UNIVERSITY OF CALIFORNIA SAN DIEGO

Adaptive Policies for Scheduling High-speed Elastic Networks Subject to Reconfiguration Overhead

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

in

Electrical Engineering
(Communication Theory and Systems)

by

Chang-Heng Wang

Committee in charge:

Professor Tara Javidi, Chair
Professor Massimo Franceschetti
Professor Bill Lin
Professor George Porter
Professor Ruth Williams

2019

The dissertation of Chang-Heng Wang is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2019

DEDICATION

To my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Tara Javidi, who has always been a patient and encouraging advisor. I really enjoyed my time working with her. I thank her for giving all the freedom and support for me to explore the directions I am interested in.

I would like to thank Ruth Williams, Massimo Franceschetti, George Porter, and Bill Lin for serving on my thesis committee and giving valuable comments and suggestions to my thesis. I would also like to thank all of my collaborators, George Porter, Siva Theja Maguluri, Antonia Maria Tulino, and Jaime Llorca. This thesis could not be completed without your help.

I thank my labmates, Sung-En Chiu, Yongxi Lu, Anusha Lalitha, Nancy Ronquillo, and Shekar Shubanshu. Despite that most of us work on diverse topics, the conversation with them are always interesting and inspiring.

The internship experiences during my PhD study were very pleasant and very beneficial to my skill development. I am grateful for my mentors and the companies that provide me with the opportunities. I thank Longson at MediaTek. Doing an internship in my hometown has been a remarkable experience for me. I thank Jaime and Antonia at Nokia Bell labs. I really enjoyed the work and leisure time I spent with them. I also thank my roommate and colleague Giuseppe and my landlord Sue for enriching my stay at New Jersey. I thank Onur, John, and all my colleagues and fellow interns at Toyota InfoTechnology Center. The year long internship had been very fruitful and was very ideal to combine theory and hands-on experiences.

My fellow Taiwanese graduate students have always been the greatest support during my Ph. D. journey. I especially thank the Taiwanese volleyball team, playing volleyball weekly had been an essential part of my life here at San Diego. I thank my old friend Shawn, who has been my classmate and very supportive friend for more than ten years. He was also the first person to welcome me at San Diego and provided plenty of help when I first arrived.

I would like to thank my fiance Wei-Tang for giving me endless love and support during the past few years. Meeting her is definitely the best thing happened during my time in San Diego.

Finally, and most importantly, I would like to thank my parents, my sister, and all my family for all the support throughout my entire life.

This work has been partially supported by L3 Communications, NSF Center for Integrated Access Networks (NSF Grant EEC-0812072), NSF grant CNS-1329819 Event-Based Information Acquisition, Learning, and Control in High-Dimensional Cyber-Physical Systems, and NSF grant CNS-1619129 Caching and Coding for Media Delivery in Next Generation Wireless Networks.

Chapter 3, in part, is a reprint of the material as it appears in the paper: C.-H. Wang, T. Javidi, G. Porter, ”End-to-end Scheduling for All-Optical Data Centers”, IEEE Conference on Computer Communications (INFOCOM), pp 406-414, 2015. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in part, is a reprint of the material as it appears in the paper: C.-H. Wang, T. Javidi, ”Adaptive Policies for Scheduling with Reconfiguration Delay: An End-to-end Solution for All-Optical Data Centers”, IEEE/ACM Transactions on Networking (TON), vol 25, pp 1555-1568, 2017. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in part, is a reprint of the material as it appears in the paper: C.-H. Wang, S. T. Maguluri, T. Javidi, “Heavy Traffic Queue Length Behavior in Switches with Reconfiguration Delay”, IEEE Conference on Computer Communications (INFOCOM), pp 1-9, 2017. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in full, is currently being prepared for submission for publication as: C.-H. Wang, S. T. Maguluri, T. Javidi, ”Toward Optimal Heavy Traffic Queue Length Behavior for Switches with Reconfiguration Delay.” The dissertation author was the primary investigator and author of this material.

Chapter 5, in full, is a reprint of the material as it appears in the paper: C.-H. Wang, J. Llorca, A. M. Tulino, T. Javidi, “Dynamic Cloud Network Control under Reconfiguration Delay and Cost”, IEEE/ACM Transactions on Networking (TON), vol 27, pp 491-504, 2019. The dissertation author was the primary investigator and author of this paper.

VITA

2011	Bachelor of Science, Electrical Engineering, National Taiwan University
2015	Master of Science, Electrical and Computer Engineering, University of California San Diego
2019	Doctor of Philosophy, Electrical and Computer Engineering, University of California San Diego

PUBLICATIONS

C.-H. Wang, J. Llorca, A. M. Tulino, T. Javidi, “Dynamic Cloud Network Control under Reconfiguration Delay and Cost”, *IEEE/ACM Transactions on Networking (TON)*, vol 27, pp 491-504, 2019.

W. Wang, C.-H. Wang, T. Javidi, “Reliable Shortest Path Routing with Applications to Wireless Software-Defined Networking”, *IEEE Global Communications Conference (GLOBECOM)*, pp 1-6, 2018.

C.-H. Wang, T. Javidi, “Adaptive Policies for Scheduling with Reconfiguration Delay: An End-to-end Solution for All-Optical Data Centers”, *IEEE/ACM Transactions on Networking (TON)*, vol 25, pp 1555-1568, 2017.

C.-H. Wang, S. T. Maguluri, T. Javidi, “Heavy Traffic Queue Length Behavior in Switches with Reconfiguration Delay”, *IEEE Conference on Computer Communications (INFOCOM)*, pp 1-9, 2017.

C.-H. Wang, T. Javidi, G. Porter, “End-to-end Scheduling for All-Optical Data Centers”, *IEEE Conference on Computer Communications (INFOCOM)*, pp 406-414, 2015.

T. Akta, C.-H. Wang, T. Javidi, “WiCOD: Wireless control plane serving an all-optical data center”, *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pp 299-306, 2015.

C.-H. Wang, C.-T. Chou, P. Lin, M. Guizani, “Performance evaluation of ieee 802.15. 4 nonbeacon-enabled mode for internet of vehicles”, *IEEE Transactions on Intelligent Transportation Systems*, vol 16, pp 3150-3159, 2015.

M.-C. Yang, C.-H. Wang *, T.-Y. Hu *, Y.-C. F. Wang, (* indicates equal contribution), “Learning context-aware sparse representation for single image super-resolution”, *IEEE International Conference on Image Processing (ICIP)*, pp 1349-1352, 2011.

ABSTRACT OF THE DISSERTATION

Adaptive Policies for Scheduling High-speed Elastic Networks Subject to Reconfiguration Overhead

by

Chang-Heng Wang

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

University of California San Diego, 2019

Professor Tara Javidi, Chair

A scheduling policy plays a critical role in achieving good performance for most networked systems. While the optimal scheduling problems have been studied to a great extent in the literature, the effect of reconfiguration overhead receives far less attention. With the advancement of high speed and highly flexible networking technologies, the reconfiguration overhead becomes more significant in the design of an efficient scheduling policy.

We first consider the scheduling problem in a generalized switch model subject to reconfiguration delay. We propose the Adaptive MaxWeight policy, and prove that the Adaptive

MaxWeight policy is throughput optimal. We also extend the idea of the Adaptive MaxWeight and propose a general method to transform scheduling policies that loses the throughput optimality under nonzero reconfiguration delay into adaptive policies that recover the throughput optimality guarantee.

We then further analyze the average queue length of the Adaptive MaxWeight policy in the heavy traffic regime. In particular, we consider a sequence of switch systems with arrival rates approaching to the boundary of the capacity region, where the limiting arrival rate has saturated traffic for all input ports and all output ports. In the heavy traffic regime, we derive an upper bound for the expected sum of queue length, which achieves the optimal scaling with respect to the traffic load as well as to the reconfiguration delay.

Finally, we consider the scheduling problem in a distributed computing network subject to reconfiguration delay and reconfiguration cost. We show that while the capacity region remains unchanged regardless of the reconfiguration delay/cost values, a reconfiguration-agnostic policy may fail to guarantee throughput-optimality and minimum cost under nonzero reconfiguration delay/cost. We then present the Adaptive Dynamic Cloud Network Control policy that allows network nodes to make local flow scheduling and resource allocation decisions while implicitly controlling the frequency of reconfiguration in order to support any input rate in the capacity region and achieve arbitrarily close to minimum cost for any finite reconfiguration delay/cost values.

Chapter 1

Introduction

Many important problems in telecommunications, computing, or manufacturing systems involve efficiently managing the contention for resources in a rapidly changing environment.

As technologies evolve to provide larger and larger throughput in communication and computing capabilities, the issue of reconfiguration overhead becomes more challenging than ever. For systems introducing novel technology to boost the bandwidth, but cannot reduce the reconfiguration overhead or even introduce larger overhead, this issue could no longer be neglected and could potentially have significant impact on the performance of the systems.

The evolution of the information technology also opens up the opportunity for system to quickly react to the dynamic fluctuation of the workload. The softwarization of resource management system enables the dynamic allocation of resources to elastically scale up or down services depending on the instantaneous workload. Similar to the reconfiguration overhead in the scheduling problems, the dynamic allocation of resources may also incur certain overhead, which needs to be taken into account in order to efficiently utilizing such flexibility.

While the optimal scheduling problems for several communication networks have been an active research field [31, 42, 43, 46], the effect of reconfiguration overhead on scheduling performance received far less attention. Moreover, prior work in the research mostly address the reconfiguration overhead in static sense, which does not react to dynamic workload variations, or some in quasi-static sense, which makes scheduling decisions based on potentially out-dated information.

In this thesis, we consider the impact of reconfiguration overheads such as reconfiguration delay and reconfiguration cost on scheduling problems.

1.1 Scheduling with Reconfiguration Overhead

1.1.1 Switch Model with Reconfiguration Delay

In the first part of the thesis, we consider a generalized switch model with the presence of reconfiguration delay. The application of the considered model includes optical networks, wireless networks, and satellite communications.

In recent years, the optical switch emerges as a promising candidate supporting the next generation network as it is transparent to data bandwidth and more power efficient than traditional electronic switches. However, the main challenge for optical switches is that optical networks typically exhibit a nonzero reconfiguration delay upon reconfiguring the switch schedule. For instance, candidate technologies such as microelectromechanical systems (MEMS) [47] involve mechanically directing laser beams and thus require a certain time to finish circuit reconfiguration due to physical limits. During the circuit reconfiguration, reliable packet transmission could not be supported in the network.

For wireless networks, as higher frequency bands are being adopted for larger data bandwidth, beamforming techniques are getting wide adoption to mitigate the increased attenuation. As a result, the reconfiguration delay becomes a significant factor, since the reconfiguration of the beam direction can take up to several hundred microseconds. Similarly, in satellite communications, mechanical steerable antennas are used to serve multiple ground stations. When a antenna steer from one station to another, it may incur a reconfiguration delay around few milliseconds.

1.1.2 Reconfiguration Delay and Cost in Distributed Computing Networks

In the second part of the thesis, we consider the scheduling of distributed computing networks with the flexibility of dynamic resource allocation, subject to reconfiguration overheads including reconfiguration delay and cost.

The emergence of network function virtualization (NFV) and software defined networking (SDN) enables network services to be deployed in the form of interconnected software functions instantiated over commercial off-the-shelf servers at multiple cloud locations and interconnected via a programmable network fabric. This allows cloud network operators to host a large variety of services over a common general purpose infrastructure and dynamically allocate resources according to changing demands, reducing both capital and operational expenses.

The unprecedented flexibility of the cloud networking paradigm provides exciting opportunities for future service scenarios and stimulates research in key technical areas such as optimal function placement, service flow routing, and joint cloud/network resource allocation. One line of research addressed the virtual network functions placement problem from a static global optimization point of view, in which the goal is to find the placement of virtual functions and the routing of network flows that meet service demands with minimum cost [3, 9, 15, 51].

1.2 Notation

In this section, we briefly introduce the common notation that would be used throughout this thesis.

- $\mathbb{N} = \{0, 1, 2, \dots\}$ denotes the set of natural numbers, or non-negative integers.
- $\mathbb{R}$ denotes the set of real numbers, and $\mathbb{R}_+$ denotes the set of non-negative real numbers.
- $\mathbb{E}[X]$ denotes the expected value of a random variable X.
- $\mathrm{Var}(X)$ denotes the variance of a random variable X.

- $|\mathcal{X}|$ denoted the number of elements in a set $\mathcal{X}$.

- $|x|$ denotes the absolute value of a number $x \in \mathbb{R}$.

- $\|\mathbf{X}\|_p$ denotes the l_p-norm of a vector $\mathbf{X}$. For a r-dimensional vector $\mathbf{X} \in \mathbb{R}^r$, the l_p-norm is defined as $\|\mathbf{X}\|_p = (\sum_{i=1}^{r} X_i^p)^{\frac{1}{p}}$.

