1; Commodity data 2 with arrival rate λ_2 is input from node 4 with destination node 2;

We first consider CSI-Tx case. The routing schemes of interests under CSI-Tx assumption are BP-D, BP-R, ORCD-D and ORCD-R. Figure 5.2 and 5.3

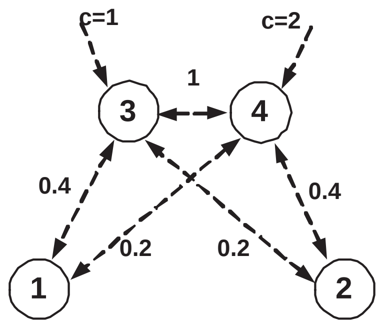

Figure 5.1: A 4 nodes network with 2 commodities. $p_{13} = p_{31} = 1$, $p_{24} = p_{24} = p_{13} = p_{31} = 0.4$, $p_{23} = p_{32} = p_{14} = p_{41} = 0.2$.

show the simulation results of these schemes under the 4 nodes network in Figure 5.1. It is easy to see that the network capacity of each commodity is bounded by $C_{max} = 0.4 + 0.2 - 0.2 \cdot 0.4$. In this simulation, we constraint our arrival rate to satisfy $\lambda_1 = \lambda_2 = \rho C_{max}$, where ρ is the traffic load, which is a real number between 0 and 1. For an easier comparison, in Figure 5.3 we use the average queue backlog of classical backpressure algorithm (BP-D) as a performance reference. For each input rate, the average queue backlog of different strategies is normalized by the corresponding average queue backlog of classical backpressure algorithm (BP-D). (Hence BP-D has a normalized queue backlog 1 for all input rates.) By Little's theorem, average queue backlog is proportional to average delay, and hence normalized queue backlog can also be interpreted as normalized delay.

Several observations are made from the results:

1) BP-R has uniformly better delay performance than BP-D. BP-D and BP-R perform almost the same in low traffic region. But the performance of BP-R starts to improve as the traffic increases.

2) ORCD-R has uniformly better delay performance than BP-R. The performance gain of ORCD-R over BP-R is more evident in low traffic region. And the gain decreases as the traffic increases. And the two curves almost converge as the traffic load approaches 1.

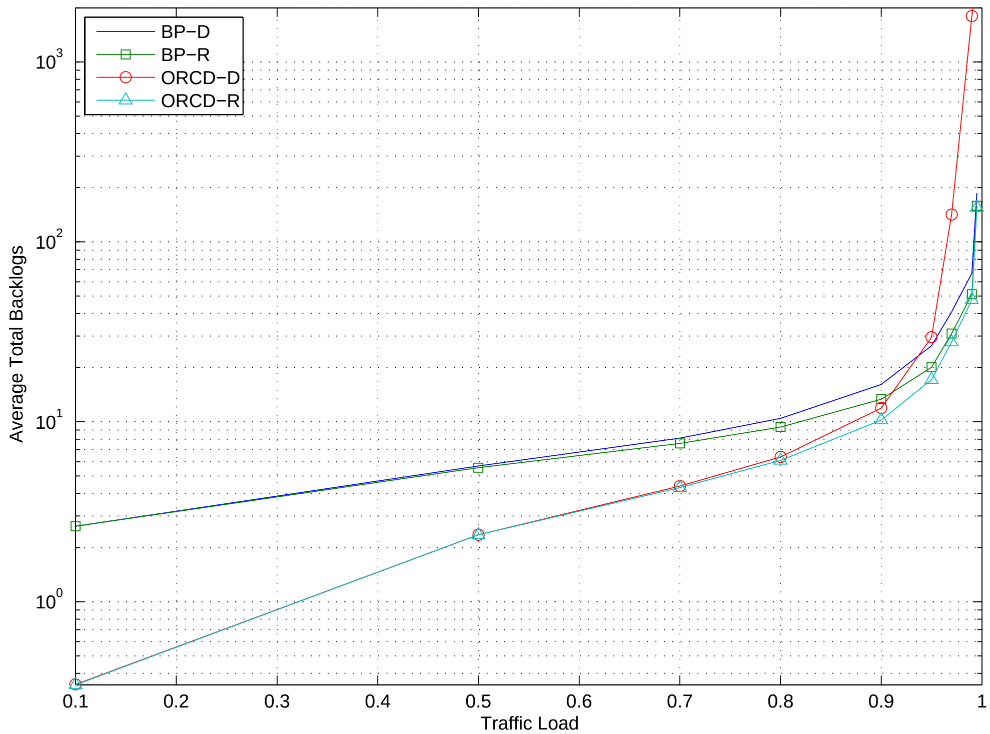

Figure 5.2: Average total backlogs of the 4 nodes network in Figure 5.1

3) ORCD-R has uniformly better delay performance than ORCD-D. ORCD-D and ORCD-R perform almost the same in low traffic region. But the performance of ORCD-D degrades rapidly in high traffic region and eventually becomes instable. From this example we can see that selecting commodity based on D criterion is generally not throughput optimal for K policy based algorithm.

A heuristic explanation why R criterion shows better delay performance than D criterion: To reduce backlog build-up, it is important to avoid empty queue. With the same backlog difference among different commodities, R criterion tends to give priority to the commodity that has higher tendency to have empty queues. For example, in Figure 5.4, node 1 has two commodities to forward. For the given backlogs, D criterion would select commodity 2 while R criterion

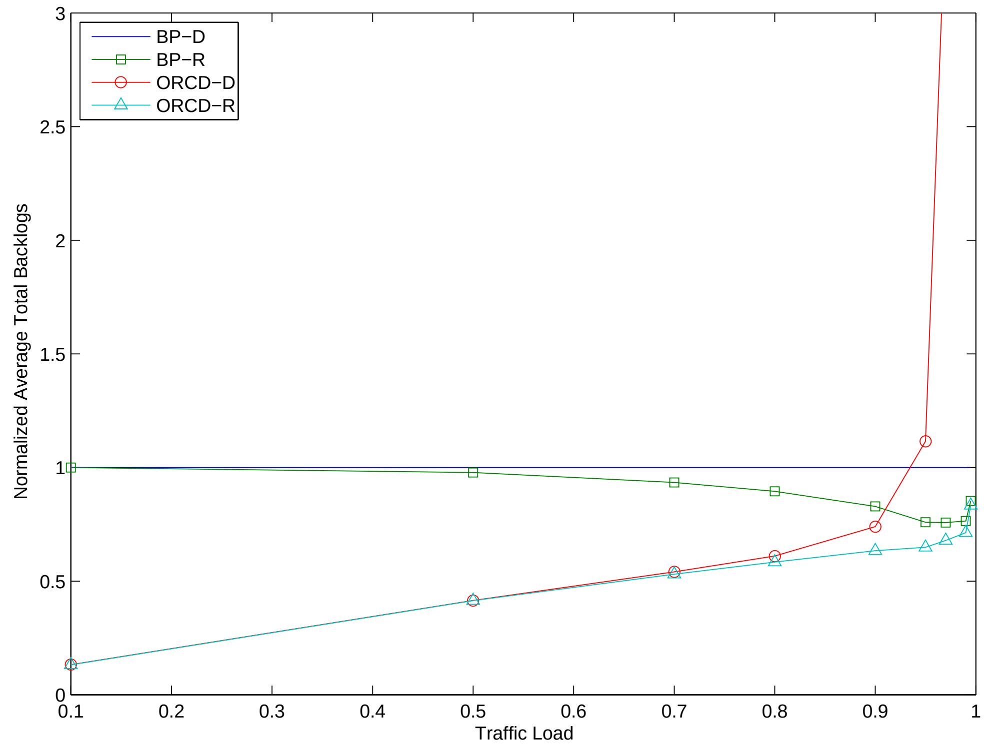

Figure 5.3: Normalized average total backlogs of the 4 nodes network in Figure 5.1

would select commodity 1. Since commodity 1 at node 2 has smaller backlog than commodity 2 at node 3, it is more likely to become empty in future. Hence intuitively, selecting commodity 1 is a better choice in terms of avoiding empty queue and reducing backlog build-up.

5.4 Numerical Results: CDI-Tx Case

Now we turn to CDI-Tx case. The routing schemes proposed under CDI-Tx assumption include DIVBAR [25], E-DIVBAR [20], DIVBAR-R and Opportunistic ORCD-R. We are going to compare these routing schemes under the same 2-commodity network shown in Figure 5.1.

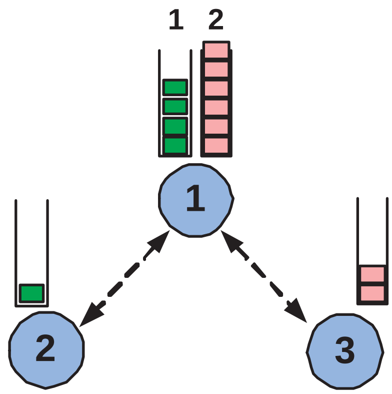

Figure 5.4: An example in which R criterion and D criterion choose different commodities

Figure 5.5 and 5.6 show the simulation results of the 4 nodes network in Figure 5.1. We constraint our arrival rate to satisfy $\lambda_1 = \lambda_2 = \rho C_{max}$, where C_{max} is the upper bound of arrival rate of each commodity as given in Section 5.3. As expected, the maximum arrival rate that λ_1 and λ_2 can achieve simultaneously under CDI-Tx assumption is lower than that under CSI-Tx assumption (around 80%). This is due to the fact that the network under CDI-Tx assumption doesn't have CSI at transmitters and thus has a smaller capacity region.

For ease of comparison, in Figure 5.6 we normalized the average queue backlog of different strategies by the average queue backlog of DIVBAR. Hence DIVBAR has a normalized queue backlog 1 for all input rates. By Little's theorem, average queue backlog is proportional to average delay. So the normalized queue backlog can also be interpreted as normalized delay.

Several observations are made from the results:

1) As expected , E-DIVBAR shows better delay performance than DIVBAR especially in the low traffic region [20].

2) DIVBAR-R shows better delay performance than DIVBAR especially in the high traffic region.

3) From 1) and 2), it is natural to combine E-DIVBAR and DIVBAR-R to

take advantages of both schemes. The combined scheme E-DIVBAR-R is the same as DIVBAR-R described above except that instead of using actual queue backlog $Q_a^c(t)$, the following modified backlog metric $\tilde{Q}_a^c(t)$ is used throughput the whole algorithm [20]:

$$\tilde{Q}_a^c(t) := Q_a^c(t) + X_a^c \tag{5.12}$$

where X_a^c is proportional to the estimated number of hops from node a to destination of commodity data c. For example, in our simulation, X_a^c is taking value as ETX defined in [42]. As we can see that E-DIVBAR-R has evident delay performance improvement over DIVBAR in both low traffic and high traffic region. Moreover, it outperforms DIVBAR-R in all traffic region, and outperforms E-DIVBAR except for the low traffic region.

4) Finally, the opportunistic ORCD-R has uniformly the best delay performance among all schemes in all traffic region.