- $\mathbb{1}_E$ denotes the indicator function for an event E.

- $[x]^+ = \max\{x, 0\}$ is a rectified linear function.

1.2.1 Queueing System

We now introduce a common framework for the problem formulations that are considered in this thesis. In the following chapters, we would specify how each system model fits in this common framework.

We consider a system that consists of a set of queues $\mathcal{Q}$, indexed by $q \in \mathcal{Q}$. The queueing system is assumed to be a discrete time system, where time is divided into consecutive fixed duration time slots, with each time slot indexed by $t \in \mathbb{N}$. Each queue in the system holds a specific type of workload awaiting to be serviced in a first-come-first-serve (FCFS) fashion. We use the term *packet* as a unit for the workload.

We let $Q_q(t)$ denote the number of packets in queue q at the beginning of time slot t. Within each time slot, there may be some packets arriving at each queue, we let $A_q(t)$ denote the number of packets arriving at queue q at time slot t.

Within each time slot, the queueing system may be set to provide service to a number of queues in $\mathcal{Q}$. We let $S_q(t)$ denote the number of packets that the system scheduled to serve for queue q at time t.

For each queue $q \in \mathcal{Q}$, the maximum number of packets that could be scheduled to be served at time t is the capacity allocated for queue q, which is dependent on the amount of resources allocated for queue q. We let $k_q(t) \in \{0, 1, \ldots, K_q\}$ denote the number of unit resources allocated for queue q at time t, where K_q is the maximum number of unit resources available for queue q and is a fixed value. The capacity allocated for queue q is a function of the amount of allocated resource, and is denoted as $C_q(k)$ where $k \in \{0, 1, \ldots, K_q\}$. The allocation of the resource also incurs an operational cost, where the cost of k units of resource assigned for queue q is $w_q(k)$ where $k \in \{0, 1, \ldots, K_q\}$. By the definition, we have

$$S_q(t) \leq C_q(k_q(t)), \qquad \forall q \in \mathcal{Q},\ \forall t \in \mathbb{N}. \tag{1.1}$$

Note that $S_q(t)$ does not necessarily represents the number of packets departing from queue q at time t if any one of the following two possible scenarios occurs:

1. The number of packets in queue q is less than $S_q(t)$.

2. Queue q is under reconfiguration, which we elaborate in the following paragraph.

We assume that after each schedule or resource reconfiguration for $q \in \mathcal{Q}$, packets in queue q could not be serviced for a fixed time duration. This time duration is referred as the reconfiguration delay of queue q, and is denoted as Δ_q. Let $\{t_k^{(q)}\}_{k=1}^{\infty} \subset \mathbb{N}$ denote the time instances when the schedule is reconfigured for queue q, then $\forall t \in \cup_{k=0}^{\infty}[t_k^{(q)}, t_k^{(q)} + \Delta_q]$, queue q is considered under reconfiguration and packets in queue q could not depart at time t even if $S_q(t) > 0$.

For ease of presentation, we let $r_q(t)$ denote the time for queue q remaining in the reconfiguration delay, with $r_q(t) = 0$ indicating that queue q is not in reconfiguration at time slot t. Formally speaking, we define $r_q(t) = [\Delta_q - (t - \max_k\{t_k^{(q)} : t_k^{(q)} \leq t\})]^+$. The number of packets that is actually scheduled to be serviced for queue q at time t is then given as

$$\bar{S}_q(t) = S_q(t) \mathbb{1}_{\{r_q(t)=0\}}. \tag{1.2}$$

We adopt the convention that the packet departure occurs at the end of each time slot, and the queue length of a queue q evolves according to the following queue length dynamics:

$$\begin{aligned} Q_q(t+1) =& [Q_q(t) - \bar{S}_q(t) + A_q(t)]^+ \quad (1.3) \\ =& Q_q(t) - S_q(t)\mathbb{1}_{\{r_q(t)=0\}} + A_q(t) + U_q(t) \quad (1.4) \end{aligned}$$

where $U_q(t)$ denotes the unused service of queue q at time t. By definition, the unused service is nonzero only when the number of packets in a queue is less than the number of packets that could depart.

$$Q_q(t+1)U_q(t) = 0, \qquad \forall q \in \mathcal{Q},\ \forall t \in \mathbb{N}. \quad (1.5)$$

1.3 Contributions

The main contribution of this thesis is solving the dynamic scheduling problems for queueing systems subject to reconfiguration overhead and dynamic workload. The scheduling problems covered include systems operated under the centralized scheduling framework as well as the distributed scheduling framework.

For the centralized scheduling framework, we consider the generalized switch model and solve the scheduling problem subject to reconfiguration delay. The contributions include the following:

- We develop the Adaptive MaxWeight (AMW) policy and prove the throughput optimality of the AMW policy.
- We generalize the AMW policy to a general scheme that transforms any scheduling policy to an adaptive policy to accommodate the reconfiguration delay. We show that under mild assumptions, for policies that are throughput optimal under no reconfiguration delay but

lose the throughput optimality due to reconfiguration delay, their transformed adaptive policies recover the throughput optimality even under the presence of reconfiguration delay.

- We characterize the effect of reconfiguration delay on the queue length performance by deriving a queue length lower bound of any scheduling policy for switches with reconfiguration delay.
- We prove that in the heavy traffic regime, where the arrival rate approaches to a all-port saturate matrix, the AMW policy approximates the optimal queue length scaling with respect to the traffic load as well as to the reconfiguration delay.

For the distributed scheduling framework, we consider a model consisting of a network of computing nodes where each node has communication and computation capabilities, and solve the joint scheduling and cost minimization problem subject to reconfiguration overheads. The contributions include the following:

- We prove that the capacity region and the achievable minimum time average cost remains the same even in the presence of reconfiguration delay and cost, given that reconfiguration delay and cost values are finite.
- We show that a reconfiguration-agnostic policy that is throughput optimal and achieves arbitrarily close to the minimum time average cost under zero reconfiguration overhead does not necessarily retain these properties when reconfiguration delay/cost exists.
- We develop the Adaptive Dynamic Cloud Network Control (ADCNC) policy and prove the throughput and cost optimality of the ADCNC policy, given the reconfiguration delays and costs are finite fixed values. The proposed ADCNC policy utilizes only queue length information and does not require any prior knowledge on arrival statistics.

1.4 Thesis Outline

We briefly introduce the layout of this thesis as follows. Chapter 1 has introduced the basic notation used throughout this thesis and the contributions of this work. Chapter 2 introduces some key assumptions that are common to all the models considered in this thesis, and briefly introduce the methodology to be used.

Chapter 3 considers the scheduling problem for switches with reconfiguration delay, and introduces Adaptive MaxWeight and its extension to a framework of adaptive policy construction that guarantees throughput optimality. Chapter 4 focuses on the delay performance analysis of Adaptive MaxWeight policy for switches with reconfiguration delay and shows that Adaptive MaxWeight policy may arbitrarily approximate the optimal queue length scaling in the heavy traffic regime.

Chapter 5 generalizes the framework to include reconfiguration cost as another type of reconfiguration overhead, and considers the scheduling problem with reconfiguration overhead in a distributed network setting.

Finally, chapter 6 concludes the thesis and introduces some potential future research direction.

Chapter 2

Model Assumptions and Methodology

In this chapter, we first state some basic assumptions that apply to all of the queueing systems considered in this thesis. We then briefly introduce the performance metrics considered for the scheduling policies of interest and the methodology used for the analysis in this thesis.

2.1 Model Assumptions

Throughout this thesis, we assume that the arrival processes satisfy the following assumptions. The first one assumes an upper bound on the maximum number of packet arrival within a time slot, which is a mild assumption for many practical systems.

Assumption 2.1. *The arrival process at each queue is bounded from above at any given time. In other words, there exists a constant $A_{\max} < \infty$ such that*

$$A_q(t) \leq A_{max}, \qquad \forall q \in \mathcal{Q},\ \forall t \tag{2.1}$$

The following two assumptions regard the independence of the arrival processes.

Assumption 2.2. *The arrival process at each queue is independent from the arrival process at another queue. That is, given any $q, q' \in \mathcal{Q}, q \neq q'$, then for any $n < \infty$ and any sequence $t_1, \dots, t_n \in \mathbb{N}$, the random vectors $(A_q(t_1), \dots, A_q(t_n))$ and $(A_{q'}(t_1), \dots, A_{q'}(t_n))$ are independent.*

Assumption 2.3. *The arrival process at each queue is independent and identically distributed (i.i.d.) over time. The mean of the arrival process is a finite constant, and is denoted as $\lambda_q = \mathbb{E}[A_q(0)] < \infty$.*

For queueing systems where the resource allocation may be dynamically scheduled, we make the following assumption which is generally true for most systems in practice.

Assumption 2.4. *For any queue $q \in \mathcal{Q}$, we assume that the capacity $C_q(k)$ and the cost $w_q(k)$ for queue q are strictly increasing with k, where k is the amount of resource assigned for queue q. In other words, for any k such that $0 \leq k \leq K_q - 1$, we have $C_q(k) < C_q(k+1)$ and $w_q(k) < w_q(k+1)$.*

The schedule of the network is determined by a *scheduling policy*. A scheduling policy determines the schedule at time slot t based on the history of queue lengths, schedule, and reconfiguration status, *i.e.* $\mathbf{H}(t) = \{\{\mathbf{Q}(\tau)\}_{\tau=0}^{t}, \{\mathbf{S}(\tau)\}_{\tau=0}^{t-1}, \{\mathbf{r}(\tau)\}_{\tau=0}^{t-1}\}$. In other words, a policy π maps the history $\mathbf{H}(t)$ to the scheduling decision $\mathbf{S}(t) \in \mathcal{S}$ at each time slot t. Note that this mapping may potentially be probabilistic, where the history $\mathbf{H}(t)$ maps to a distribution on $\mathcal{S}$ and the scheduling decision $\mathbf{S}(t)$ is a sample drawn from $\mathcal{S}$ according to such distribution.

In this thesis, we restrict our discussions only to Markov scheduling policies, where the schedule decision depends only on the current state.

Assumption 2.5. *A* ***Markov scheduling policy*** *determines the schedules at each time t, and each schedule $\mathbf{S}(t) \in \mathcal{S}$ depends on the history solely through the current state $X(t) = \Big(\mathbf{Q}(t), \mathbf{S}(t-1), \mathbf{r}(t-1)\Big)$.*

Restricting attention to this class of policies ensures the process $\{X(t)\}_{t=0}^{\infty}$ to be a Markov process, where $X(t) = \Big(\mathbf{Q}(t), \mathbf{S}(t-1), \mathbf{r}(t-1)\Big)$ is the system state at time t. In this case we denote the Markov scheduling policy as π and use the notation $\mathbf{\Pi}(t)$ to denote the choice of schedule generated by π given the state $X(t)$.

2.2 Performance Metrics

In this thesis, we consider the scheduling problem for queueing systems subject to reconfiguration overhead. The main performance metrics for the scheduling policies are as follows:

- Throughput
- Average delay

For systems that allow the reconfiguration of the resource allocation, we also consider the following performance metric:

- Average operational cost

2.2.1 Throughput Optimality

In the literature, throughput optimality is one of the most fundamental requirement for a scheduling policy of interest. In short, the throughput optimality guarantees a scheduling policy to stabilize any arrival traffic that could be stabilized by some scheduling policy. We make this notion formal with the following definitions.

To evaluate the throughput of a scheduling policy, we first introduce the notion of stability.

Definition 2.1. A queue $q \in \mathcal{Q}$ is **strongly stable** if its queue length process $Q_q(t)$ satisfies

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left[Q_q(\tau)\right] < \infty \tag{2.2}$$

Throughout this thesis, we consider the strong stability as defined in Definition 2.1. For a queueing system with a set of queues $\mathcal{Q}$, we say that **the queueing system is strongly stable** if each queue $q \in \mathcal{Q}$ is strongly stable.

With the notion of queueing system stability defined as above, we may then characterize the throughput of a scheduling policy by every arrival traffic that the policy could stabilize. In this thesis, we focus on scheduling policies that maximizes the throughput, or achieves throughput optimality, as defined below.

Definition 2.2. An arrival traffic $\mathbf{A}(t) = (A_1(t), \ldots, A_{|\mathcal{Q}|}(t))$ is **admissible** if there exists a scheduling policy under which the queueing system is strongly stable. For an admissible arrival traffic $\mathbf{A}(t)$, the corresponding arrival rate vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{|\mathcal{Q}|})$ is said to be **admissible**.

Definition 2.3. The **capacity region** is defined as the set of all admissible arrival rate vector and is denoted as $\boldsymbol{\Lambda}$.

Definition 2.4. A scheduling policy is **throughput optimal** if for any admissible arrival traffic, the queueing system is strongly stable under the operation of this scheduling policy.