5.5 Acknowledgement

Chapter 5 is in part a reprint of the material in paper: H. Zhuang and R. Cruz, "Throughput Optimal Routing in Multi-commodity Wireless Ad-hoc Networks", *in preparation*. The dissertation author was the primary investigator and author of the paper.

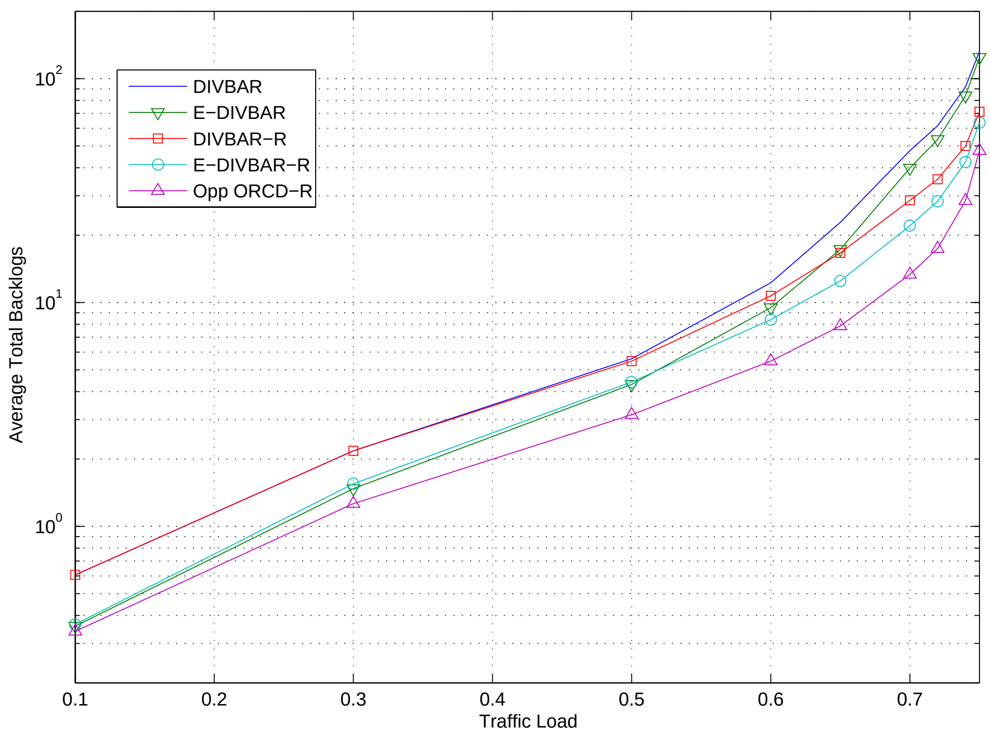

Figure 5.5: Average total backlogs of the 4 nodes network in Figure 5.1

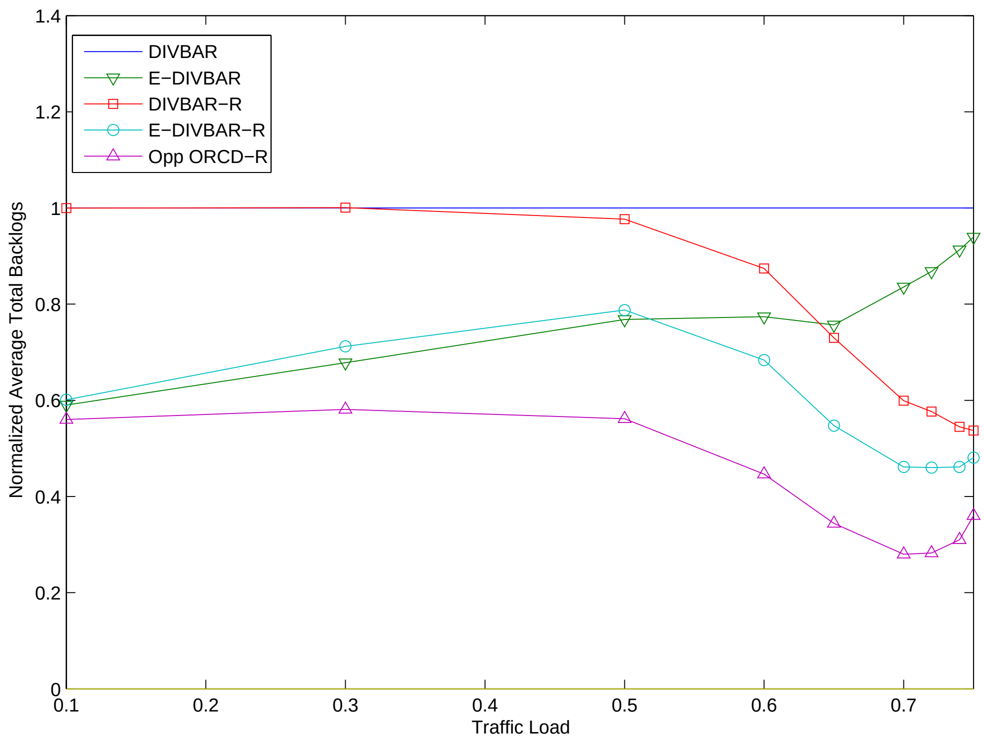

Figure 5.6: Normalized average total backlogs of the 4 nodes network in Figure 5.1

Chapter 6

Concluding Remarks

This dissertation considered the problem of routing multi-commodity data over a multi-hop wireless ad-hoc network. The primary goal is to find routing policies beyond backpressure type that not only ensure throughput optimality but also have improved average delay performance.

The main contributions of this dissertation include

1) In the single commodity scenario, by considering a class of continuous, differentiable, and piece-wise quadratic Lyapunov functions, we proposed a large class of throughput optimal routing policies called K policies, which include backpressure algorithm as a special case. The proposed class of Lyapunov functions allow for the routing policies to control the traffic along short paths for a large portion of state-space while ensuring a negative expected drift, hence, enabling the design of routing policies with much improved delay performances. We specialized our result to recover and prove the throughput optimality of two known routing policies,i.e. backpressure and ORCD.

2) We then extended K-policy to multi-commodity case by considering non-quadric Lyapunov functions. A multi-commodity version of ORCD algorithm for multi-commodity routing problem was proposed based on K-R policy and was

shown to be throughout optimal under mild conditions. Interestingly, the algorithm we proposed was designed to select the commodity that has the maximum backlog ratio instead of the maximum difference of backlogs as in the classical backpressure algorithm. Indeed, we showed via a counter example that selecting commodity based on the difference of backlog might not be throughput optimal for K policy based algorithm.

3) We showed via simulations that R criterion shows better delay performance than D criterion in all scenarios we considered. Particularly, ORCD-R and opportunistic ORCD-R show very promising delay performance improvement over with existing schemes.

Some of future research areas include

1) Analytical delay performance evaluation: In this dissertation, the delay performance was evaluated via numerical simulations. No analytical delay performance analysis is given. In [33], a congestion bound as a by product of Lyapunov drift criterion was derived for backpressure algorithm. A similar congestion bound can be derived for K policy. However, we found that the congestion bound produced by Lyapunov drift criterion is rather loose, which makes it not useful in practice for performance evaluation. A more accurate analytical delay performance evaluation will be an important topic of future research.

2) Flow control and fairness: In this dissertation, we have considered the problem of controlling a network to achieve stability when the exogenous arrival rate is within the capacity region. However, in practice, the capacity is hardly known and very often the exogenous arrival is outside the capacity region. In this case, the network cannot be stabilized without a transport layer flow control mechanism to limit the amount of data that is admitted. Therefore, it is of interests to design a cross-layer strategy for joint flow control and routing that provides stability while achieving optimal network fairness. There are many different ways

to measure fairness. For example, in [43][33][44][45], the authors measure fairness in terms of a utility function of the long term admission rates of each session. By extending Lyapunov drift theory to enables stability and performance optimization to be treated simultaneously, the authors proposed a modified backpressure type dynamic control algorithm that stabilizes the network and at the same time pushes the achieved utility arbitrarily close to optimal. This performance however, comes at the cost of a linear increase in network congestion. Though similar techniques can be applied to K policy based routing algorithm, we found that the convergence to optimal utility is rather slow due to the loose congestion bound used in the analysis. It is therefore of interests to develop a practical flow control mechanism to more efficiently control the data flow and achieve stability and fairness at the same time.

Appendix A

Proofs

A.1 Proof of Lemma 3.1

1)

Consider a one-step confinement list of P defined by

$$P' := (C_1, ..., C_{i-1}, \{k\}, C_i/\{k\}, C_{i+1}, ..., C_{|P|})$$

where $k \in C_i$ and $C_i/\{k\}$ is the set of all nodes in C_i except node k.

Since P is critical and k is a non-isolated node of P, P' is also critical, i.e. $P' \in \mathbb{P}_{\mathcal{L}}$. Then $D_K^{\mathcal{L}}(P)$ by definition has the following constraint:

$$W_K(|C_{<i}|, 1)Q_k \geq W_K(|C_{<i}| + 1, |C_i| - 1)Q_{C_i/\{k\}} \tag{A.1}$$

where $C_{<i} := \bigcup_{j=1}^{i-1} C_j$ is the set of nodes in priority classes higher than C_i. i.e.

$$(W_K(|C_{<i}|, 1) + W_K(|C_{<i}| + 1, |C_i| - 1))Q_k \geq W_K(|C_{<i}| + 1, |C_i| - 1)Q_{C_i} \tag{A.2}$$

After some algebra, we have

$$\begin{aligned} Q_k &\geq \frac{W_K(|C_{<i}|, |C_i| - 1)}{W_K(|C_{<i}|, 1) + W_K(|C_{<i}| + 1, |C_i| - 1)} Q_{C_i} \\ &= \frac{W_K(|C_{<i}|, |C_i|)}{W_K(|C_{<i}|, 1)} Q_{C_i} = \frac{K - 1}{K^{|C_i|} - 1} Q_{C_i} \end{aligned} \tag{A.3}$$

2) By definition, $D_K^{\mathcal{L}}(P)$ also has the following constraints:

$$W_K(|C_{<s+1}|, |C_{s+1}|)Q_{C_{s+1}} \geq W_K(|C_{<s}|, |C_s|)Q_{C_s} \tag{A.4}$$

for $s = 1, ..., i-1$. So

$$\frac{W_K(|C_{<i}|, |C_i|)}{W_K(|C_{<s}|, |C_s|)}Q_{C_i} \geq Q_{C_s} \tag{A.5}$$

for $s = 1, ..., i-1$.

Summing over $s = 1, ..., i-1$ yields

$$W_K(|C_{<i}|, |C_i|)\sum_{s=1}^{i-1}\frac{1}{W_K(|C_{<s}|, |C_s|)}Q_{C_i} \geq \sum_{s=1}^{i-1} Q_{C_s} \tag{A.6}$$

By Proposition 3.5,

$$\sum_{s=1}^{i-1}\frac{1}{W_K(|C_{<s}|, |C_s|)} = \sum_{s=1}^{i-1}\frac{1}{W_K(\sum_{j=1}^{s-1}|C_j|, |C_s|)} = \frac{1}{W_K(0, |C_{<i}|)} \tag{A.7}$$

Substitute (A.7) into (A.6)

$$\frac{W_K(|C_{<i}|, |C_i|)}{W_K(0, |C_{<i}|)}Q_{C_i} \geq Q_{C_{<i}} \tag{A.8}$$

Combining (A.8) and (A.3)

$$Q_k \geq \frac{W_K(0, |C_{<i}|)}{W_K(|C_{<i}|, 1)}Q_{C_{<i}} = \frac{K^m(K-1)}{K^m - 1} > (K-1)Q_{C_{<i}} \tag{A.9}$$

where we use the fact that $|C_{<i}| := \sum_{j=1}^{i-1}|C_j| > 0$, when $i \geq 2$.

A.2 Proof of Proposition 3.6

To prove that $\{D_K^{\mathcal{L}}(P), P \in \mathbb{P}_{\mathcal{L}}\}$ are a complete partition, we have to prove the following two statements:

1) Prove $int(D_K^{\mathcal{L}}(P_1)) \cap int(D_K^{\mathcal{L}}(P_2)) = \varnothing$ for and $P_1, P_2 \in \mathbb{P}_{\mathcal{L}}$ and $P_1 \neq P_2$. i.e. no intersections between any of two cones except for the boundary points.

Pick any two distinct list $P = \{C_1, C_2, ..., C_{|P|}\}$ and $P' = \{C'_1, C'_2, ..., C'_{|P'|}\}$ from $\mathbb{P}_{\mathcal{L}}$. There are two cases:

Case A: P and P' has common refinement, i.e. that there exist P' such that $P' \preceq P$ and $P' \preceq P'$ (Note that it is possible that $P' = P$ or $P' = P'$). Then there exist a priority class from one of the pair (WLOG say C'_j from P') and two consecutive priority classes from the other list of the pair (say C_k and C_{k+1} from P) such that for some sets of nodes U_1, U_2, V_1 and V_2, the following relationship hold

$$C'_j = U_1 \cup U_2 \tag{A.10}$$

$$C_k = V_1 \cup U_1 \tag{A.11}$$

and

$$C_{k+1} = U_2 \cup V_2 \tag{A.12}$$

where U_1 and U_2 are non-empty while V_1 and V_2 could be empty. Such sets could always be found for any distinct lists with a common refinement.

The following facts are important:

Fact 1: The one-step refinements of P, $\{C_1, ..., C_{k-1}, V_1, U_1, C_{k+1}, ..., C_{|P|}\}$ and $\{C_1, ..., C_k, U_2, V_2, C_{k+2}, ..., C_{|P|}\}$ are critical and thus belong to $\mathbb{P}_{\mathcal{L}}$. (The argument is trivial if V_1 and V_2 are empty)

Fact 2: The one-step refinement of P', $\{C'_1, ..., C'_{j-1}, U_1, U_2, C'_{j+1}, ..., C'_{|P'|}\}$ is critical and thus belongs to $\mathbb{P}_{\mathcal{L}}$.

(Quick proof of Fact 1 and 2: Suppose $\{C_1, ..., C_{k-1}, V_1, U_1, C_{k+1}, ..., C_{|P|}\}$ is not critical. Since P is critical, the only possibility is that some node in V_1 has no path to destination via nodes in priority classes not lower than that of V_1. On the other hand, the fact that P' is critical implies that all nodes in V_1 must have a path to destination via nodes in priority classes not lower than that of V_1, which result in a contradiction. The proof of the rest statements are similar)

If V_1 and V_2 are not empty, by Fact 1, we have the following constraints for

$D_K^{\mathcal{L}}(P)$:

$$W_K(|C_{<k}|, |V_1|)Q_{V_1} \geq W_K(|C_{<k}| + |V_1|, |U_1|)Q_{U_1} \tag{A.13}$$

$$W_K(|C_{<k+1}|, |U_2|)Q_{U_2} \geq W_K(|C_{<k+1}| + |U_2|, |V_2|)Q_{V_2} \tag{A.14}$$

From (A.13), we have

$$\frac{Q_{U_1} + Q_{V_1}}{W_K(|C_{<k}| + |V_1|, |U_1|)} \geq (\frac{1}{W_K(|C_{<k}|, |V_1|)} + \frac{1}{W_K(|C_{<k}| + |V_1|, |U_1|)})Q_{U_1} \tag{A.15}$$

Noticing that $Q_{U_1} + Q_{V_1} = Q_{C_k}$ by (A.11) and applying Proposition 3.5 to the right side hand, we have

$$\frac{1}{W_K(|C_{<k}| + |V_1|, |U_1|)}Q_{C_k} \geq \frac{1}{W_K(|C_{<k}|, |C_k|)}Q_{U_1} \tag{A.16}$$

From (A.14), we have

$$(\frac{1}{W_K(|C_{<k+1}|, |U_2|)} + \frac{1}{W_K(|C_{k+1}| + |U_2|, |V_2|)})Q_{U_2} \geq \frac{1}{W_K(|C_{<k+1}|, |U_2|)}(Q_{U_2} + Q_{V_2}) \tag{A.17}$$

Noticing that $Q_{U_2} + Q_{V_2} = Q_{C_{k+1}}$ by (A.12) and applying Proposition 3.5 to the left side hand, we have

$$\frac{1}{W_K(|C_{<k+1}|, |C_{k+1}|)}Q_{U_2} \geq \frac{1}{W_K(|C_{<k+1}|, |U_2|)}Q_{C_{k+1}} \tag{A.18}$$

Note that (A.16) and (A.18) hold trivially when V_1 and V_2 are empty (then $U_1 = C_k$ and $U_2 = C_{k+1}$)

Since $\{C_1, ..., C_{k-1}, C_k \cup C_{k+1}, ...C_{|P|}\}$ is a one-step confinement list of P, we have the following constraint of $D_K^{\mathcal{L}}(P)$

$$W_K(|C_{<k}|, |C_k|)Q_{C_k} \geq W_K(|C_{<k+1}|, |C_{k+1}|)Q_{C_{k+1}} \tag{A.19}$$

Combining (A.16), (A.18) and (A.19), we have

$$W_K(|C_{<k+1}|, |U_2|)Q_{U_2} \geq W_K(|C_{<k}|, |U_1|)Q_{U_1} \tag{A.20}$$

Note that strict inequality holds in (A.20) for $\boldsymbol{Q} \in intD_K^{\mathcal{L}}(P)$.

On the other hand, from Fact 2, we have the following constraint for $D_K^{\mathcal{L}}(P')$:

$$W_K(|C_{<k+1}|, |U_2|)Q_{U_2} \leq W_K(|C_{<k}|, |U_1|)Q_{U_1} \tag{A.21}$$

And strict inequality holds for $\boldsymbol{Q} \in intD_K^{\mathcal{L}}(P')$.

By comparing (A.20) and (A.21), we conclude that $intD_K^{\mathcal{L}}(P) \cap intD_K^{\mathcal{L}}(P') = \varnothing$.

Case B: P and P' don't have a common refinement. Now we claim that there exist non-isolated node a from P' and non-isolated node b from P such that a is in a higher priority class than b under P but in a lower priority class than b under P'. i.e. we must have

$$a \in C_{<k}, b \in C_{\geq k} \tag{A.22}$$

and

$$b \in C'_{<j}, a \in C'_{\geq j} \tag{A.23}$$

for some priority class index j,k.

If the claim is true, then by Lemma 3.20 (2), for $\boldsymbol{Q} \in intD_K^{\mathcal{L}}(P')$, we have

$$Q_a > (K-1)Q_{C'_{<j}} \geq Q_b \tag{A.24}$$

Whereas for $\boldsymbol{Q} \in intD_K^{\mathcal{L}}(P)$, we have

$$Q_b > (K-1)Q_{C_{<k}} \geq Q_a \tag{A.25}$$

Clearly, (A.24) and (A.25) cannot hold simultaneously when $K \geq 2$. Therefore, $intD_K^{\mathcal{L}}(P) \cap intD_K^{\mathcal{L}}(P') = \varnothing$.

It remains to show the claim is true. First of all, since P and P' don't have a common refinement, there exist $\tilde{a}$ and $\tilde{b}$ (not necessarily non-isolated) such that

$$\tilde{a} \in C_{<k}, \tilde{b} \in C_{\geq k} \tag{A.26}$$

and

$$\tilde{b} \in C'_{<j}, \tilde{a} \in C'_{\geq j} \tag{A.27}$$

for some priority class index j,k.

If $\tilde{b}$ is a non-isolated node of P and $\tilde{a}$ is a non-isolated node of P', then we are done. If not, we can do the following to find non-isolated nodes a and b:

Since P is a critical list, by definition, there exists a path starting from $\tilde{a}$ to destination 0 that only possing nodes in $C_{<k}$. Now consider this path under priority list P'. Since the path starting from $\tilde{a} \in C'_{\geq j}$ to destination 0, there exists a node (name it a) belongs to $C'_{\geq j}$ and transmit to nodes in $C'_{<j}$. By construction, a is a non-isolated node under P', and $a \in C_{<k}$, $a \in C'_{\geq j}$.

By symmetry, we can repeat the same procedure for node $\tilde{b}$ to find b.

This completes the proof of the claim.

Now combining Case A and Case B, we conclude that

$int(D_K^{\mathcal{L}}(P)) \cap int(D_K^{\mathcal{L}}(P')) = \varnothing$ for any $P, P' \in \mathbb{P}_{\mathcal{L}}$ and $P \neq P'$.

2) Prove $\bigcup_{P \in \mathbb{P}_{\mathcal{L}}} D_K^{\mathcal{L}}(P) = \mathbb{R}_+^N$. i.e. each of point in $\mathbb{R}_+^N$ must belong to $D_K^{\mathcal{L}}(P)$ for some P.

The proof of this part can be found in the appendix of [46].

□

A.3 Proof of Lemma 4.3

To prove Lemma 4.3, we need a few more lemmas:

Lemma A.1 *If K policy $\mathcal{P}_K$ ($K \geq 2$) based on $\mathbb{P}_{\mathcal{L}}$ is used in a network with link set $\mathcal{L}'$ and $\mathcal{L}' \subseteq \mathcal{L}$, then for any priority list $P^c = (C_1^c, C_2^c, ..., C_{|P^c|}^c) \in \mathbb{P}_{\mathcal{L}}^c$, and*

$\boldsymbol{Q}^c(t) \in D_K^{\mathcal{L}}(P^c)$

$$\frac{G(W_K(|C^c_{<i}|,|C^c_i|)Q^c_{C^c_i}(t+1))}{W_K(|C^c_{<i}|,|C^c_i|)} - \frac{G(W_K(|C^c_{<i}|,|C^c_i|)Q^c_{C^c_i}(t))}{W_K(|C^c_{<i}|,|C^c_i|)}$$
$$\leq \beta_K - 2g(W_K(|C^c_{<i}|,|C^c_i|)Q^c_{C^c_i}(t))(\mu^c_{C^c_i,out}(t) - \mu^c_{C^c_i,in}(t) - A^c_{C^c_i}(t)) \quad \text{(A.28)}$$

where β_K is some constant.