2.2.2 Average Delay Analysis

While the throughput optimality provides a measure for the robustness of a scheduling policy, in practice it is also important to evaluate the delay performance of a scheduling policy. In this thesis, we consider the average delay performance, which by Little's law [26], is linearly proportional to the average queue length. In particular, we are interested in characterizing the expected sum of queue length under the steady state distribution:

$$\mathbb{E}\left[\sum_{q\in\mathcal{Q}}\bar{Q}_q\right] \tag{2.3}$$

where $\bar{Q}_q$ is the steady state distribution of $Q_q(t)$.

Unlike single queue systems, it is in general impossible to characterize a closed form for the expected queue length for queueing systems with scheduling constraints, e.g. generalized switch model. Therefore prior works usually approach the delay performance analysis by studying the expected queue length in asymptotic regimes with respect to either the number of queues in the system or the traffic load.

In chapter 4, we consider the queue length behavior of switches with reconfiguration delay in the heavy traffic regime, where the arrival rates approaches to the boundary of the capacity region.

Chapter 3

Switches with Reconfiguration Delay

3.1 System Model

Consider a $N \times N$ generalized switch model with N input ports and N output ports. Each input port may have packets destined for each of the output port. We assume a virtual output queue architecture where each input port maintains separate queues for packets destined for different output ports. Therefore, each input port $i \in \{1, 2, \ldots, N\}$ maintains N separate queues, which are denoted by (i, j), where $j \in \{1, 2, \ldots, N\}$. The queueing system then contains a total of N^2 queues, and $\mathcal{Q} = \{(i, j) : 1 \leq i \leq N, 1 \leq j \leq N\}$.

We consider a time-slotted system, with each time slot indexed as $t \in \mathbb{N}$. Each slot duration is the time required for the switch to transmit a single packet, which is assumed to be a fixed value. Let $A_{ij}(t)$ be the number of packets arrived at queue (i, j) at time t, and let $Q_{ij}(t)$ be the number of packets in queue (i, j) at the beginning of each time slot t. For ease of notation, we denote them in matrix notations as $\mathbf{A}(t) = [A_{ij}(t)], \mathbf{Q}(t) = [Q_{ij}(t)]$, and thus $\mathbf{A}(t), \mathbf{Q}(t) \in \mathbb{N}^{N \times N}, \forall t \in \mathbb{N}$.

We assume the arrival processes $A_{ij}(t)$ satisfy Assumptions 2.1-2.3. In other words, $A_{ij}(t)$ are independent over $i, j \in \{1, 2, \ldots, N\}$, $i \neq j$. Each process $A_{ij}(t)$ is i.i.d. over time slots. Let the mean of $A_{ij}(t)$ be the traffic rate $\lambda_{ij} = \mathbb{E}[A_{ij}(0)]$, and define the traffic rate matrix $\boldsymbol{\lambda} = [\lambda_{ij}] \in \mathbb{R}_{+}^{N\times N}$.

3.1.1 Schedules and Scheduling Policies

At each time slot, the switch may schedule to serve some queues (i, j) by connecting input port i to output port j and transmitting a packet from input i to output port j, where $1 \leq i, j \leq N$. We let $\mathbf{S}(t) \in \{0, 1\}^{N\times N}$ denote the schedule of the switch at time t, where $S_{ij}(t) = 1$ if queue (i, j) is being scheduled, and $S_{ij}(t) = 0$ if otherwise.

The set of feasible schedules is denoted as $\mathcal{S} \subseteq \{0, 1\}^{N\times N}$, where each schedule $\mathbf{S} \in \mathcal{S}$ indicates some queues that could be scheduled at the same time. The set of feasible schedules $\mathcal{S}$ is dependent on the physical constraints of the system. One basic and common constraint is that each input port can only transmit to one output port, and each output port can only receive from one input port, *i.e.* $\sum_i S_{ij}(t) \leq 1, \sum_j S_{ij}(t) \leq 1$. Under such constraint, we have $\mathcal{S} \subseteq \mathcal{P}$, where $\mathcal{P}$ is the set of all permutation matrices. While in a typical crossbar switch, we usually have $\mathcal{S} = \mathcal{P}$, in this chapter we allow a more generalized setting where $\mathcal{S}$ can be any subset of $\{0, 1\}^{N\times N}$.

Upon reconfiguring a schedule, the network incurs a reconfiguration delay, during which no packet could be transmitted. In this paper we focus on the effect of the reconfiguration delay on the performance of scheduling policies. We make this notion formal through the following two definitions:

Definition 3.1. Let $\{t_k^S\}_{k=1}^{\infty}$ denote the time instances when the schedule is reconfigured. The schedule between two schedule reconfiguration time instances remains the same, *i.e.*

$$\mathbf{S}(\tau) = \mathbf{S}(t_k^S), \ \ \forall \tau \in [t_k^S, t_{k+1}^S - 1]$$

Definition 3.2. Let Δ_r be the reconfiguration delay associated with reconfiguring the schedule of the network. During the period of schedule reconfiguration, no packet transmission could occur in the network. In other words, $\forall t \in \cup_{k=0}^{\infty}[t_k^S, t_k^S + \Delta_r]$, the switch does not serve any of its queues. We assume the reconfiguration delay to be an integer multiple of a time slot.

The schedule of the network is determined by a *scheduling policy*. In this paper, we restrict our discussions only to Markov scheduling policies defined as below.

Definition 3.3. A **Markov scheduling policy** determines the schedules at each time t, $\{\mathbf{S}(t)\}_{t=0}^{\infty}$, and each schedule $\mathbf{S}(t) \in \mathcal{S}$ depends on the history solely through the current state $X_t = \Big(\mathbf{S}(t-1), \mathbf{Q}(t)\Big)$. Restricting attention to this class of policies ensures the process $\{X_t\}_{t=0}^{\infty}$ to be a Markov process. In this case we denote the Markov scheduling policy as π and use the notation $\mathbf{\Pi}(t)$ to denote the choice of schedule generated by π given the state $X_t = \Big(\mathbf{S}(t-1), \mathbf{Q}(t)\Big)$.

A specific scheduling policy of interest here is the MaxWeight policy [46] defined below.

Definition 3.4. Given a schedule $\mathbf{S} \in \mathcal{S}$, the **weight** of the schedule $\mathbf{S}$ is a continuous, non-negative real function on the queue lengths, $W_{\mathbf{S}} : \mathbb{N}^{N \times N} \to \mathbb{R}_+$ with $W_{\mathbf{S}}(\mathbf{0}) = 0$. The **weight function** is defined as $W : \mathcal{S} \times \mathbb{N}^{N \times N} \to \mathbb{R}$ with $W(\mathbf{S}, \cdot) = W_{\mathbf{S}}(\cdot)$.

We also abuse the notation to let $W_{\mathbf{S}}(t) = W_{\mathbf{S}}(\mathbf{Q}(t))$ denote the weight of schedule S at time t whenever there is no confusion.

Definition 3.5. Given a weight function W, the **MaxWeight** policy determines the schedule $\mathbf{\Pi}^*(t)$ as the schedule that has the maximum weight among feasible schedules at time t, *i.e.*

$$\mathbf{\Pi}^*(t) = \arg\max_{\mathbf{S} \in \mathcal{S}} W_{\mathbf{S}}(t).$$

We also define the **maximum weight** at time t as the weight of the MaxWeight schedule at time t, *i.e.* $W^*(t) = \max_{\mathbf{S} \in \mathcal{S}} W_{\mathbf{S}}(t)$.

We define the traffic load as follows:

Definition 3.6. The **load** of traffic rate matrix $\boldsymbol{\lambda}$ is defined as

$$\rho(\boldsymbol{\lambda}) = \inf\{r : \boldsymbol{\lambda} \in r\bar{\boldsymbol{\Lambda}},\ 0 < r < 1\}$$

where $\bar{\boldsymbol{\Lambda}}$ is the closure of $\boldsymbol{\Lambda}$. We shall use ρ instead of $\rho(\boldsymbol{\lambda})$ whenever there is no confusion.

When $\Delta_r = 0$, several scheduling policies (e.g. [46], [45], [19]) have been shown to achieve throughput optimality in the literature. However, in the regime of $\Delta_r > 0$, these scheduling policies typically lose the throughput optimality guarantee since they do not address the fact that each schedule reconfiguration would result in a significant reduction in the duty cycle and hence decrease the utility of the network. The challenge for scheduling in the $\Delta_r > 0$ regime is that both the quality of the schedules and the rate of schedule reconfiguration affects the performance of a scheduling policy. It is not hard to see that there is a tradeoff between these two factors: Reducing the rate of reconfiguration means that the network is forced to stick with a schedule longer and loses the chance to use a better schedule; on the other hand, pursuing a better schedule most of the time inevitably increases the rate of reconfiguration and thus the incurred overhead. Therefore, a good scheduling policy in the $\Delta_r > 0$ regime must strive to achive the balance between these two factors.

3.2 Adaptive MaxWeight Policy

It is known that for switches without reconfiguration delay, the MaxWeight policy is throughput optimal [31] and has optimal delay scaling in the heavy traffic regime [28, 29]. However, with the presence of reconfiguration delay, the MaxWeight policy is not even throughput optimal since it does not account for the overhead of frequent schedule reconfiguration.

The main idea behind the Adaptive MaxWeight is to avoid excessive schedule reconfigurations, and only reconfigure the schedule when the current schedule is not “good” enough. Using the schedule weight as the measure of a schedule, the Adaptive MaxWeight computes the schedule weight difference between the current schedule and the MaxWeight schedule, W^* (which is the "best" schedule under this measure), and compares this weight difference to a threshold which is a function of the maximum weight, $g(W^*)$. When the schedule weight difference exceeds the threshold, we reconfigure the schedule to the MaxWeight schedule, otherwise keep the current schedule. A pseudo-code for the Adaptive MaxWeight scheduling policy is given in Algorithm 1.

Algorithm 1 Adaptive MaxWeight Scheduling Policy

```
Require: Sublinear and increasing function $g(\cdot)$
  for each $t = 0, 1, \ldots$ do
    $\mathbf{S}^*(t) \leftarrow \arg\max_{\mathbf{S}\in\mathcal{S}} \sum_{ij} Q_{ij}(t)S_{ij}$
    $W^*(t) \leftarrow \max_{\mathbf{S}\in\mathcal{S}} \sum_{ij} Q_{ij}(t)S_{ij}$
    $W(t) \leftarrow \sum_{ij} Q_{ij}(t)S_{ij}(t-1)$
    $\Delta W(t) \leftarrow W^*(t) - W(t)$
    if $\Delta W(t) > g(W^*(t))$ then
      $\mathbf{S}(t+1) \leftarrow \mathbf{S}^*(t)$
    else
      $\mathbf{S}(t+1) \leftarrow \mathbf{S}(t)$
    end if
  end for
```

3.2.1 Throughput Optimality of Adaptive MaxWeight

We first state the following lemma which is essential to the throughput optimality of Adaptive MaxWeight policy. Lemma 3.1 establishes a lower bound on the schedule duration that is dependent on the maximum schedule weight. The implication is that when the total queue length is larger, the frequency of reconfiguration becomes lower, which avoids excessive reconfiguration overhead.

Lemma 3.1. *Consider a crossbar switch operated under Adaptive MaxWeight policy with hysteresis function $g(\cdot)$, where $g(\cdot)$ is a sublinear and strictly increasing function. For any $T > 0$,* *define* $M = g^{-1}(N(A_{max}+1)T) + NT$*. Suppose a schedule reconfiguration occurs at time* t *and the queue lengths at time* t *satisfies* $W^*(t) > M$*, then no reconfiguration could occur in* $[t+1, t+T]$.

Proof. We prove this lemma by contradiction. Assume that the schedule reconfigures at some time slots within $[t+1, t+T]$, and let τ be the first such time slot. It then suffices to show that at time slot τ, the schedule weight difference $\Delta W(\tau) = W^*(\tau) - W(\tau)$ cannot exceed the threshold $g(W^*(\tau))$.

By the assumption, the schedule is reconfigured as $\mathbf{S}(t) = W^*(t)$ at time t, and maintains the schedule until time slot τ, hence $\mathbf{S}(\tau-1) = \mathbf{S}(t) = W^*(t)$.