Proof:

By Lemma 3.4, when K policy is used, there exists α_K such that if $\boldsymbol{Q}(t) \in D_K^{\mathcal{L}}(P)$ and $Q_{C^c_i}(t) > \alpha_K$, then

$$Q_{C^c_i}(t+1) \leq Q_{C^c_i}(t) - \mu^c_{C^c_i,out}(t) + \mu^c_{C^c_i,in}(t) + A^c_{C^c_i}(t) \quad \text{(A.29)}$$

On the other hand, if $Q_{C^c_i}(t) \leq \alpha_K$, then

$$Q_{C^c_i}(t+1) \leq Q_{C^c_i}(t) + \mu^c_{C^c_i,in}(t) + A^c_{C^c_i}(t) \quad \text{(A.30)}$$

(A.28) follows by applying Lemma 4.2 with

$$Q^+ = W_K(|C^c_{<i}|,|C^c_i|)Q_{C^c_i}(t+1)$$
$$Q = W_K(|C^c_{<i}|,|C^c_i|)Q_{C^c_i}(t)$$
$$A = W_K(|C^c_{<i}|,|C^c_i|)(\mu^c_{C^c_i,in} + A^c_{C^c_i})$$
$$v = W_K(|C^c_{<i}|,|C^c_i|)\alpha_K$$
$$\mu = W_K(|C^c_{<i}|,|C^c_i|)\mu^c_{C^c_i,out}$$
$$c = NW_K(|C^c_{<i}|,|C^c_i|)\max\{\mu^c_{max}, A_{max}\}$$

.

□

The following lemma is a trivial extension of Lemma 3.6 for the non-quadratic Lyapunov function.

Lemma A.2 *If K policy $\mathcal{P}_K$ $(K \geq 2)$ based on $\mathbb{P}_{\mathcal{L}}$ is used in a network with link set $\mathcal{L}'$ and $\mathcal{L}' \subseteq \mathcal{L}$, then for any priority list $P = (C_1^c, C_2^c, ..., C_{|P|}^c) \in \mathbb{P}_{\mathcal{L}}$, and $\boldsymbol{Q}(t) \in D_K^{\mathcal{L}}(P)$*

$$L_K(\boldsymbol{Q}(t+1); \mathbb{P}_{\mathcal{L}}) - L_K(\boldsymbol{Q}(t); \mathbb{P}_{\mathcal{L}}) \leq \gamma_K - 2 \sum_c \sum_i g(W_K(|C_{<i}^c|, |C_i^c|) Q_{C_i^c}(t))(\mu_{C_i^c,out}^c(t) - \mu_{C_i^c,in}^c(t) - A_{C_i^c}^c(t)) + o(\|\boldsymbol{Q}(t+1) - \boldsymbol{Q}(t)\|) \quad \text{(A.31)}$$

where γ_K is some constant.

Proof: Except for using Lemma A.1 instead of Lemma 3.5, the proof is almost identical to the proof of Lemma 3.6.

□

Now, we are ready to prove Theorem 4.3.

Proof of Theorem 4.3:

Given queue state $\boldsymbol{Q}$, if for each commodity c, $\boldsymbol{Q}^c \in D_K^{\mathcal{L}}(P^c)$, where $P^c = (C_1^c, C_2^c, ..., C_{|P|}^c)$, then by Lemma A.2:

$$L_K(\boldsymbol{Q}(t+1); \mathbb{P}_{\mathcal{L}}) - L_K(\boldsymbol{Q}(t); \mathbb{P}_{\mathcal{L}}) \leq \gamma_K - 2 \sum_c \sum_i g(W_K(|C_{<i}^c|, |C_i^c|) Q_{C_i^c}^c(t))(\mu_{C_i^c,out}^c(t) - \mu_{C_i^c,in}^c(t) - A_{C_i^c}^c(t)) + o(\|\boldsymbol{Q}(t+1) - \boldsymbol{Q}(t)\|) \quad \text{(A.32)}$$

Taking the conditional expectation yields,

$$
\begin{aligned}
&\Delta L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}}) = \mathbb{E}\{L_K(\boldsymbol{Q}(t+1);\mathbb{P}_{\mathcal{L}}) - L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}})|\boldsymbol{Q}(t)\}\\
\leq&\gamma_K - 2\sum_c\sum_i g(W_K(|C^c_{<i}|,|C^c_i|)Q^c_{C^c_i}(t))\mathbb{E}\left\{(\mu^c_{C^c_i,out}(t) - \mu^c_{C^c_i,in}(t) - A^c_{C^c_i}(t))|\boldsymbol{Q}(t)\right\}\\
&+ o(\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|)\\
=&\gamma_K - 2\mathbb{E}[f_K(\boldsymbol{Q}(t),\boldsymbol{\mu}(t))] - \sum_c\sum_i g\left(W_K(|C^c_{<i}|,|C^c_i|)Q^c_{C^c_i}(t)\right)\lambda_{C^c_i}\\
&+ o(\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|)
\end{aligned}
\tag{A.33}
$$

Since $\boldsymbol{\lambda}$ is strictly within the capacity region $\gamma\Lambda$, there exist a positive vector $\epsilon > 0$ such that $\frac{\boldsymbol{\lambda}}{\gamma} + \boldsymbol{\epsilon} \in \Lambda$. By Theorem 2.4 there exists a stationary randomized algorithm that makes decisions based only on the current topology state (and hence independent of the current queue backlog) so that

$$
\mathbb{E}\left\{(\tilde{\mu}_{C^c_i,out}(t) - \tilde{\mu}_{C^c_i,in}(t) - \frac{A^c_{C^c_i}(t)}{\gamma})|\boldsymbol{Q}(t)\right\} \geq \epsilon \tag{A.34}
$$

where $\tilde{\mu}_{C^c_i,out}(t)$ and $\tilde{\mu}_{C^c_i,out}(t)$ denote the amount of date transmitted under such randomizeD criterion.

By the given condition, we have

$$
\mathbb{E}_{\boldsymbol{H}}\left\{f_K(\boldsymbol{Q},\hat{\boldsymbol{\mu}}(t))\right\} \geq \gamma\max_{\pi\in\Pi}\mathbb{E}_{\boldsymbol{H}}\left\{f_K(\boldsymbol{Q},\tilde{\boldsymbol{\mu}}(t))\right\} - D \tag{A.35}
$$

So start from (A.33), we have

$$
\begin{aligned}
&\Delta L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}}) \\
\leq&\gamma_K - 2\mathbb{E}\left[f_K(\boldsymbol{Q}(t),\hat{\boldsymbol{\mu}}(t))\right] - \sum_c\sum_i g\left(W_K(|C^c_{<i}|,|C^c_i|)Q^c_{C^c_i}(t)\right)\lambda_{C^c_i} \\
&+ o(\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|) \\
\leq&\gamma_K + D - 2\gamma\mathbb{E}\left[f_K(\boldsymbol{Q}(t),\tilde{\boldsymbol{\mu}}(t))\right] - \sum_c\sum_i g\left(W_K(|C^c_{<i}|,|C^c_i|)Q^c_{C^c_i}(t)\right)\lambda_{C^c_i} \\
&+ o(\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|) \\
=&\gamma_K + D \\
&- 2\gamma\sum_c\sum_i g(W_K(|C^c_{<i}|,|C^c_i|)Q^c_{C^c_i}(t))\mathbb{E}\left\{(\tilde{\mu}_{C^c_i,out}(t) - \tilde{\mu}_{C^c_i,in}(t) - \frac{A^c_{C^c_i}(t)}{\gamma})|\boldsymbol{Q}(t)\right\} \\
&+ o(\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|)
\end{aligned}
\tag{A.36}
$$

By using (A.34), we have

$$
\Delta L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}}) \leq \gamma_K + D - 2\gamma\epsilon\sum_{c,i} g(W_K(|C^c_{<i}|,|C^c_i|)Q^c_{C^c_i}(t)) + o(\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|) \tag{A.37}
$$

Since $W_K(|C^c_{<i}|,|C^c_i|) \geq 1$ for any K, C^c_i and $g(\cdot)$ is non-decreasing, (A.37) implies

$$
\begin{aligned}
&\Delta L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}}) \leq \gamma_K + D - 2\gamma\epsilon\sum_c\sum_i g(Q^c_{C^c_i}(t)) + o(\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|) \\
=&\gamma_K + D - 2\gamma\epsilon\sum_c\sum_a Q^c_a(t) + o(\frac{\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|}{\sum_c\sum_a Q^c_a(t)})\sum_c\sum_a Q^c_a(t) \\
=&\gamma_K + D - 2\gamma\left(\epsilon - o(\frac{\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|}{\sum_c\sum_a Q^c_a(t)})\right)\sum_c\sum_a Q^c_a(t)
\end{aligned}
\tag{A.38}
$$

Since $\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|$ is bounded, $\frac{\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|}{\sum_c\sum_a Q^c_a(t)} \to 0$ as $\sum_c\sum_a Q^c_a(t)$ increases. Therefore, for sufficiently large $\sum_c\sum_a Q^c_a(t)$, $o(\frac{\|\boldsymbol{Q}(t+1)-\boldsymbol{Q}(t)\|}{\sum_c\sum_a Q^c_a(t)}) \leq \frac{\epsilon}{2}$. Thus we conclude for sufficiently large $\sum_c\sum_a Q^c_a(t)$, we have

$$
\mathbb{E}\{L_K(\boldsymbol{Q}(t+1);\mathbb{P}_{\mathcal{L}}) - L_K(\boldsymbol{Q}(t);\mathbb{P}_{\mathcal{L}})|\boldsymbol{Q}(t)\} \leq \gamma_K + D - \gamma\epsilon\sum_c\sum_a Q^c_a(t) \tag{A.39}
$$

The stability follows from Lemma 4.1.

□

Appendix B

A Framework for Optimal Power and Rate Allocation in Multiuser Fading Channels with Average Throughput Constraints

B.1 Introduction

Data applications, such as Internet service, which have become more and more popular in the emerging new generation of wireless systems, have fundamentally different QoS requirement and traffic characteristics than voice applications. Although data application usually require larger long-term throughput, the traffic is burstier and relatively delay tolerant. Using the average throughput, instead of instantaneous SINR, as a QoS measure to exploit the relative delay tolerance of data applications can lead to more efficient resources allocation strategies. By adapting both rate and power based on the channel conditions, the total system capacity could be further increased and the power assumption could be reduced. The problem of adapting power and rate when both transmitter and receiver can track the channel has been extensively studied by the information theory community in the context of ergodic capacity. It has been shown in [47] that single user single antenna ergodic capacity can be achieved with "water filling over fading states". The results have been generalized to multiuser scalar multiaccess channel (MAC) [48] and broadcast channel (BC) [49]. A water-filling technique for vector MAC is proposed in [50], which can asymptotically achieve the maximum sum capacity in a large system with many users and receive antennas. In [51][52], the authors develop "simultaneous water-filling" to maximize the ergodic sum capacity of the MIMO MAC under individual power constraints.