Since at most one packet could depart at each queue in a time slot, we have a lower bound for the schedule weight $W(\tau)$:

$$\begin{aligned} W(\tau) &= \Big\langle \mathbf{Q}(\tau), \mathbf{S}(\tau-1) \Big\rangle = \Big\langle \mathbf{Q}(\tau), W^*(t) \Big\rangle \\ &\geq \Big\langle \mathbf{Q}(t), W^*(t) \Big\rangle - N(\tau - t) \\ &\geq W^*(t) - NT \end{aligned} \tag{3.1}$$

On the other hand, since the arrival at each queue is bounded by $A_{\max}$, we have an upper bound for the maximum weight $W^*(\tau)$:

$$\begin{aligned} W^*(\tau) &= \Big\langle \mathbf{Q}(\tau), S^*(\tau) \Big\rangle \\ &\leq \Big\langle \mathbf{Q}(t), S^*(\tau) \Big\rangle + NA_{\max}(\tau - t) \\ &\leq W^*(t) + NA_{\max}T \end{aligned} \tag{3.2}$$

From (3.1) and (3.2) we have the following bound on the weight difference:

$$\Delta W(\tau) = W^*(\tau) - W(\tau) \leq N(A_{\max}+1)T \tag{3.3}$$

On the other hand, by the assumption that $W^*(t) > M$, and since at most one packet could depart at each queue in a time slot, the maximum weight at time τ is lower bounded by:

$$\begin{aligned} W^*(\tau) &= \Big\langle \mathbf{Q}(\tau), \mathbf{S}^*(\tau) \Big\rangle \geq \Big\langle \mathbf{Q}(\tau), \mathbf{S}^*(t) \Big\rangle \\ &\geq \Big\langle \mathbf{Q}(t), \mathbf{S}^*(t) \Big\rangle - N(\tau - t) \\ &\geq W^*(t) - NT \\ &> M - NT \end{aligned} \tag{3.4}$$

From (3.3) and (3.4), and since $g(\cdot)$ is an increasing function, we have:

$$g\Big(W^*(\tau)\Big) > g(M - NT) \geq N(A_{\max} + 1)T \geq \Delta W(\tau) \tag{3.5}$$

which contradicts the assumption that a schedule reconfiguration occurs at time slot τ. We may then conclude that no schedule reconfiguration could occur in $[t+1, t+T]$.

□

With the upper bound on the frequency of schedule reconfiguration provided by Lemma 3.1, we are now ready to state and prove the throughput optimality of the Adaptive MaxWeight policy.

Theorem 3.1. *Given any reconfiguration delay $\Delta_r > 0$, and given any sublinear and increasing hysteresis function g, the Markov chain $\mathbf{X}(t)$ is positive recurrent for any admissible traffic rate matrix under the Adaptive MaxWeight. Therefore, the Adaptive MaxWeight is throughput optimal.*

Proof. Consider stopping times $t_k = kT$, where $T > \frac{\Delta_r}{1-\rho}$, and consider the Lyapunov function $V(\mathbf{Q}) = \Big\langle \mathbf{Q}, \mathbf{Q} \Big\rangle = \sum_{i,j} Q_{ij}^2$. Denote $\mathbf{\Delta}(t) = \mathbf{A}(t) - \mathbf{S}(t)\mathbb{1}_{\{r(t)=0\}}$, and we may write the expected drift of the Lyapunov function as

$$\begin{aligned}
&\mathbb{E}\left[V\left(\mathbf{Q}(t_{k+1})\right)-V(\mathbf{Q}(t_k))|\mathbf{Q}(t_k)\right]\\
=&\sum_{t=t_k}^{t_{k+1}-1}\mathbb{E}\left[V\left(\mathbf{Q}(t+1)\right)-V(\mathbf{Q}(t))|\mathbf{Q}(t_k)\right]\\
=&\sum_{t=t_k}^{t_{k+1}-1}\mathbb{E}\left[\Big\langle\mathbf{Q}(t+1),\mathbf{Q}(t+1)\Big\rangle-\Big\langle\mathbf{Q}(t),\mathbf{Q}(t)\Big\rangle\Big|\mathbf{Q}(t_k)\right]\\
=&\sum_{t=t_k}^{t_{k+1}-1}\mathbb{E}\left[2\Big\langle\mathbf{Q}(t),\mathbf{\Delta}(t)\Big\rangle+\Big\langle\mathbf{\Delta}(t),\mathbf{\Delta}(t)\Big\rangle\Big|L(t_k)\right] \qquad (3.6)
\end{aligned}$$

By the assumption of finite support of the arrival process, we have $\Delta_{ij}(t)\leq A_{\max}$ for any t, and thus $\Big\langle\mathbf{\Delta}(t),\mathbf{\Delta}(t)\Big\rangle\leq N^2A_{\max}^2$.

It now remains to bound the first term in (3.6).

$$\begin{aligned}
\mathbb{E}\left[\Big\langle\mathbf{Q}(t),\mathbf{\Delta}(t)\Big\rangle\Big|\mathbf{Q}(t_k)\right]=&\mathbb{E}\left[\Big\langle\mathbf{Q}(t),\mathbf{A}(t)-\mathbf{S}(t)\mathbb{1}_{\{r(t)=0\}}\Big\rangle\Big|\mathbf{Q}(t_k)\right]\\
=&\Big\langle\mathbf{Q}(t),\boldsymbol{\lambda}\Big\rangle-\mathbb{E}\left[\Big\langle\mathbf{Q}(t),\mathbf{S}(t)\Big\rangle\Big|\mathbf{Q}(t_k)\right]\\
&+\mathbb{E}\left[\Big\langle\mathbf{Q}(t),\mathbf{S}(t)\Big\rangle\mathbb{1}_{\{r(t)>0\}}\Big|\mathbf{Q}(t_k)\right]\\
=&\mathbb{E}\left[\Big\langle\mathbf{Q}(t),\boldsymbol{\lambda}-\mathbf{S}^*(t)\Big\rangle+\Big\langle\mathbf{Q}(t),\mathbf{S}^*(t)-\mathbf{S}(t)\Big\rangle\Big|\mathbf{Q}(t_k)\right]\\
&+\mathbb{E}\left[\Big\langle\mathbf{Q}(t),\mathbf{S}(t)\Big\rangle\mathbb{1}_{\{r(t)>0\}}\Big|\mathbf{Q}(t_k)\right] \qquad (3.7)
\end{aligned}$$

where $\mathbf{S}^*(t)$ is the MaxWeight schedule at time t. By the definition of the traffic load ρ, we may write $\boldsymbol{\lambda}=\rho\sum_{i=1}^{I}\mathbf{P}_i$, where $\mathbf{P}_i\in\mathcal{S}$ for $i=1,\ldots,I$. Then by the definition of the MaxWeight schedule, we have

$$\Big\langle\mathbf{Q}(t),\boldsymbol{\lambda}-\mathbf{S}^*(t)\Big\rangle\leq-(1-\rho)W^*(t) \qquad (3.8)$$

Also by the definition of the Adaptive MaxWeight policy, we have

$$\Big\langle \mathbf{Q}(t), \mathbf{S}^*(t) - \mathbf{S}(t) \Big\rangle = W^*(t) - W(t) \leq g(W^*(t)) \tag{3.9}$$

Apply (3.8) and (3.9) into (3.7), and use $W(t) \leq W^*(t)$, we then obtain

$$\begin{aligned} \mathbb{E}\left[\Big\langle \mathbf{Q}(t), \mathbf{\Delta}(t) \Big\rangle \Big| \mathbf{L}(t_k)\right] \leq & \mathbb{E}\left[-(1-\rho)W^*(t) + g(W^*(t)) \Big| \mathbf{L}(t_k)\right] \\ & + \mathbb{E}\left[W^*(t)\mathbb{1}_{\{r(t)>0\}} \Big| \mathbf{L}(t_k)\right] \end{aligned} \tag{3.10}$$

For an interval $[t_k, t_{k+1}]$ where $\mathbf{Q}(t_k)$ satisfies $W^*(t_k) > M + NT$, Lemma 3.1 implies that the schedule reconfiguration may occur at most once in the interval $[t_k, t_{k+1}]$. This is because if a schedule reconfiguration occurs at $\tau \in [t_k, t_{k+1}]$, we have that $W^*(\tau) \geq W^*(t_k) - NT > M$, and hence no reconfiguration may occur in $[\tau+1, t_{k+1}]$ by Lemma 3.1. Thus if $W^*(t_k) > M+NT$, then we have

$$\sum_{t=t_k}^{t_{k+1}-1} \mathbb{1}_{\{r(t)>0\}} \leq \Delta_r \tag{3.11}$$

We thus have $\forall \mathbf{Q}(t_k) : W^*(t_k) > M$,

$$\begin{aligned} & \mathbb{E}\left[V(\mathbf{Q}(t_{k+1})) - V(\mathbf{Q}(t_k)) \Big| \mathbf{Q}(t_k)\right] \\ \leq & \sum_{t=t_k}^{t_{k+1}-1} \mathbb{E}\left[-2(1-\rho)W^*(t) + 2g\Big(W^*(t)\Big) + 2W^*(t)\mathbb{1}_{\{r(t)>0\}} \Big| \mathbf{L}(t_k)\right] + TN^2A_{\max}^2 \\ \leq & -2T(1-\rho)\left(W^*(t_k) - NT\right) + 2Tg\left(W^*(t_k) + NA_{\max}T\right) \\ & + 2\Delta_r\left(W^*(t_k) + NA_{\max}T\right) + TN^2A_{\max}^2 \\ = & -2T(1-\rho-\frac{\Delta_r}{T})W^*(t_k) + 2Tg\Big(W^*(t_k) + NA_{\max}T\Big) \\ & + 2T\Big((1-\rho)NT + NA_{\max}\Delta_r\Big) + TN^2A_{\max}^2 \end{aligned} \tag{3.12}$$

Now since $g(\cdot)$ is a sublinear function, we get a negative Lyapunov drift when $W^*(t_k)$ is large. Specifically, let M' satisfies that $(1-\rho-\frac{\Delta_r}{T})M' = 2g(M'+NA_{\max}T)+2\Big((1-\rho)NT+NA_{\max}\Delta_r\Big)+N^2A_{\max}^2$ and let $B=\max\{M,M'\}$, we have $\forall \mathbf{Q}(t_k): W^*(t_k) > B$,

$$\mathbb{E}\left[V(\mathbf{Q}(t_{k+1}))-V(\mathbf{Q}(t_k))\Big|\mathbf{Q}(t_k)\right] \leq -T(1-\rho-\frac{\Delta_r}{T})W^*(t_k) \tag{3.13}$$

Since $\sum_{ij} Q_{ij}(t) \geq W^*(t) \geq \frac{1}{N}\sum_{ij} Q_{ij}(t)$, we have $\forall \mathbf{Q}(t_k): \sum_{ij} Q_{ij}(t_k) > NB$,

$$\mathbb{E}\left[V(\mathbf{Q}(t_{k+1}))-V(\mathbf{Q}(t_k))\Big|\mathbf{Q}(t_k)\right] \leq -\frac{T}{N}(1-\rho-\frac{\Delta_r}{T})\sum_{ij} Q_{ij}(t_k) \tag{3.14}$$

and by Fact 1, (3.13) implies that for any $\rho < 1$, the DTMC describing the queue length evolution is positive recurrent under the Adaptive MaxWeight policy, and the queue lengths satisfy

$$\lim_{k\to\infty}\mathbb{E}\left[||\mathbf{L}(t_k)||\right] < \infty. \tag{3.15}$$

□

3.2.2 Queue Dynamics under Adaptive MaxWeight

Fig. 3.1 illustrates the sample paths of the total queue length, the maximum weight, and the schedule weight of a network under the Adaptive MaxWeight policy. The green vertical lines mark the times of schedule reconfiguration, t_k^S, and the schedule weight is considered as zero during the reconfiguration delay (time period of length Δ_r following each schedule reconfiguration instance t_k^S). We first observe that the schedule duration increases as the total queue length increases. This is an essential component in achieving the network stability: let T be the mean schedule duration, then in order to ensure stability under $\Delta_r > 0$, the rate of schedule reconfiguration must satisfy $1-\frac{\Delta_r}{T} > \rho$. The schedule duration increases with the queue length until this condition is satisfied.

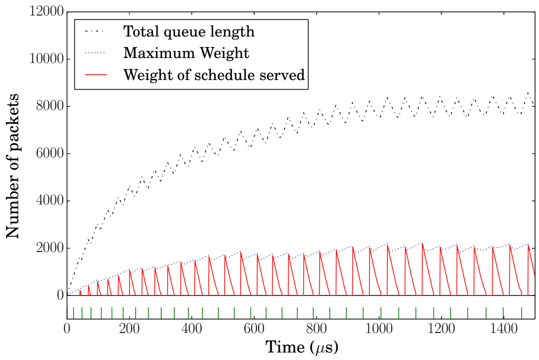

Figure 3.1: Sample paths of the total queue length, maximum weight, and the schedule weight, under the Adaptive MaxWeight policy. The number of ports is $N = 8$, the traffic load is $\rho = 0.6$, and the parameters of the AMW policy are $\delta = 0.01, \gamma = 0.1$. The green lines mark the schedule reconfiguration time instances t_k^S.

A somewhat more interesting observation is in the mechanism of this schedule duration adjustment. Recall that in the construction of Adaptive MaxWeight policy, it requires no explicit adjustment of the schedule duration. In other words, the duration of a schedule is not explicitly set at the time it is reconfigured. Instead, through the schedule determination by weight comparison at each time slot, the schedule duration is implicitly "adapted" to the appropriate value. After each schedule reconfiguration, the schedule weight begins from the maximum weight and decreases as the network is serving the queues associate with the schedule. When the total queue length is larger (and accordingly the maximum weight is larger), the threshold is larger and it takes longer for the weight difference between the maximum weight and the schedule weight to surpass the threshold. Recall that this property is characterized by lemma 3.1 shown in the previous subsection.

Note that this adaptive mechanism differs from other scheduling policies in the literature that explicitly adjust the schedule duration ([39], [37], [6]). We give a brief introduction of these policies and categorize them based on this schedule provisioning behavior in section 3.4. The performance comparison of these policies is then evaluated through simulations and presented in the section 3.6.

3.3 A Generalized Class of Adaptive Policies

3.3.1 g-Adaptive Variants of Scheduling Policies

Given a Markov scheduling policy π and current state X_t, let $\mathbf{\Pi}(t) = \pi(X_t)$ denote the schedule generated by π at time t. With the weight function W, let $W^{\pi}(t) = W_{\mathbf{\Pi}(t)}(t)$ denote the weight of the schedule $\mathbf{\Pi}(t)$ at time t.

At any given time t, $\Delta W(t) = W^{\pi}(t) - W_{\mathbf{S}(t-1)}(t)$ measures the potential improvement (in terms of schedule weight) associated with following policy π instead of sticking with the previous schedule $\mathbf{S}(t-1)$. Since each schedule change results in a loss in duty cycle, the proposed class of adaptive policies show some inertia against frequent schedule reconfiguration. More precisely, let us define a **hysteresis function** g where $g : \mathbb{R} \rightarrow \mathbb{R}$ is a nonnegative, continuous, strictly increasing, and sublinear (*i.e.* $\lim\limits_{x\rightarrow\infty} \frac{g(x)}{x} = 0$) function. Our proposed g-adaptive Markov policy π^g uses the new schedule $\mathbf{\Pi}(t)$ only if $\Delta W(t) > g(W^{\pi}(t))$. In other words,

$$\begin{aligned} \mathbf{\Pi}^g(t) &= \pi^g\Big(\mathbf{S}(t-1), \mathbf{Q}(t)\Big) \\ &= \begin{cases} \mathbf{S}(t-1) & \text{if } \Delta W(t) \leq g(W^{\pi}(t)) \\ \mathbf{\Pi}(t) & \text{if } \Delta W(t) > g(W^{\pi}(t)) \end{cases} \end{aligned} \tag{3.16}$$

Note that the proposed g-adaptive Markov policy could be constructed from any Markov scheduling policy. Given the Markov policy π, we call π^g the g-adaptive variant of π.

The intuition behind this construction is that the g-adaptive policy holds on to the previous schedule as long as it is "good enough" relative to the current schedule generated by π, $\Pi(t)$. The sublinearity of the function $g(\cdot)$ is a technical assumption in order to achieve the throughput optimality, which would become clear in the analysis given in the next subsection.

We now give some examples for possible combinations of hysteresis function $g(\cdot)$ and scheduling policy π.

Example 3.1. (g-adaptive variant of the Pipelined MaxWeight policy)

Under the pipelined MaxWeight policy [41], the scheduler determines the schedule at time t to be the schedule maximizing the weight at time $t - K$, for some fixed scalar $K < \infty$. Intuitively, one can think of the pipelined MaxWeight as a scheduler that initiates MaxWeight computation at each time slot but obtains and enforces it only K time slots later. Therefore, the schedule at time t is the MaxWeight schedule based on $\mathbf{Q}(t - K)$. The selection of the function $g(\cdot)$ could be any continuous, strictly increasing, and sublinear function, e.g. $g(x) = \log(1 + x)$.

In the following three examples, we introduce scheduling policies that are MaxWeight policies with respect to different definitions of the weight function W. For ease of presentation, we consider the hysteresis function to be $g(x) = (1 - \gamma)x^{1-\delta}$.

Example 3.2. (g-adaptive variant of the Projective Cone Scheduling policy)
In [40], the weight function W takes the form $W_{\mathbf{S}}(\mathbf{Q}) = \Big\langle vec(\mathbf{S}), \mathbf{P}vec(\mathbf{Q})\Big\rangle$ where $P \in \mathbb{R}^{N^2 \times N^2}$ and $vec(\cdot)$ is the vectorization of a matrix. The MaxWeight policy corresponding to this weight function W is called the Projective Cone Scheduling (PCS) policy. It was shown that if $\mathbf{P}$ is positive definite, symmetric, and has non-positive off-diagonal elements, then the PCS policy is throughput optimal.

Example 3.3. (g-adaptive variant of the MaxWeight-α policy)

For any fixed $\alpha > 0$, let the weight function be $W_{\mathbf{S}}(\mathbf{Q}) = \sum_{i,j=1}^{N} Q_{ij}^{\alpha}(t)S_{ij}(t)$, then the corresponding MaxWeight policy is referred as the MaxWeight-α policy in the literature. Under the MaxWeight-α policy, the schedule at time t is given by

$$\mathbf{\Pi}_{MW\alpha}(t) = \arg\max_{\mathbf{S} \in \mathcal{S}} \sum_{i,j=1}^{N} Q_{ij}^{\alpha}(t)S_{ij}(t) \tag{3.17}$$

Since the MaxWeight scheduling policy is well known for its high computation complexity, in the literature several lower complexity policies have also been proposed with good stability conditions when $\Delta_r = 0$. We consider the adaptive variant of these policies as well.

Example 3.4. (g-adaptive variant of Tassiulas random policy)

The Tassiulas Random policy [45] utilizes random schedule selection and memory to determine the schedule. It compares the weight between the last schedule and a randomly selected schedule (according to an arbitrary distribution on the feasible schedule $\mathcal{S}$, say uniformly random). Let $\mathbf{Z}(t)$ be the randomly selected schedule at time t, then the schedule determined by the Tassiulas random policy is given by

$$\mathbf{\Pi}_Z(t) = \begin{cases} \mathbf{\Pi}_Z(t-1) & \text{if } W_{\mathbf{\Pi}_Z(t-1)}(t) \geq W_{\mathbf{Z}(t)}(t) \\ \mathbf{Z}(t) & \text{otherwise} \end{cases} \tag{3.18}$$

Example 3.5. (g-adaptive variant of the Hamiltonian policy)

The Hamiltonian policy (ALGO 3 in [19]) utilizes the Hamiltonian walk on the set of permutation matrices and memory to determine the schedule. It compares the schedule weight between the last schedule and the schedule on the Hamiltonian path and select the schedule with higher weight. Specifically, let $\mathbf{H}(t)$ be the schedule on the Hamiltonian path at time t, then the schedule determined by the Hamiltonian policy is given by

$$\mathbf{\Pi}_H(t) = \begin{cases} \mathbf{\Pi}_H(t-1) & \text{if } W_{\mathbf{\Pi}_H(t-1)}(t) \geq W_{\mathbf{H}(t)}(t) \\ \mathbf{H}(t) & \text{otherwise} \end{cases} \tag{3.19}$$

3.3.2 Conditions for Throughput Optimal g-Adaptive Variants

Before we begin the analysis of the proposed adaptive policies, we state the following assumptions. For the weight function, we first focus only on Lipshitz continuous weight functions, and consider more general cases in later discussions:

Assumption 3.1. *Assume the weight function $W : \mathcal{S} \times \mathbb{R}^{N^2} \to \mathbb{R}$ satisfies that, for each schedule $\mathbf{S} \in \mathcal{S}$, the weight of the schedule, $W_{\mathbf{S}} : \mathbb{R}^{N^2} \to \mathbb{R}$, is Lipschitz continuous* [1].

We utilize the Foster-Lyapunov Theorem (cf. Fact 1 in Appendix) to show the throughput optimality of the adaptive policies. Let $V : \mathbb{R}^{N \times N} \to \mathbb{R}$ be a non-negative, real-valued Lyapunov function, and define the drift of V at time t as $\Delta V(t) = V(\mathbf{Q}(t+1)) - V(\mathbf{Q}(t))$. The general procedure of establishing the throughput optimality is to show that the conditional Lyapunov drift $\mathbb{E}\left[\Delta V(t) | \mathbf{Q}(t)\right]$ to be negative in all but a finite subset of the states $\mathbf{Q}(t)$. The following assumption characterizes the Lyapunov functions (with respect to a weight function) used for throughput analysis of most of the scheduling policies under zero reconfiguration delay.

Assumption 3.2. *Given the weight function W, the non-negative, real-valued Lyapunov function $V : \mathbb{R}^{N \times N} \to \mathbb{R}$ satisfies that, under the schedule $\mathbf{S}$, the expected Lyapunov drift satisfies:*

$$\mathbb{E}\left[\Delta V(t) \middle| \mathbf{Q}(t), \mathbf{S}(t) = \mathbf{S}\right] \leq \Lambda(\mathbf{Q}(t)) - W_{\mathbf{S}}(\mathbf{Q}(t)) \tag{3.20}$$

where $\Lambda(\cdot)$ is a term that is not dependent on $\mathbf{S}$.

Furthermore, suppose the schedule $\mathbf{S}$ is the MaxWeight schedule $\mathbf{\Pi}^*(t)$, then

$$\Lambda(\mathbf{Q}(t)) - W^*(t) \leq -\epsilon W^*(t) + K \tag{3.21}$$

for some fixed constant $\epsilon, K > 0$.

[1] Given X, Y metric spaces with corresponding metric d_X, d_Y, a function $F : X \to Y$ is Lipschitz continuous if there exists a real constant $B \geq 0$ such that $d_Y(F(x_1), F(x_2)) \leq B d_X(x_1, x_2)$.

Note that (3.20) determines an upper bound on the conditional Lyapunov drift which depends on the schedule $\mathbf{S}$ only through the weight function $W_{\mathbf{S}}$, and (3.21) is needed to establish the negative drift. We use the MaxWeight policy as an example to illustrate Assumption 3.2. Take the weight function as $W_{\mathbf{S}}(\mathbf{Q}) = \langle \mathbf{S}, \mathbf{Q} \rangle$ and the Lyapunov function as $V(\mathbf{Q}) = \langle \mathbf{Q}, \mathbf{Q} \rangle$. We have the conditional Lyapunov drift given by

$$\mathbb{E}\left[\Delta V(t) \middle| \mathbf{Q}(t)\right] \leq \langle \boldsymbol{\lambda}, \mathbf{Q}(t) \rangle - W_{\mathbf{S}}(t) + N^2 A_{\max}^2, \tag{3.22}$$

hence we have $\Lambda(\mathbf{Q}(t)) = \langle \boldsymbol{\lambda}, \mathbf{Q}(t) \rangle + N^2 A_{\max}^2$. Also, with $\mathbf{S} = \mathbf{\Pi}^*(t)$, we have $\langle \boldsymbol{\lambda}, \mathbf{Q} \rangle - W^*(t) \leq -(1-\rho) W^*(t)$. Hence

$$\Lambda(\mathbf{Q}(t)) - W^*(t) \leq -(1-\rho) W^*(t) + N^2 A_{\max}^2 \tag{3.23}$$

Remark. Note that the conditional Lyapunov drift expression in Assumption 3.2 is evaluated under zero reconfiguration delay. In the case of nonzero reconfiguration delay and the network is in reconfiguration at time t, we may simply evaluate the conditional drift in (3.20) with $W_{\mathbf{S}}(\mathbf{Q}(t)) = 0$, indicating the schedule is not serving the network.

With the previous assumptions, we now introduce a class of scheduling policies that guarantee bounded expected weight differences to the MaxWeight schedule at all times, and show the throughput optimality of their g-adaptive variant.

Condition 3.1. *Given the weight function W, the scheduling policy π satisfies the following property:*

There exists a constant $G < \infty$ such that the schedule weight of the scheduling policy π, $W^\pi(t) = W_{\mathbf{\Pi}(t)}(t)$, satisfies

$$\mathbb{E}^\pi \left[W^\pi(t) | \mathbf{Q}(t) \right] \geq W^*(t) - G, \ \ \forall t \tag{3.24}$$

Theorem 3.2. *Given any reconfiguration delay $\Delta_r > 0$. Assume the arrival traffic is admissible and satisfies Assumptions 2.1- 2.3. Assume the weight function W satisfies Assumption 3.1 and there exists a non-negative, real-valued Lyapunov function $V : \mathbb{R}^{N \times N} \to \mathbb{R}$ that satisfies Assumption 3.2 with weight function W.*

Suppose the Markov policy π satisfies Condition 3.1 with weight function W, then the g-adaptive variant of π is throughput optimal.