Prior works focused on maximizing the sum capacity or weighted sum capacity, a commonly used figure of interest in multiuser information theory. However, in QoS-based wireless networks, one is more interested in the dual problem: minimizing the average sum power while satisfying the minimum average throughput constraints. Furthermore, information theoretical approaches assume optimal coding and decoding, which are hard to implement in real systems. In practice, simpler suboptimal techniques such as linear multiuser transmitter and receiver (e.g. zero-forcing, linear MMSE) are often used due to their lower complexity. Motivated by these observations, in this work, we are seeking a generic framework to characterize the optimal power and rate allocation polices in a multiuser multiple antenna system with individual average throughput constraints for each user. The approach taken in this work is strongly motivated by the results in [48]. We will focus on the uplink (e.g. multiple access channel) for ease of presentation. However, it is worth noting that most results in the uplink can be carried over to the downlink (e.g. broadcast channel).

B.2 System Model and Problem Formulation

B.2.1 Channel Model

Consider the uplink of a multiuser narrowband wireless system where a set of K users each with a single antenna is communicating with a base station equipped with N antennas. To capture the time-varying nature of the wireless channel, we adopt the *block flat fading* channel model. Let $\boldsymbol{h}_i[n]$ $(i = 1, 2, ..., K)$ denote the channel gain vector between user i and the base station at block time n, where the j-th component $(j = 1, 2, ..., N)$ is the complex channel gain between user i and the j-th antenna of the base station. Denote $\boldsymbol{H}[n] = [\boldsymbol{h}_1[n], ..., \boldsymbol{h}_K[n]]$. The joint fading state process, $\{\boldsymbol{H}[n]\}_{n=1}^{\infty}$ is assumed to be stationary and ergodic. For a fixed block time n, $\boldsymbol{H}[n]$ is a N by K random matrix, which we assume has a continuous density on its sample space $\mathcal{H} \subset \mathfrak{C}^{N\times K}$. We assume the channel is perfectly known at both transmitter and receiver. The uplink discrete time channel model is given by

$$\boldsymbol{y}_{ul}[t] = \boldsymbol{H}[n]\boldsymbol{x}_{ul}[t] + \boldsymbol{w}_{ul}[t], \quad t = 1, 2, ... \tag{B.1}$$

where integer t is the symbol time index, and $n = \lfloor t/L \rfloor$ is the block time index with block length L. $\boldsymbol{x}_{ul}[t] = [x_1[t], ..., x_K[t]]^T$ is the K by 1 transmit vector, where $x_i[t]$ is the information stream of user i. $\boldsymbol{y}_{ul}[t]$ is the N by 1 receive vector. $\boldsymbol{w}_{ul}[t]$ is an N by 1 additive white complex circularly Gaussian process with covariance matrix $\sigma^2\boldsymbol{I}$.

B.2.2 Power and Rate Allocation Policies

A simple power allocation policy $\mathcal{P}$ is a mapping from the fading state space $\mathcal{H}$ to $\Re_+^K$, i.e. $\mathcal{P}(\boldsymbol{H}) = [\mathcal{P}_1(\boldsymbol{H}), ..., \mathcal{P}_K(\boldsymbol{H})]^T$, where $\mathcal{P}_i(\boldsymbol{H})$ is the power allocated to user i when channel is in fading state $\boldsymbol{H}$. One limitation of the simple power allocation policy is that it doesn't allow timesharing. In this work, we consider a general power allocation policy that is a nonnegative function of fading state $\boldsymbol{H}$ and a parameter z, where z is referred as a timesharing parameter. The power allocation process is given by $\{\mathcal{P}(\boldsymbol{H}[n], Z[n])\}_{n=1}^{\infty}$ where $\{Z[n]\}_{n=1}^{\infty}$ is an i.i.d. random process. We assume that $Z[n]$ is uniformly distributed over $[0, 1]$ and independent of $\boldsymbol{H}[n]$. Similarly, a general rate allocation policy $\mathcal{R}$ is a nonnegative function of fading state $\boldsymbol{H}$ and timesharing parameter z. $\mathcal{R}$ is called a simple rate allocation policy if it is only of function of $\boldsymbol{H}$, independent of z.

Let $\bar{\mathcal{P}}$ and $\bar{\mathcal{R}}$ denote the average power of a power allocation policy and average throughput of a rate allocation policy, respectively i.e.,

$$\bar{\mathcal{P}} := \mathbb{E}[\mathcal{P}(\boldsymbol{H}[n], Z[n])], \quad \bar{\mathcal{R}} := \mathbb{E}[\mathcal{R}(\boldsymbol{H}[n], Z[n])]$$

where the expectation is taken with respect to the distribution of $\boldsymbol{H}[n]$ and $Z[n]$.

B.2.3 Detector

An important element of this work is the detector set and function which is defined with a view towards enabling consideration of practical implementation structures. In the uplink of a wireless system, the maximum instantaneous rates users can achieve depends not only on the power allocation $\boldsymbol{p}$ and channel state $\boldsymbol{H}$, but also on the MAC and physical layer schemes employed by the system, which usually include user scheduling, coding, decoding, multiuser detection etc. We capture the impact of all these options through a detector φ and its associated detector function.

Definition B.1 *(detector function) For a given detector φ, its detector function $\boldsymbol{r}^{\varphi}(\boldsymbol{H}, \boldsymbol{p}) = [r_1^{\varphi}(\boldsymbol{H}, \boldsymbol{p}), ..., r_K^{\varphi}(\boldsymbol{H}, \boldsymbol{p})]^T$ is a function that maps fading state $\boldsymbol{H} \in \mathcal{H}$ and power allocation $\boldsymbol{p} = [p_1, ..., p_K]^T \in \Re_+^K$ to a vector of rates, where p_i and*

$r_i^{\varphi}(\boldsymbol{H}, \boldsymbol{p})$ *are the power and rate of the user* i*, respectively. In what follows, we consider detectors with the following structure:*

$$r_i^{\varphi}(\boldsymbol{H}, \boldsymbol{p}) = \Xi(p_i I_i^{\varphi}(\boldsymbol{H}, \boldsymbol{p})), \quad i = 1, 2..., K \tag{B.2}$$

where $\boldsymbol{I}^{\varphi}(\boldsymbol{H}, \boldsymbol{p}) = [I_1^{\varphi}(\boldsymbol{H}, \boldsymbol{p}), ..., I_K^{\varphi}(\boldsymbol{H}, \boldsymbol{p})]$ *for all* $\boldsymbol{p} \geq 0$ *satisfies*

A1) $\boldsymbol{I}^{\varphi}(\boldsymbol{H}, \boldsymbol{p}) \geq 0$ *(Positivity)*

A2) If $\boldsymbol{p} \geq \boldsymbol{p}'$*, then* $\boldsymbol{I}^{\varphi}(\boldsymbol{H}, \boldsymbol{p}) \leq \boldsymbol{I}^{\varphi}(\boldsymbol{H}, \boldsymbol{p}')$ *(Monotonicity)*

And the real function $\Xi : \Re_+ \mapsto \Re_+$ *satisfies*

B1) $\Xi(\gamma)$ *is a nondecreasing function of* γ*.*

B2) $\Xi(0) = 0$

Here, $\boldsymbol{I}^{\varphi}(\boldsymbol{H}, \boldsymbol{p})$ is similar to the interference function introduced in [53], and $p_i I_i^{\varphi}(\boldsymbol{H}, \boldsymbol{p})$ has a physical meaning as SINR. For the case with ideal Gaussian coding and decoding, $\Xi(\gamma) = \log(1 + \gamma)$, which is the classical Shannon capacity formula.

Usually, there is more than one detector implemented in a system and this will become clearer in the application section. Let Φ denote the set of detectors that is implemented in a system. Let $\mathcal{U}_{\boldsymbol{p}}$ be the set of all feasible power allocation dictated by the system design. For a given detector set Φ, let $\mathcal{F}^{\Phi}$ denote the set of all *feasible* rate and power allocation policy pairs.

$$\mathcal{F}^{\Phi} := \{(\mathcal{R}, \mathcal{P}) : \mathcal{P}(\boldsymbol{H}, z) \in \mathcal{U}_{\boldsymbol{p}}, \mathcal{R}(\boldsymbol{H}, z) = \boldsymbol{r}^{\varphi}(\boldsymbol{H}, \mathcal{P}(\boldsymbol{H}, z)), \\ \varphi \in \Phi \quad \text{for all } \boldsymbol{H} \in \mathcal{H} \text{ and } 0 \leq z \leq 1\} \tag{B.3}$$

Let $\mathcal{C}^{\Phi} \subset \Re_+^K$ denote the set of all admissible rates under detector set Φ. i.e.

$$\mathcal{C}^{\Phi} = \{\boldsymbol{R} \in \Re_+^K : \boldsymbol{R} = \bar{\mathcal{R}} \text{ for some } (\mathcal{R}, \mathcal{P}) \in \mathcal{F}^{\Phi}\} \tag{B.4}$$

$\mathcal{C}^{\Phi}$ is a convex set since timesharing is allowed.

B.2.4 Problem Formulation

For a given system with detector set Φ, we are interested in the optimal power allocation policy $\mathcal{P}$ and the rate allocation policy $\mathcal{R}$ that minimize the average weighted sum transmit power while satisfying an average throughput constraint. The problem is formally defined as follows:

> **Problem A:** Given a detector set Φ, some power weight $\boldsymbol{\lambda} \in \Re_+^K$, and rate requirement $\boldsymbol{R}^t \in \mathcal{C}^\Phi$, we want to find a policy pair $(\mathcal{R}^*, \mathcal{P}^*)$ (referred to as optimal policy pair) that achieves the minimum of following constrained optimization problem:
>
> $$\min_{(\mathcal{R},\mathcal{P})\in\mathcal{F}^\Phi} \boldsymbol{\lambda}^T\bar{\mathcal{P}} \quad \text{subject to} \quad \bar{\mathcal{R}} \geq \boldsymbol{R}^t \tag{B.5}$$

Note that the sum power is just a special case where $\boldsymbol{\lambda}$ is an all 1 vector. Let $\mathcal{S}_A$ denote the set of all optimal rate and power allocation pairs that achieve the minimum of (B.5).

B.3 Structural Results

Problem A is a constrained optimization problem over functions. A direct solution appears to be difficult. In this section, we will provide some structural results for the optimal policies in terms of their range. Similar to the results in [48], we will show that for each channel state, there is a corresponding optimization problem over vectors (Problem B). The average rate constraint is taken care of by the rate weight vector $\boldsymbol{\mu}$, which plays a similar role as the Lagrangian coefficient in nonlinear programming. Problem B is defined as follows:

Problem B: Given a vector $\boldsymbol{\mu} \in \Re^K$, a power weight $\boldsymbol{\lambda} \in \Re_+^K$, a detector set Φ and a fading state $\boldsymbol{H} \in \mathcal{H}$, find vector pair $(\varphi^*, \boldsymbol{p}^*)$ that achieves the maximum of following constrained optimization problem:

$$\max_{\varphi\in\Phi,\boldsymbol{p}\in\mathcal{U}_p} \boldsymbol{\mu}^T\boldsymbol{r}^\varphi(\boldsymbol{H},\boldsymbol{p}) - \boldsymbol{\lambda}^T\boldsymbol{p} \tag{B.6}$$

Generally, there could be infinitely many solutions for Problem B. Let $\mathcal{S}_B(\boldsymbol{\mu}, \boldsymbol{H})$ denote the set of all $(\varphi^*, \boldsymbol{p}^*)$ pairs that achieves the maximum of (B.6). For any $(\varphi^*, \boldsymbol{p}^*) \in \mathcal{S}_B(\boldsymbol{\mu}, \boldsymbol{H})$, we refer to $\boldsymbol{p}^*$ and $\boldsymbol{r}^{\varphi^*}(\boldsymbol{H}, \boldsymbol{p}^*)$ as the optimal power and optimal rate of Problem B, respectively.

The relationship between Problem A and Problem B is established by the following theorem. Due to space limitations, the proof is omitted.

Theorem B.1 *Given* Φ*,* $\boldsymbol{R}^t \in int\mathcal{C}^\Phi$ *and* $\boldsymbol{\lambda} > 0$*,* $(\mathcal{R}', \mathcal{P}') \in \mathcal{S}_A$ *(solution set of Problem A) if and only if the following two conditions are satisfied:*

1. *There exists* $\boldsymbol{\mu} \in \Re^K$ *(not necessarily unique) such that for almost every given* $\boldsymbol{H} \in \mathcal{H}$*,*$0 \le z \le 1$*, there exists* $\varphi' = \varphi'(\boldsymbol{H}, z) \in \Phi$ *(not necessarily unique) such that*

$$\mathcal{R}'(\boldsymbol{H}, z) = \boldsymbol{r}^{\varphi'}(\boldsymbol{H}, \mathcal{P}'(\boldsymbol{H}, z)),$$

$$(\varphi', \mathcal{P}'(\boldsymbol{H}, z)) \in \mathcal{S}_B(\boldsymbol{\mu}, \boldsymbol{H})$$

2. $\bar{\mathcal{R}}' = \boldsymbol{R}^t$*.*

Theorem B.1 relates Problem A to Problem B, a simpler optimization problem. However, Problem B determines the range of the optimal policy and determines an unique policy only if the rate and power vectors resulting from solving Problem B is unique. Otherwise, Theorem B.1 offers no constructive procedure for determining the optimal policies. Fortunately, for many practical receiver structures this does not appear to be an issue.

Another issue in applying Theorem B.1 is determining $\boldsymbol{\mu}$. Finding an analytical solution for $\boldsymbol{\mu}$ in Theorem B.1 is difficult. We therefore resort to a numerical approach and propose the use of the following Robbins-Monro (RM) stochastic approximation algorithm [54] to choose $\boldsymbol{\mu}$ and the rate solution in an adaptive manner.

$$\boldsymbol{\mu}[n+1] = \boldsymbol{\mu}[n] - a_n(\boldsymbol{r}(\boldsymbol{\mu}[n], \boldsymbol{H}[n]) - \boldsymbol{R}^t) \tag{B.7}$$

where $\{a_n\}$ are positive step sizes, $\boldsymbol{r}(\boldsymbol{\mu}[n], \boldsymbol{H}[n])$ is an optimal rate of Problem B for current $\boldsymbol{\mu}[n]$ and $\boldsymbol{H}[n]$. If Problem B has multiple optimal rates, $\boldsymbol{r}(\boldsymbol{\mu}[n], \boldsymbol{H}[n])$ is chosen to be an arbitrary solution.

B.4 Applications and Numerical Experiments

In this section, we will apply the structural results to different uplink schemes and compare their performance numerically. We assume $\mathcal{U}_{\boldsymbol{p}} = \bigotimes_{i=1}^{K}[0, \infty)$ and unless specified, $\Xi(\gamma) = \log(1+\gamma)$. The application of the framework is quite straightforward and involves the following steps. For each scheme, a detector set and detector function for each detector are first defined. Then Problem B is solved using the defined detector function and current channel state. The solution is related to Problem A by Theorem B.1. To demonstrate the application and utility of the framework, we now consider some popular detectors.

B.4.1 TDMA with dynamic slot assignment and variable rate (DSA-VR)

In this scheme, only one user is allowed to transmit at any one time. The detector set is then by $\Phi_{TDMA} = \{\varphi_1, ..., \varphi_K\}$, where φ_i is the detector when only user i is transmitting. The detector function is given by

$$r_j^{\varphi_i}(\boldsymbol{H}, \boldsymbol{p}) = \begin{cases} \log(1 + p_i\|\boldsymbol{h}_i\|^2) & j = i \\ 0 & j \neq i \end{cases} \tag{B.8}$$

Note that the detector function implicitly supports the one user at a time transmit policy and this simple example serves to indicate the multifaceted nature of the detector function and hence the framework. By substituting (B.8) into (B.6), Problem B in this setting is reduced to

$$\max_{1\le i\le K} \max_{p_i} \left\{\mu_i \log(1 + p_i\|\boldsymbol{h}_i\|^2) - \lambda_i p_i\right\} \tag{B.9}$$

The solution to the inner optimization of (B.9) is based on the classic water-filling principle

$$p_i = \left[\frac{\mu_i}{\lambda_i} - \frac{1}{\|\boldsymbol{h}_i\|^2}\right]^+ \tag{B.10}$$

And the criterion of choosing the optimal user to transmit is

$$i = \arg\max_{1\le i\le K} \left(\mu_i \left[\log \frac{\mu_i\|\boldsymbol{h}_i\|^2}{\lambda_i}\right]^+ - \left[\mu_i - \frac{\lambda_i}{\|\boldsymbol{h}_i\|^2}\right]^+\right) \tag{B.11}$$

Given current channel state $\boldsymbol{H}$, (B.11) is used to decide which user transmits. If user i is selected, it is allocated with power level given in (B.10) and rate level $\log(1 + p_i\|\boldsymbol{h}_i\|^2)$. Note that it is possible that no user is scheduled to transmit when all users experience deep fading.

B.4.2 TDMA with dynamic slot assignment and fixed rate (DSA-FR)

It is similar to the DSA-VR scheme. The only difference is that the transmission rate of user i is fixed to be C if it is selected for transmission, where $C = \sum_{i=1}^{K} R_i^t$ is the total throughput requirement. This can be done by modifying function Ξ as follows:

$$\Xi(\gamma) = \begin{cases} C & \text{if } \log(1+\gamma) \ge C \\ 0 & \text{if } \log(1+\gamma) < C \end{cases} \tag{B.12}$$

B.4.3 Zero-forcing (ZF)

To further confirm the general nature of the framework developed, we now consider a more complex multiuser receiver, the zero-forcing receiver. Define active antenna set $\mathcal{U}$ as a set that contains indexes of all users with nonzero power allocations. The detector set is then given by

$$\Phi_{ZF} = \{\varphi_{\mathcal{U}} : \mathcal{U} \subset \{1, ..., K\}, 1 \leq |\mathcal{U}| \leq N\} \tag{B.13}$$

where $\varphi_{\mathcal{U}}$ denotes the detector with active user set $\mathcal{U}$. The detector function is given by

$$r_i^{\varphi_{\mathcal{U}}}(\boldsymbol{H}, \boldsymbol{p}) = \begin{cases} \log\left(1 + \frac{p_i}{\|\boldsymbol{g}_{\mathcal{U},i}\|^2}\right) & i \in \mathcal{U} \\ 0 & i \notin \mathcal{U} \end{cases} \tag{B.14}$$

where

$$[\boldsymbol{g}_{\mathcal{U},i_1}, ..., \boldsymbol{g}_{\mathcal{U},i_{|\mathcal{U}|}}] = \boldsymbol{H}_{\mathcal{U}}(\boldsymbol{H}_{\mathcal{U}}^H \boldsymbol{H}_{\mathcal{U}})^{-1} \tag{B.15}$$

To solve Problem B with detector set Φ_{ZF}, we need to consider the following problem

$$\max_{\mathcal{U}} \sum_{i \in \mathcal{U}} \Big(\max_{p_i \geq 0} \mu_i \log\big(1 + \frac{p_i}{\|\boldsymbol{g}_{\mathcal{U},i}\|^2}\big) - \lambda_i p_i \Big) \tag{B.16}$$

The inner maximization (B.16) is simple, and the maximum is obtained when

$$p_i = \left[\frac{\mu_i}{\lambda_i} - \|\boldsymbol{g}_{\mathcal{U},i}\|^2\right]^+ \tag{B.17}$$

and the criterion for choosing optimal active antenna set $\mathcal{U}^*$ is

$$\mathcal{U}^* = \arg\max_{\mathcal{U}} \sum_{i \in \mathcal{U}} \Big(\mu_i \left[\log \frac{\mu_i}{\lambda_i \|\boldsymbol{g}_{\mathcal{U},i}\|^2}\right]^+ - \left[\mu_i - \lambda_i \|\boldsymbol{g}_{\mathcal{U},i}\|^2\right]^+ \Big) \tag{B.18}$$

The framework has also been applied to other more complex receiver structures such as optimal linear receiver (linear MMSE), MMSE-SIC etc and used to provide interesting insight.

We now provide some numerical results to provide a feel for the utility of the framework. For this numerical study, we assume $\boldsymbol{\lambda} = [1, ..., 1]^T$. For comparison purpose, we also include two traditional TDMA schemes: FSA-FR, and FSA-VR. In both schemes, the users transmit in a round-robbin manner. The difference is that in the former one, the user transmits with fixed rate when selected, while the latter uses water-filling power and rate allocation.

The above schemes can be grouped into 3 classes: Class 1 is TDMA schemes with no DSA, including FSA-FR and FSA-VR; Class 2 is TDMA schemes with DSA, including DSA-FR and DSA-VR; Class 3 is SDMA schemes including ZF, linear MMSE, and MMSE-SIC.

Figure B.1 shows the numerical results of the above schemes under different system configurations. Each curve corresponds to one configuration. For example, "4x2 C=4 equal rate" denotes 4 users, 2 receive antennas at the base station, total throughput requirement is 4 bits/symbol and users have the same rate requirement, e.g. $\boldsymbol{R}^t = [1, 1, 1, 1]$ bits/symbol. "4x2 C=4 [2,1,2/3,1/3]" is similar except that the rate requirement of users are not equal and is given by $\boldsymbol{R}^t = [2, 1, 2/3, 1/3]$. The rate weight $\boldsymbol{\mu}$ is found by the adaptive algorithm in (B.7) with diminishing step size $a_n = \frac{1}{n}$. Several observations can be made based on the numerical results:

- Class 2 outperforms Class 1 and the gain is more pronounced when the number of users is large. However, within Class 2, DSA-VR is only slightly better than DSA-FR. This implies that in a system with many users, by taking advantage of multiuser diversity, scheduling alone can achieve most of performance gain. The benefit of additional rate adaption is much smaller than those obtained with scheduling. In practice, DSA-FR might be preferable due to its lower complexity compared with DSA-VR.