The proof of Theorem 3.2 is given in section 3.8. The proof utilizes the Foster-Lyapunov Theorem and consists of two main components. The first one shows that the schedule of the adaptive policy π^g has weight difference to the MaxWeight schedule bounded by a sublinear function of the maximum weight. This potentially gives the negative Lyapunov drift. The second part is that the rate of schedule reconfiguration becomes smaller as the queue lengths become larger. With this property, we show that the overhead incurred by the schedule reconfiguration delay becomes arbitrarily small when the total queue lengths $\sum_{ij} Q_{ij}(t)$ increases. This suffices to give the guarantee of negative expected drift when $\sum_{ij} Q_{ij}(t)$ is large and thus guarantee the stability. With Theorem 3.2, we are now ready to show the throughput optimality of some example adaptive policies given in the previous subsection.

Corollary 3.1. *The adaptive policies in examples 3.1 and 3.2 are throughput optimal.*

Proof. With the example weight function given for Assumption 3.1 and 3.2, we have these assumptions satisfied.

For example 3.1, Condition 3.1 is satisfied with $G = N(A_{\max} + 1)K$. To see this, recall that $W^{\pi}(t) = W_{\mathbf{\Pi}(t)}(\mathbf{L}(t)) = \left\langle \mathbf{L}(t), \mathbf{\Pi}(t) \right\rangle$, and note that at most one packet could depart from each queue at each time slot, and hence

$$\begin{aligned}\Big\langle \mathbf{L}(t), \mathbf{\Pi}(t) \Big\rangle &\geq \Big\langle \mathbf{L}(t-K), \mathbf{\Pi}(t) \Big\rangle - NK \\ &= \Big\langle \mathbf{L}(t-K), \mathbf{\Pi}^*(t-K) \Big\rangle - NK. \end{aligned} \tag{3.25}$$

Since the arrivals are bounded by $A_{\max}$, we also have

$$\Big\langle \mathbf{L}(t), \mathbf{\Pi}^*(t) \Big\rangle \leq \Big\langle \mathbf{L}(t-K), \mathbf{\Pi}^*(t-K) \Big\rangle + NA_{\max}K. \tag{3.26}$$

From (3.25) and (3.26), $W^\pi(t) \geq W^*(t) - N(A_{\max}+1)K$ and thus Condition 1 is satisfied. □

Corollary 3.2. *The g-adaptive variant of the Hamiltonian policy given in example 3.5 and the g-adaptive variant of the Tassiulas random policy given in example 3.4 are throughput optimal.*

Proof. In [19] the Hamiltonian policy is shown to satisfy Condition 3.1 with $G = 2N(N!)$, hence by Theorem 3.2 it achieves throughput optimality.

For the Adaptive Tassiulas policy, let $\Pr\{\mathbf{Z}(t) = \mathbf{\Pi}^*(t)\} \geq \epsilon > 0$ for all t and for some ϵ. In [22], the Tassiulas random policy is shown to satisfy Condition 3.1 with $G = 2N\frac{1-\epsilon}{\epsilon}$, hence by Theorem 3.2 it achieves throughput optimality. □

The throughput optimality guarantees could also be extended to some policies with non-Lipshitz continuous weight functions. In particular, we consider weight functions in the following form:

Assumption 3.3. *Assume the weight function W is defined as*

$$W_{\mathbf{S}}(\mathbf{Q}) = \sum_{i,j=1}^{N} f(Q_{ij})S_{ij} \tag{3.27}$$

for some continuous, non-decreasing function f with $f(0) = 0$.

For the ease of exposition, we restrict the policies of interest to those that have strictly bounded weight difference to the MaxWeight schedule (rather than just expected weight difference being bounded).

Condition 3.2. *Given the weight function W, the scheduling policy π satisfies the following property:*

There exists a constant $G < \infty$ such that the schedule weight of the scheduling policy π, $W^{\pi}(t) = W_{\mathbf{\Pi}(t)}(t)$, satisfies

$$W^{\pi}(t) \geq W^{*}(t) - G, \ \ \forall t \tag{3.28}$$

Theorem 3.3. *Given any reconfiguration delay $\Delta_r > 0$. Assume the arrival traffic is admissible and satisfies Assumptions 2.1- 2.3. Assume the weight function W satisfies Assumption 3.3 with function f. Suppose the scheduling policy π satisfies Condition 3.2 under the weight function W. If the sublinear function $g(\cdot)$ satisfies $\lim\limits_{x \to \infty} \frac{f'(x)}{g(f(x))} = 0$, then the g-adaptive variant of π achieves throughput optimality.*

Corollary 3.3. *Given $\alpha > 0$, and suppose that the sublinear function g satisfies $\lim\limits_{x \to \infty} \frac{x^{\alpha - 1}}{g(x^{\alpha})} = 0$, then the g-adaptive variant of MaxWeight-α is throughput optimal.*

3.4 Benchmark Policies

For scheduling policies accounting for nonzero reconfiguration delay, we classify them into two categories: “quasi-static scheduling” and “dynamic scheduling.” Quasi-static scheduling policies, also refered to as “batch scheduling,” [39] select a series of schedules based on a single schedule computation process, as shown in Figure 3.2 (a). We argue that under these policies, the generated schedules may depend on very out-dated information, especially for schedules employed later in a batch. This is the source of significant performance degradation. In contrast, under dynamic scheduling policies, each schedule is generated based on the most up-to-date queue information, as shown in Figure 3.2 (b). We now describe several example policies for each class and discuss their performance.

3.4.1 Quasi-static Scheduling Policies

In [39], batch scheduling policies are further classified as fixed batch scheduling (FBS) policies or adaptive batch scheduling (ABS) policies depending on how to determine the time instances for schedule computation. If the time between two schedule computation is a fixed duration then it is a FBS policy, otherwise it is a ABS policy. In general, a FBS policy does not guarantee stability unless the number of schedules employed within a batch is restricted to avoid too frequent schedule reconfiguration. The traffic matrix scheduling (TMS) policy [37] is a special case of FBS policy which could guarantee stability if the traffic load is known in advance. Under the TMS policy, the schedule computation occurs at $t_k = kW$ for some integer parameter $W < \infty$ and $k = 0, 1, \dots$ The TMS policy utilizes the Birkhoff von-Neumann (BvN) decomposition [8] to determine the schedules in the following W time slots. More specifically, the scheduler takes the queue length information $\mathbf{Q}(t_k)$ and scales it to a doubly stochastic matrix $\mathbf{B}(t_k)$ which indicates the relative service requirement in the following W slots and performs the BvN decomposition on $\mathbf{B}(t_k)$ as $\mathbf{B}(t_k) = \sum_{i=1}^{D} \alpha_i \mathbf{P}_i$. Note the number of terms D in the decomposition may vary (while $D \leq N^2 - 2N + 2$). In practice, the parameter D is set as a fixed number to avoid excessive schedule changes and D largest weighted schedules are chosen. If the arrival traffic load is known a priori, an appropriate choice of parameters ensures the stability of TMS. Besides the TMS policy, other scheduling policies that determines schedules based on decomposition of the workload (*e.g.* [10, 24, 48]) could also be considered as variants of the FBS policy.

The ABS policy proposed in [39] determines each schedule computation time as the

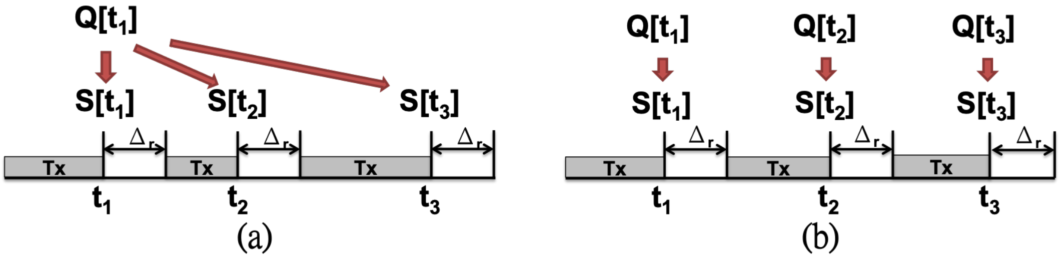

Figure 3.2: Timing diagram for different scheduling strategies. (a) Quasi-static policies: Series of schedules determined in a single schedule computation, and some schedules could depend on out-dated queue information when being deployed. (b) Dynamic policies: Each schedule is computed based on the most up-to-date edge queue information.

time the packets from last batch are cleared. The ABS guarantees rate stability [39] which is a weaker notion of stability compared to the strong stability considered in this paper. In fact, the expected queue length may be unbounded under the ABS policy, hence it is not considered in the performance comparison in this work. On the other hand, the Variable Frame MaxWeight (VFMW) policy proposed in [6] can be viewed as a variant to the ABS policy. The VFMW policy selects only one schedule for each batch, and the batch duration is a function of the batch size (instead of being the time that the batch is cleared). While the VFMW policy is shown in [6] to be throughput optimal, it has poor delay performance since it selects and fixes the schedule duration disregarding the arrivals in the schedule duration, which is a similar problem to other quasi-static policies.

3.4.2 Dynamic Scheduling Policies

In contrast to the quasi-static policies, under the dynamic scheduling policies, the schedule being employed at each schedule reconfiguration instance is based on the most up-to-date queue length information. Frame-based policies, such as the Fixed Frame MaxWeight (FFMW) policy [25], are examples for dynamic policies. The FFMW policy selects a fixed period T and sets the schedule reconfiguration times at $t_n = nT, n = 0, 1, 2, \ldots$ At each t_n, the schedule is selected as the MaxWeight schedule at time t_n, therefore each schedule is based on the most up-to-date queue information and result in an improved delay performance. Unfortunately, like the TMS policy, however, the FFMW requires prior knowledge of the traffic load to guarantee the network stability which is similar to the TMS policy. Note that by the definition given, our proposed g-adaptive policies are other examples of dynamic policies. In contrast to the FFMW, however, our proposed g-adaptive policies achieve throughput optimality without prior knowledge of the traffic load. Moreover, the g-adaptive policies also have good delay performance, as would be shown through simulations in the next section.

3.5 Application: All-Optical Data Center Network

In this section, we introduce a practical example of the switch scheduling with reconfiguration delay, which proposes the adoption of optical switches in data center networks.

Massive data centers serve as the basis of a huge variety of online services and applications nowadays. The underlying network interconnects face increasingly stringent performance requirements such as high data bandwidth and low latency. All-optical networks emerge as a promising candidate for the next generation data center networks and benefit from two technical breakthroughs: (1) the advancement of dense wavelength division multiplexing (DWDM) in optical fibers and (2) the optical switches substituting traditional electronic switches, which typically incur high cost and high power demand when supporting high data bandwidth. However, due to the inherently bufferless nature of optical switches, the data transmission in an all-optical network will need to be conducted in an end-to-end fashion. In other words, all of the buffering occurs at the end hosts or the top of the rack (ToR) switches, and the data transmission would take place only when the connection between hosts is established. An example of this architecture is proposed in [27], in form of a prototype design with extremely promising properties. From an end-to-end perspective, this architecture with zero in-network buffering can be viewed as a single crossbar switch interconnecting the end hosts (or ToR switches), with the exception that the full

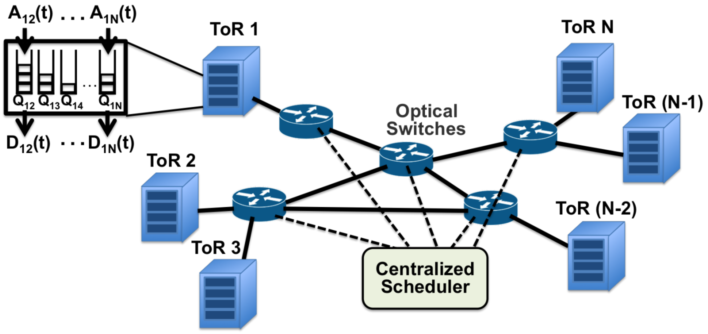
$A_{12}(t)$ . . . $A_{1N}(t)$
Q_{12} Q_{13} Q_{14} ... Q_{1N}
$D_{12}(t)$ . . . $D_{1N}(t)$
ToR 1
ToR 2
ToR 3
ToR N
ToR (N-1)
ToR (N-2)
Optical Switches
Centralized Scheduler

Figure 3.3: An illustration of the system model

bisection bandwidth is not always guaranteed. Under this architecture, an efficient utilization of the all-optical network depends on a centralized controller that can efficiently schedule the end-to-end transmissions.