- Class 3 outperforms Class 2 due to the spatial multiplexing of SDMA schemes. The gain increases with total throughput and the number of receive antennas. When total throughput requirement is low ($C = 4$), simple TDMA-DSA schemes in Class 2 performs quite close to SDMA schemes in Class 3.

- The performances of schemes within Class 3 are quite close. Linear MMSE is slightly better than ZF as expected. Somewhat surprisingly, MMSE-SIC, which is the optimal scheme, is only about 0.5 db better than linear receiver

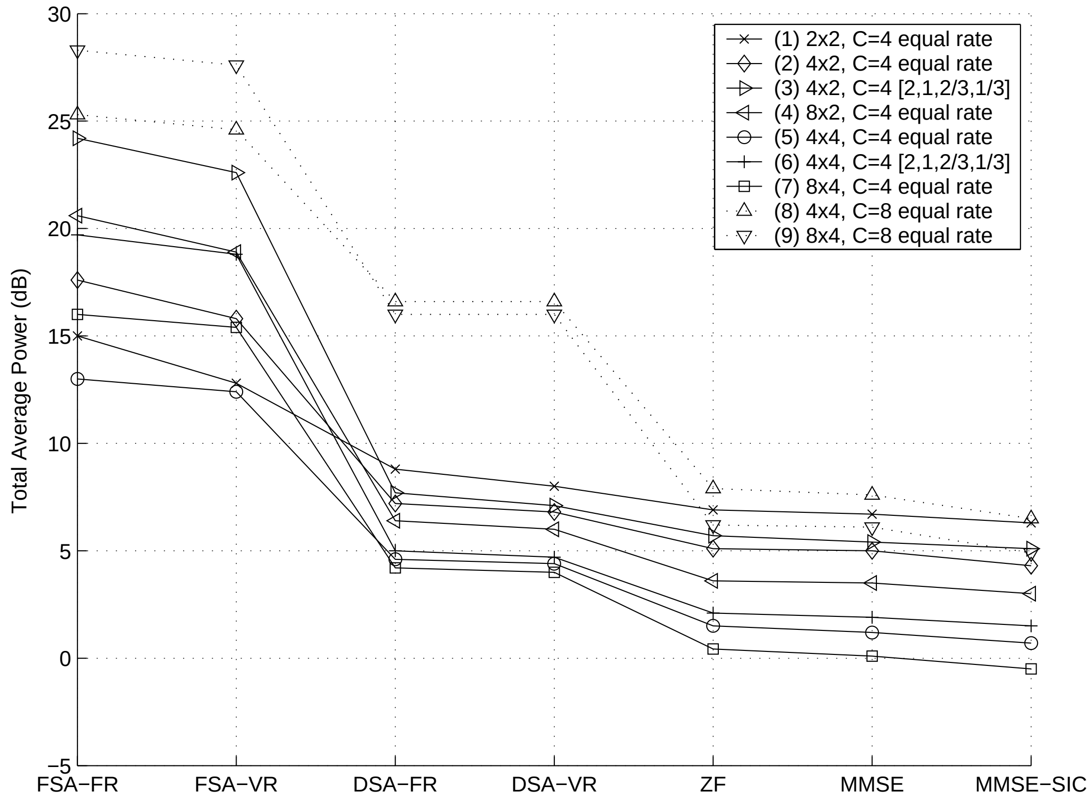

Figure B.1: Comparison of total average power consumption of different schemes

(ZF and MMSE). Considering that error propagation is likely in practical implementations of MMSE-SIC, its performance in reality can be even worse.

B.5 Acknowledgement

Appendix B, in full, is a reprint of the material in paper: H. Zhuang, E. Masry and B. Rao, "Optimal Power And Rate Allocation Framework For The Uplink With Individual Average Rate Requirements", *ICASSP*, 2008. The dissertation author was the primary investigator and author of the paper.

Bibliography

[1] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[2] D. Bertsekas and R. Gallager, *Data Networks.* Prentice-Hall, 1992.

[3] J. Garcia-Luna-Aceves and S. Murthy, "A path-finding algorithm for loop-free routing," *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, 1997.

[4] J. Moy, "OSPF version 2," *RFC 1247*, Jul. 1991.

[5] M. Pearlman and Z. Haas, "Determining the optimal configuration for the zone routing protocol," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, Aug. 1999.

[6] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV)for mobile computers," in *Proceedings of the SIGCOMM 94 Conference on Communications Architectures, Protocols and Applications*, Aug. 1994.

[7] M. Pursley, H. Russell, and P. Staples, "Routing for multimedia traffic in wireless frequency-hop communication networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 5, May 1999.

[8] V. Rodoplu and T. Meng, "Minimum energy mobile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, Aug. 1999.

[9] M. Corson and A. Ephremides, "A distributed routing algorithm for mobile wireless networks," *Wireless Networks*, pp. 61–81, 1995.

[10] E. Gafni and D. Bertsekas, "Distributed algorithms for generating loop-free routes in networks with frequently changing topology," *IEEE Trans. Commun.*, vol. 29, no. 1, 1981.

[11] D. B. Johnson, "Routing in ad hoc networks of mobile hosts," in *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, Dec. 1994.

[12] D. Maltz, J. Broch, J. Jetcheva, and D. Johnson, "The effects of on-demand behavior in routing protocols for multihop wireless ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, Aug. 1999.

[13] V. Park and M. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proceedings of INFOCOM 97*, Apr. 1997.

[14] C. Perkins, "Ad-hoc on-demand distance vector routing," *Internet draft*, Nov. 1997.

[15] P. Larsson, "Selection diversity forwarding in a multihop packet radio network with fading channel and capture," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 2, no. 4, p. 4754, Oct. 2001.

[16] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance," *IEEE Trans. Mobile Comput.*, vol. 2, no. 4, 2003.

[17] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," *ACM SIGCOMM Computer Communication Review*, vol. 35, p. 3344, Oct. 2005.

[18] S. Das and S. Jain, "Exploiting path diversity in the link layer in wireless ad hoc networks," in *World of Wireless Mobile and Multimedia Networks 2005, WoWMoM 2005*, Jun. 2005, pp. 22–30.

[19] C. Lott and D. Teneketzis, "Stochastic routing in ad-hoc networks," *IEEE Trans. Automatic Control*, vol. 51, pp. 52–72, Jan. 2006.

[20] M. J. Neely and R. Urgaonkar, "Optimal backpressure routing for wireless networks with multi-receiver diversity," *University of Southern California Technical Report*, Oct. 2006.

[21] P. Gupta and T. Javidi, "Towards throughput and delay optimal routing for wireless ad-hoc networks," in *Asilomar Conference*, 2007, pp. 249–254.

[22] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automatic Control*, vol. 37, no. 12, pp. 1936–1949, 1992.

[23] K. Ross and N. Bambos, "Local search scheduling algorithms for maximal throughput in packet switches," *Infocom 2004*, 2004.

[24] ——, "Projective cone scheduling (PCS) algorithms for packet switches of maximal throughput," *IEEE/ACM Transactions on Networking (TON)*, 2009.

[25] M. J. Neely and R. Urgaonkar, "Optimal backpressure routing for wireless networks with multi-receiver diversity," *Ad Hoc Networks*, vol. 7, no. 5, pp. 862–881, Jul. 2009.

[26] L. Bui, R. Srikant, and A. L. Stolyar, "Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing," in *INFOCOM Mini-Conference 2009*, 2009.

[27] L. Ying, S. Shakkottai, and A. Reddy, "On combining shortest path and back-pressure routing over multihop wireless networks," *INFOCOM 2009*, 2009.

[28] L. Huang and M. J. Neely, "Delay reduction via lagrange multipliers in stochastic network optimization," *WiOpt 200*.

[29] S. Sarkar and S. Ray, "Arbitrary throughput versus complexity tradeoffs in wireless networks using graph partitioning," *IEEE Trans. Automatic Control*, vol. 53, no. 10, pp. 2307–2323, Nov. 2008.

[30] L. Ying and S. Shakkottai, "On throughput-optimal scheduling with delayed channel state feedback," in *Information Theory and Applications Workshop*, 2008, pp. 339–344.

[31] Y. Xi and E. M. Yeh, "Throughput optimal distributed control of stochastic wireless networks," in *WiOpt*, 2006.

[32] B. Smith and B. Hassibi, "Wireless erasure networks with feedback," in *CoRR abs/0804.4298*, 2008.

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

[34] S. P. Meyn and R. L. Tweedie, *Markov chains and stochastic stability (2nd Edition)*. Springer Verlag, London, 2009.

[35] S. P. Meyn, *Control Techniques for Complex Networks*. Cambridge University Press, 2008.

[36] M. J. Neely, "Dynamic power allocation and routing for satellite and wireless networks with time varying channels," *Ph.D. dissertation, Massachusetts, Institute of Technology, LIDS*, 2003.

[37] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–203, 2005.

[38] M. El-Taha and S. S. Jr., *Sample-Path Analysis of Queueing Systems.* Kluwer Academic Publishers, Boston, 1999.

[39] F. M. Guillemin and R. R. Mazumdar, "On pathwise analysis and existence of empirical distributions for G/G/1 queues," *Stoc. Proc. Appl.*, vol. 67, no. 1, 1997.

[40] F. Foster, "On the stochastic matrices associated with certain queuing processes," *Ann. Math. Statistics*, vol. 24, pp. 355–360, 1953.

[41] M. J. Neely, "Optimal backpressure routing for wireless networks with multi-receiver diversity," in *Conference on information sciences and systems (CISS)*, Mar. 2006.

[42] D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path matric for multi-hop wireless routing," *In proc. ACM/IEEE MobiCom*, Sep. 2003.

[43] M. J. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," in *Proceedings of IEEE INFOCOM*, Mar. 2005.

[44] M. J. Neely, "Optimal energy and delay tradeoffs for multiuser wireless downlinks," *IEEE Trans. Inf. Theory*, vol. 53, no. 9, pp. 3095–3113, Sep. 2007.

[45] ——, "Energy optimal control for time varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, 2006.

[46] M. Naghshvar, H. Zhuang, and T. Javidi, "A general class of throughput optimal routing policies in multi-hop wireless networks," *Submitted to IEEE Transactions on Information Theory*, 2009.

[47] A. Goldsmith and P. Varaiya, "Capacity of fading channel with channel side information," *IEEE Trans. Inf. Theory*, vol. 43, no. 6, pp. 1986–1992, Nov. 1997.

[48] D. Tse and S. Hanly, "Multi-access fading channels: Part I: Polymatroid structure, optimal resource allocation and throughput capacities," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 2796–2815, Nov. 1998.

[49] L. Li and A. J. Goldsmith, "Capacity and optimal resource allocation for fading broadcast channels: Part I: Ergodic capacity," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 1083–1102, Mar. 2001.

[50] P. Viswanath, D. Tse, and V. Anantharam, "Asymptotically optimal water-filling in vector multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 47, no. 1, pp. 241–267, Jan. 2001.

[51] W. Yu, W. Rhee, S. Boyd, and J. M. Cioffi, "Optimal power control in multiple access fading channels with multiple antennas," in *IEEE International Conference on Communications*, 2001.