The main challenge of the scheduling for optical networks, as opposed to the traditional electronic crossbar switch scheduling, comes from the fact that optical networks typically exhibit a nonzero *reconfiguration delay* upon changing the circuit configuration. For instance, candidate technologies such as MEMS [47], WSS [13] involve mechanically directing laser beams and thus require a certain time to finish circuit reconfiguration due to physical limits. During the circuit reconfiguration, reliable packet transmission could not be supported in the network. Notice that for the architecture considered in this paper, the reconfiguration delay may further include the time for the control plane to control/communicate with the optical switches, and the time for end hosts to initiate/pause packet transmissions. For example, a state of the art system implemented in [27] reports a measured reconfiguration delay up to 20 μs (including delay caused by the control plane and end-host), which is significantly larger than the inter-frame gap of 0.96 ns (for 100 Gigabit Ethernet). This nonzero reconfiguration delay motivates the need for scheduling policies that explicitly account for the reconfiguration delay.

Fig. 3.3 gives an illustration of the proposed optical network scheduling framework. Each ToR switch can serve both as a source and a destination simultaneously. We assume no buffering in the optical network, hence all the buffering occurs in the edge of the network, *i.e.* within the

ToR switches. Each ToR switch maintains $N-1$ edge queues (either physically or virtually), which are denoted by Q_{ij}, where $j \in \{1, 2, \ldots, N\} \backslash \{i\}$. Packets going from ToR switch i to j are enqueued in the edge queue Q_{ij} before transmission.

In the next section, we present simulation results in the context of all-optical data center networks.

3.6 Simulations

We now present simulation results for the AMW policy, and compare them to the benchmark scheduling policies such as TMS, FFMW and VFMW. We also present comparison of the AMW policy against adaptive variants of low complexity policies approximating the MaxWeight policy, as given in the examples in section 3.3.

The experiments are conducted with the simulator built for the REACToR switch in [27]. The reconfiguration delay is $\Delta_r = 20\ \mu s$. In order to compare scheduling policies in optical switches, we cease the electronic switches in the hybrid switch in [27] and only utilize the optical switches. We consider $N = 100$ ToR switches, and the network topology is assumed to be non-blocking. Therefore, the set of feasible schedules $\mathcal{S}$ is in fact the set of $N \times N$ permutation matrices. Each link has data bandwidth $B = 100$ Gbps, and the packets are of the same size $p = 1500$ bytes (each takes $0.12 \mu s$ for transmission). Each edge queue can store up to 1×10^5 packets, and incoming packets are discarded when the queue is full.

The traffic is assumed to be admissible, i.e. $\rho(\boldsymbol{\lambda}) < 1$, while the load matrices $\boldsymbol{\lambda}$ are classified as the following types:

1. Uniform: $\lambda_{ij} = \rho/N$, $\forall 1 \leq i, j \leq N$.

2. Nonuniform: $\lambda_{ij} = \frac{\rho}{M} \sum_{m=1}^{M} \mathbf{P}_{ij}^{m}$ where $\mathbf{P}^m \in \mathcal{P}$ are randomly selected permutation matrices, and $M = 100$.

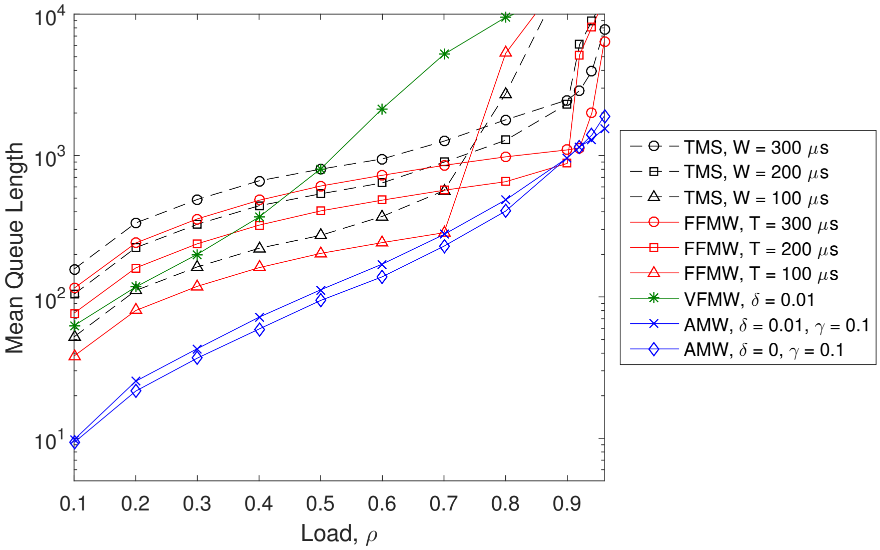

Figure 3.4: Mean queue length versus traffic load ρ under **uniform traffic**. The TMS policy reconfigures the schedule $D = 10$ times within DT slots. The scheduling rate under either the TMS or PMW is equal to $1/T$, while under AMW is adapted to the traffic load intensity.

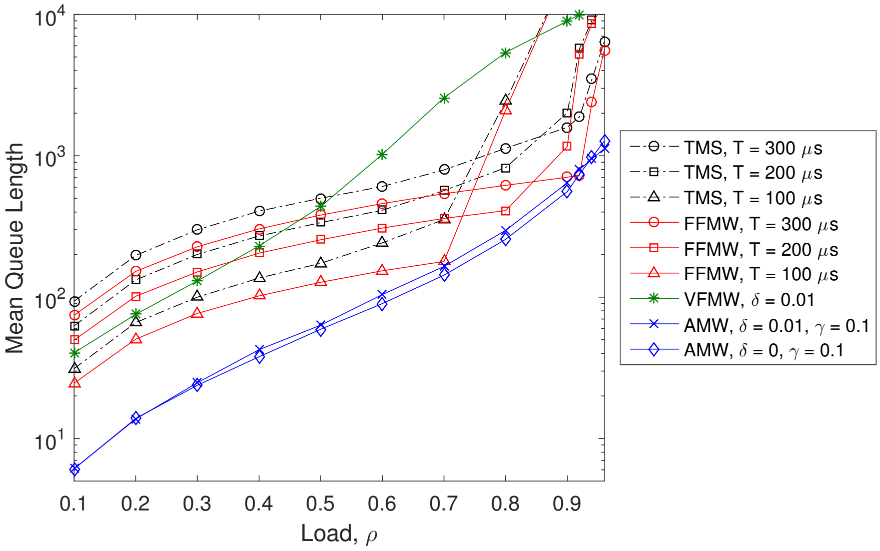

Figure 3.5: Mean queue length versus traffic load ρ under **nonuniform traffic**. The TMS policy reconfigures the schedule $D = 10$ times within DT slots.

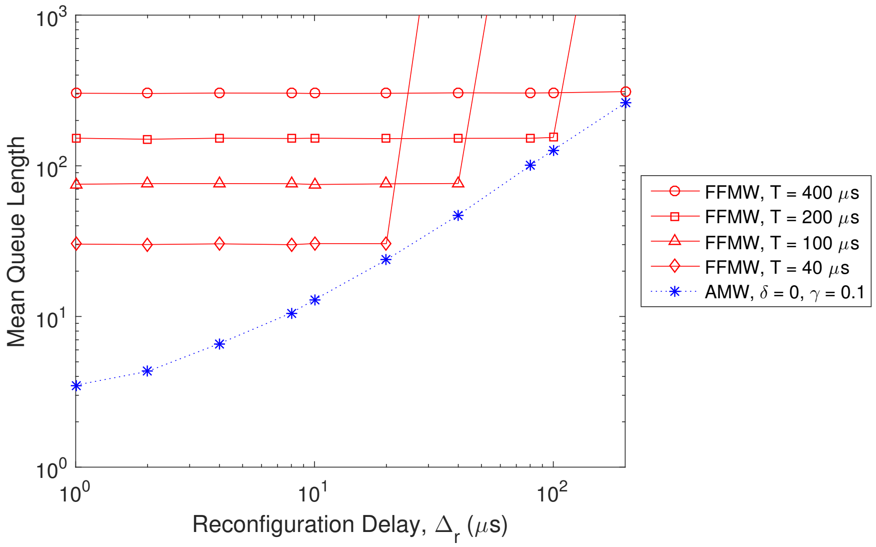

Figure 3.6: Mean queue length versus the reconfiguration delay Δ_r under nonuniform traffic. The traffic load is fixed as $\rho = 0.3$.

The performance measure used is the mean edge queue length (averaged over queues and over time). Notice that the expected average delay of the entire network is linearly related to this quantity according to the Little's law.

In Figs. 3.4 and 3.5, we compare the scheduling policies described in section 3.4 under the uniform and the nonuniform traffic, respectively. For TMS, we set the number of schedules used between two schedule computation time instances to be $Q = 10$. In Figs. 3.4 and 3.5 we can see that the TMS and FFMW perform comparably, while the FFMW slightly outperforms the TMS under the same schedule reconfiguration rate $1/T = 10/W$. We note that under both the TMS and FFMW policies, the traffic loads they could stabilize are determined by the reconfiguration rate $1/T = 10/W$. In general, a smaller T (W) value gives better delay performance at a fixed stable load, but choosing a smaller T (W) value also decreases the maximum load that the TMS or FFMW policy could stabilize, which makes the design choice highly dependent on the accuracy of the traffic load. We also note that the VFMW policy, while is known to achieve the throughput optimality, underperforms both the FFMW and TMS policies in practice. In contrast, the AMW policy shows an improved performance over the FFMW and TMS policies, and at the same time guarantees throughput optimality. Another important implication from the two figures is that even if we know exactly the traffic load ρ and fine tune the optimal schedule reconfiguration period T for the FFMW and TMS policies, they are at most comparable to the AMW policy. The AMW policy has the merit that achieves better performance (over other policies) at any traffic load without the need to adjust the parameters.

The end-to-end per packet delay of the FFMW and AMW policies under various reconfiguration delay Δ_r are shown in Fig. 3.6. The traffic load is fixed as $\rho = 0.3$. We can see that the AMW outperforms the FFMW under each Δ_r value, regardless of the parameter selection of the FFMW policy. Even more notable, the performance of the AMW actually traces the optimal performance of the FFMW. This observation, along with the observation in the previous paragraph, suggests that the adaptive strategy of the AMW allows it to capture the optimal schedule reconfiguration rate based solely on the queue lengths and no prior knowledge of the arrival statistics is required.

The previous simulation results may also be used to evaluate the practicality of the proposed all-optical network architecture, in terms of the memory required at the ToR switches. For example, consider a network with $N = 100$ ToRs with reconfiguration delay $\Delta_r = 20$ μs as shown in Fig. 3.5. Even at a very high traffic load $\rho = 0.9$, the mean total queue length in a ToR (aggregating queues destined for all ToRs) is approximately $10^3 * 1500$ bytes $* N = 150$MBytes, which is still feasible in commercially available ToR switches [2].

We now consider the effect of the parameter selection for the AMW policy. In particular, we consider the performance as a function of parameters of the hysteresis function $g(x) = (1-\gamma)x^{1-\delta}$, evaluated under various traffic loads ($\rho \in \{0.3, 0.5, 0.8, 0.9, 0.98\}$). In Fig. 3.7, we fix the threshold ratio as $\gamma = 0.01$ and evaluate the performance under various sublinear exponent δ. Note that the mean queue length becomes shorter (better performance) when δ approaches 0 (which means g is closer to a linear function), and that the mean queue length increases quickly as δ becomes larger. This implies that we should avoid selecting $g(x)$ that grows too slow with x

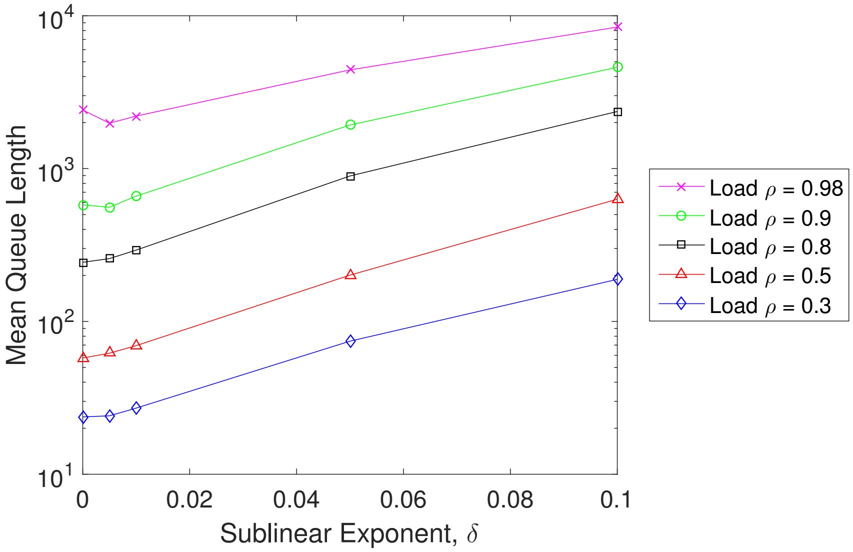

Figure 3.7: Mean queue length versus sublinear exponent δ for the AMW policy. The threshold ratio is fixed as $\gamma = 0.01$.