[52] ——, "Iterative water-filling for gaussian vector multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 50, no. 1, pp. 145–152, Jan. 2004.

[53] R. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341–1347, Sep. 1995.

[54] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, 1951.

[55] J. Broch, D. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. 4th Annu. ACM/IEEE Int. Conf. Mobile Computing and Networking*, 1998.

[56] Z. Haas, "Guest editorial, wireless ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, Aug. 1999.

[57] G. Klimov, "Time sharing service systems I," *Theory of Probability and its Applications, 19*.

[58] R. Berry and R. G. Gallager, "Communication over fading channels with delay constraints," *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1135–1149, May 2002.

[59] E. Telatar and R. G. Gallager, "Combining queueing theory with information theory for multiaccess," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 6, pp. 963–979, Aug. 1995.

[60] S. Raj, E. Telatar, and D. Tse, "Job scheduling and multiple access," *DIMACS Series in discrete mathematics and theoretical computer science*, vol. 66, 2004.

[61] E. Yeh and A. Cohen, "Throughput optimal power and rate control for queued multiaccess and broadcast communications," in *ISIT 2004. Proceedings. International Symposium*, 2004.

[62] R. Negi and S. Goel, "An information-theoretic approach to queuing in wireless channels with large delay bounds," in *Globecomm'04*, 2004.

[63] S. Kittipiyakul and T. Javidi, "Optimal operating point in MIMO channel for delay-sensitive and bursty traffic," in *IEEE ISIT 2006*, Seatle, USA, 2006.

[64] W. Wu and A. Arapostathis, "Optimal power allocation for a wireless channel under heavy traffic approximation," submitted to IEEE. Trans. Automatic Control.

[65] S. Ulukus and R. Yates, "Adaptive power control and MMSE interference suppression," *ACM Wireless Networks*, vol. 4, no. 6, pp. 489–496, Sep. 1998.

[66] J. Luo, S. Ulukus, and A. Ephremides, "Standard and quasi-standard stochastic power control algorithms," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2612–2624, Jul. 2005.

[67] F. Rashid-Farrokhi, L. Tassiulas, and K. J. R. Liu, "Joint optimal power control and beamforming in wireless networks using antenna array," *IEEE Trans. Commun.*, vol. 46, no. 10, pp. 1313–1324, Oct. 1998.

[68] F. Rashid-Farrokhi, K. J. R. Liu, and L. Tassiulas, "Transmit beamforming and power control for cellular wireless systems," *IEEE J. Select. Areas Commun.*, vol. 16, no. 8, pp. 1437–1450, Oct. 1998.

[69] H. Boche and S. Stanczak, "Optimal QoS tradeoff and power control in CDMA systems," in *IEEE INFOCOM 2004*, Hong Kong, 2004.

[70] L. Imhof and R. Mathar, "Capacity regions and optimal power allocation for CDMA cellular radio," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 2011–2019, Jun. 2005.

[71] H. Zhuang, E. Masry, and B. Rao, "A framework for optimal power and rate allocation in multiuser fading channels with average throughput constraints," *submitted to IEEE Signal Processing*.

[72] ——, "Optimal power and rate allocation framework for the uplink with individual average rate requirements," in *ICASSP08*, 2008.

[73] R. Knopp and P. A. Humblet, "Information capacity and power control in single-cell multi-user communications," in *IEEE International Conference on Communications*, 1999, pp. 331–335.

[74] S. Hanly and D. Tse, "Power control and capacity of spread spectrum wireless networks," in *Automatica*, 1999.

[75] D. Tse, "Optimal power allocation over parallel gaussian broadcast channels," unpublished.

[76] S. Vishwanath, S. Jafar, and A. Goldsmith, "Optimum power and rate allocation strategies for multipleaccess fading channels," in *Vehicular Technology Conference*, 2001.

[77] X. Wang and G. B. Giannakis, "Resource allocation for power-efficient tdma under individual rate constraints," in *IEEE International Conference on Communications*, 2007.

[78] W. Yu and W. Rhee, "Degrees of freedom in wireless multiuser spatial multiplex systems with multiple antennas," *IEEE Trans. Commun.*, vol. 54, no. 10, pp. 1747–1753, Oct. 2006.

[79] P. Viswanath and D. Tse, "Sum capacity of the vector gaussian broadcast channel and uplink-downlink duality," *IEEE Trans. Inf. Theory*, vol. 49, no. 8, pp. 1912–1921, Aug. 2003.

[80] N. Jindal, S. Vishwanath, and A. Goldsmith, "On the duality of gaussian multipleaccess and broadcast channels," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, p. 768C783, May 2004.

[81] S. Vishwanath, N. Jindal, and A. Goldsmith, "Duality, achievable rates, and sum-rate capacity of MIMO broadcast channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, p. 2658C2668, Oct. 2003.

[82] N. Jindal, W. Rhee, S. Vishwanath, S. A. Jafar, and A. Goldsmith, "Sum power iterative water-filling for multi-antenna gaussian broadcast channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 4, pp. 1570–1580, Apr. 2005.

[83] A. G. Marques, G. B. Giannakis, F. Digham, and F. J. Ramos, "Power-efficient wireless OFDMA using limited-rate feedback," *IEEE Trans. Wireless Commun.*, vol. 7, no. 2, p. 685C696, Feb. 2008.

[84] A. G. Marques, X. Wang, and G. B. Giannakis, "Optimal stochastic dual resource allocation for cognitive radios based on quantized CSI," in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, Las Vegas, USA, 2008.

[85] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, p. 1439C1451, Aug. 2006.

[86] ——, "Alternative distributed algorithms for network utility maximization: Framework and applications," *IEEE Transactions on Automatic Control*, vol. 52, no. 12, p. 2254C2269, Dec. 2007.

[87] X. Wang, G. B. Giannakis, and A. G. Marques, "A unified approach to qos-guaranteed scheduling for channel-adaptive wireless networks," in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, Proceedings of the IEEE, 2007.

[88] X. Wang and G. B. Giannakis, "Power-efficient resource allocation in time-division multiple access over fading channels," *IEEE Trans. Inf. Theory*, vol. 54, no. 3, pp. 1225–1240, Mar. 2008.

[89] A. Ribeiro and G. B. Giannakis, "Optimal fdma over wireless fading mobile ad-hoc networks," in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, Las Vegas, USA, 2008.

[90] H. Boche and M. Schubert, "Characterization of the structure of general interference functions," in *IEEE ISIT 2007*, 2007.

[91] Y. Lin, T. Javidi, R. Cruz, and L. Milstein, "Stochastic optimization and its applications in multihop wireless MIMO networks," *submitted to IEEE Transcation on Information Theory*, 2008.

[92] M. Kobayashi and G. Caire, "An iterative water-filling algorithm for maximum weighted sum-rate of gaussian MIMO-BC," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1640–1646, Aug. 2006.

[93] ——, "Joint beamforming and scheduling for a MIMO downlink with random arrivals," in *IEEE ISIT 2006*, Seatle, USA, 2006.

[94] T. Yoo and A. Goldsmith, "Capacity of fading MIMO channels with channel estimation error," in *IEEE Int. Commun. Conf. (ICC)*, Paris, France, Jun. 2004.

[95] J. Wang, S. Jin, Z. Liu, Y. Wang, and X. Yu, "Analysis of the sum capacity of MIMO broadcast systems with channel estimation errors," in *IEEE Global Telecommunications Conference*, 2006.

[96] N. Jindal, "MIMO broadcast channels with finite rate feedback," in *IEEE Global Commun. Conf. (GLOBECOM)*, St. Louis, USA, Nov. 2005.

[97] J. Roh and B. Rao, "Multiple antenna channels with partial channel state information at the transmitter," *IEEE Trans. Wireless Commun.*, vol. 3, no. 2, pp. 677–688, Mar. 2004.

[98] ——, "Design and analysis of MIMO spatial multiplexing systems with quantized feedback," *IEEE Trans. Signal Process.*, vol. 54, no. 8, pp. 2874–2886, Aug. 2006.

[99] J. Zheng, E. Duni, and B. Rao, "Analysis of multiple antenna systems with finite-rate feedback using high resolution quantization theory," 2007, to appear on IEEE Trans. on Sig. Processing.

[100] S. Srinivasa and S. Jafar, "Vector channel capacity with quantized feedback," *IEEE International Conference on Communications*, vol. 4, p. 2674 C 2678, May 2005.

[101] ——, "The optimality of beamforming: A unified view," *IEEE Global Communications Conference*, vol. 6, p. 3482 C 3486, 2005.

[102] S. Jafar and S. Srinivasa, "On the optimality of beamforming with quantized feedback," 2005, submitted to the IEEE Trans. on Commun.

[103] S. Srinivasa, S. Jafar, and S. Vishwanath, "Does beamforming achieve outage capacity with direction feedback?" in *Communication Theory Workshop*, 2005.

[104] P. Viswanath, D. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1277–1294, Jun. 2002.

[105] N. Kahale and P. Wright, "Dynamic global packet routing in wireless networks," in *Proc. IEEE INFOCOM'97*, May 1997.

[106] M. Neely, E. Modiano, and C. Rohrs, "Power allocation and routing in multibeam satellites with time-varying channels," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 138–152, Feb. 2003.

[107] H. Viswanathan and K. Kumaran, "Rate scheduling in multiple antenna downlink wireless systems," *IEEE Trans. Commun.*, vol. 53, no. 4, pp. 645–655, Apr. 2005.

[108] S. Kittipiyakul and T. Javidi, "A fresh look at optimal subcarrier allocation in OFDMA systems," in *IEEE CDC 2004*, 2004.

[109] H. Robbins and D. Siegmund, "A convergence theorem for non negative almost supermartingales and some applications," *Optimization methods in statistics*, pp. 233– 257, 1971.

[110] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel," in *ISSSE98*, Pisa, Italy, 1998.

[111] D. Tse and P. Viswanath.

[112] R. A. Horn and C. R. Johnson, *Matrix Analysis.* Cambridge Univ. Press, 1990.

[113] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York: Wiley, 1991.

[114] M. S. Bazaraa and C. M. Shetty, *Nonlinear Programming.* John Wiley & Sons, 1979.

[115] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.

[116] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[117] S. Lang, *Complex Analysis*, 4th ed. Springer, 1998.

[118] I. J. Bakelman, *Introduction to Complex Analysis*, 2nd ed. Oxford Univeristy Press, 2003.

[119] ——, *Convex Analysis and Nonlinear Geometric Elliptic Equations.* Springer, 1994.

[120] F. Morgan, *Geometric Measure Theory.* Academic Press, 1988.

[121] H. Pitt, *Measure And Integration for Use.* Clarendon Press, 1985.

[122] L. Evans and R. Gariepy, *Measure theory and fine properties of functions, Studies in Advanced Mathematics.* CRC Press, 1992.

[123] W. Rudin, *Principles of Mathematical Analysis.* New York, McGraw-Hill, 1964.

[124] R.C.Buck, *Advanced Calculus.* New York, McGraw-Hill, 1956.

[125] S.I.Resnick, *A Probability Path.* Birkhauser Boston, 1998.