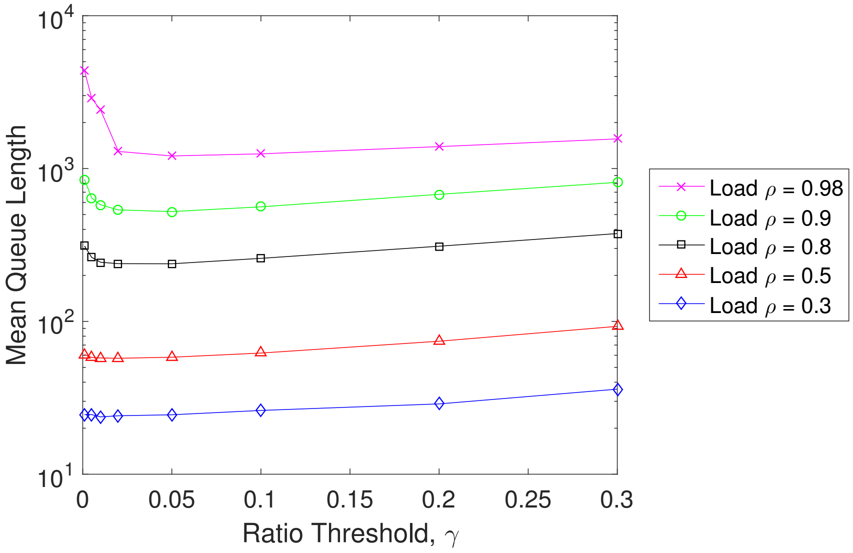

Figure 3.8: Mean queue length versus threshold ratio γ for the AMW policy. The sublinear exponent is fixed as $\delta = 0$.

(such as large δ, or even logarithmic functions like $g(x) = \log(1+x)$) for better performance, despite the fact that throughput optimality guarantee from Section 3.3 actually applies for a large class of sublinear functions $g(x)$. Another observation we could make from Fig. 3.7 regards the performance at $\delta = 0$. Note that the throughput optimality guarantee does not include this case since g becomes a linear function. However, numerically and anecdotally the performance at $\delta = 0$ is generally good, and is even the optimal at most traffic loads (except at very high traffic load). The above observations simplify the parameter selection process: Since we rarely operate the network at traffic load close to the capacity, it is reasonable just to fix δ at a value close to 0.

In Fig. 3.8, we show the performance under variousvalues of ratio threshold γ, while the sublinear exponent is fixed as $\delta = 0$. Comparing with Fig. 3.7, we first note that the effect of γ on the performance is even less significant than the effect of δ. Moreover, from the curves we could find a region of γ values (around $[0.05, 0.1]$) where the performances are close to the optimal performance at each traffic load.

Combining the above results, the non-asymptotic delay performance of the AMW policy is quite robust to the parameter selection in the sense that we could easily select a combination of (δ, γ) that has comparable performance to the optimal, even if we have no knowledge of the traffic load. This result further strengthens our analytical result that the AMW policy achieves throughput optimality without any knowledge of the traffic load. Besides the traffic load, we also consider the robustness of parameter selection under different system parameters such as number of ToR switches N, and reconfiguration delay Δ_r. In Tables 3.1 and 3.2 shown below, we compare the mean queue length under a set of fixed parameters $(\delta, \gamma) = (0.01, 0.05)$ to the minimum mean queue length (over all parameter combinations) under different system parameters, and compute the percentage of excess queue length. The traffic load is fixed as $\rho = 0.5$. We could see from the tables that the performance of the fixed parameter is close to the optimal in all these cases, and conclude that the robustness of the AMW also holds under different system parameters.

One of the shortcomings of the AMW policy is the complexity for computing the

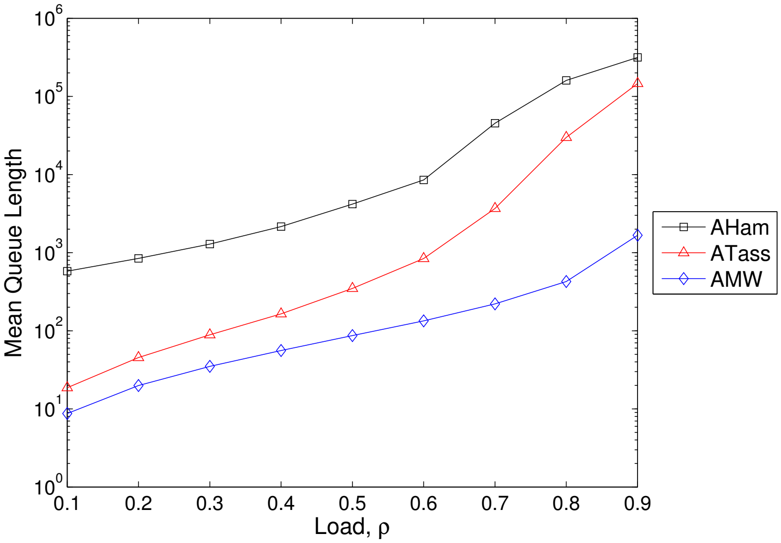

Figure 3.9: Mean queue length versus traffic load for different adaptive policies. Number of ToRs is $N = 8$ and $g(x) = (1 - \gamma)x^{1-\delta}$ with $\gamma = 0.1, \delta = 0.01$.

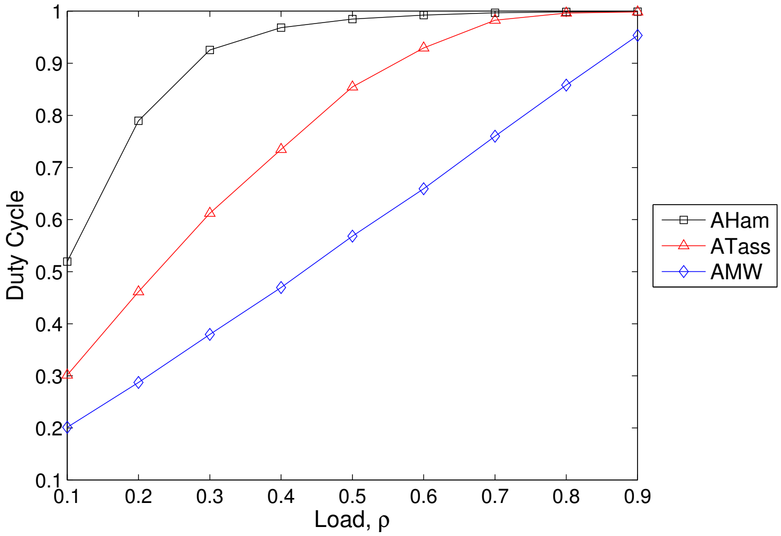

Figure 3.10: Duty cycle versus traffic load for different adaptive policies. Number of ToRs is $N = 8$ and $g(x) = (1 - \gamma)x^{1-\delta}$ with $\gamma = 0.1, \delta = 0.01$.

Table 3.1: Excess over minimum under various N, with $\Delta_r = 20\mu$s.

N	4	8	16	32	64	100
Excess (%)	2.8	1.2	0.4	1.8	1.0	2.4

Table 3.2: Excess over minimum under various Δ_r, with $N = 100$.

Δ_r (μs)	20	50	100	200	500
Excess (%)	2.4	3.1	2.4	2.1	1.6

MaxWeight schedules. Next, we consider the adaptive variants of lower complexity scheduling policies introduced in examples 3.4 and 3.5. Fig. 3.9 shows the mean queue length versus traffic load for the Adaptive Hamiltonian (AHam), the Adaptive Tassiulas random policy (ATass), and the AMW policies, under uniform traffic. Since the delay performance of the lower complexity policies degrade drastically at large number of ToRs ($G \approx N!$ in Condition 3.1 for AHam and ATass), hence we set $N = 8$ for the simulations here. We also show the duty cycles of these policies in Fig. 3.10, where the duty cycle is defined as $DC \triangleq 1 - \frac{\Delta_r}{\mathbb{E}[T]}$, and $\mathbb{E}[T]$ is the mean schedule duration. Note that a necessary condition for a scheduling policy to be throughput optimal under traffic load ρ is to satisfy $DC > \rho$, which is satisfied by all the scheduling policies shown here.

From Fig. 3.9, we could see that the delay performance for the lower complexity policies are worse than the AMW policy. We may also observe that as the traffic load gets larger, the performance difference increases. This is because the schedule weight decreases in a slower rate (due to higher arrival rate), and it takes more time for the lower complexity policies to find a schedule with high enough weight. This could be validated by Fig. 3.10 that the schedule durations become significantly long ($DC \to 1$) at high traffic loads. Note that the AHam policy has the worst delay performance for all traffic loads, this is probably because the Hamiltonian walk only changes few served queues in a time slot, and generally takes longer to find the next good schedule.

3.7 Concluding Remarks

In this chapter we consider the scheduling problem for switch model subject to reconfiguration delay. Due to the schedule reconfiguration delay, many throughput optimal scheduling policies (under zero reconfiguration delay) in the literature could not be directly applied in this problem. We propose the Adaptive MaxWeight policy that addresses the schedule reconfiguration delay and generalizes the Adaptive MaxWeight policy to a general method to develop a class of scheduling policies for scheduling with nonzero reconfiguration delay, namely the g-adaptive variant policies. Given any Markov policy π, a weight function W, and a sublinear hysteresis function $g(\cdot)$, we construct a g-adaptive variant of π which involves a weight comparison between the current schedule and the schedule generated by π, and reconfigure to the schedule generated by π when it is "significantly better." We prove the throughput optimality of the Adaptive MaxWeight and the g-adaptive variants of π given the weight of schedule generated by π is guaranteed to have bounded weight difference to the maximum weight policy (either in the deterministic or the expected sense).

Chapter 3, in part, is a reprint of the material as it appears in the paper: C.-H. Wang, T. Javidi, G. Porter, ”End-to-end Scheduling for All-Optical Data Centers”, IEEE Conference on Computer Communications (INFOCOM), pp 406-414, 2015. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in part, is a reprint of the material as it appears in the paper: C.-H. Wang, T. Javidi, ”Adaptive Policies for Scheduling with Reconfiguration Delay: An End-to-end Solution for All-Optical Data Centers”, IEEE/ACM Transactions on Networking (TON), vol 25, pp 1555-1568, 2017. The dissertation author was the primary investigator and author of this paper.

3.8 Supplementary Proofs

The following Foster Lyapunov Theorem is used to show the stability in the proofs in this section.

Fact 1 (Foster-Lyapunov [20]). *Given a system of edge queues $Q_{ij}, 1 \leq i, j \leq N$, with queue lengths $\mathbf{Q}(t) = [Q_{ij}(t)]$, which could be described by an irreducible discrete-time Markov chain (DTMC) on a countable state space $\mathbb{N}^{N \times N}$. Let f, g be two nonnegative functions on $\mathbb{N}^{N \times N}$ such that $\forall \mathbf{Q}(t_k) \in \mathbb{N}^{N \times N}$,*

$$\mathbb{E}\left[V\left(\mathbf{Q}(t_{k+1})\right) - V(\mathbf{Q}(t_k)) | \mathbf{Q}(t_k)\right] \leq -f\big(\mathbf{Q}(t_k)\big) + g\big(\mathbf{Q}(t_k)\big)$$

and suppose for some $\epsilon > 0$, the set $C = \{\mathbf{Q} \in \mathbb{N}^{N \times N} : f(\mathbf{Q}) < g(\mathbf{Q}) + \epsilon\}$ is finite, then the DTMC describing the queue length evolution is positive recurrent and we have

$$\lim_{k \to \infty} \mathbb{E}\left[f\big(\mathbf{Q}(t_k)\big)\right] \leq \lim_{k \to \infty} \mathbb{E}\left[g\big(\mathbf{Q}(t_k)\big)\right]$$

3.8.1 Proof of Theorem 3.2

Notice that in the $\Delta_r > 0$ regime, the queue dynamics depends on whether the system is in reconfiguration. We use $r(t)$ to denote the remaining time for the system to be in reconfiguration at time t, while $r(t) = 0$ indicates that the system is not in reconfiguration.

Now define $X(t) = \big(\mathbf{Q}(t), \mathbf{S}(t-1), r(t)\big)$, then $\{X(t)\}_{t=0}^{\infty}$ forms an irreducible DTMC. In the following, we consider the T-step conditional Lyapunov drift (where the value of $T < \infty$ is to be determined later) to show the stability. In other words, we determine the conditional

Lyapunov drift for the sampled process $\{X(t_k)\}_{k=0}^{\infty}$ (which is also a DTMC) at times $t_k = kT$:

$$\begin{aligned}\mathbb{E}^{\pi^g}\left[V(\mathbf{Q}(t_{_{k+1}})) - V(\mathbf{Q}(t_k))\middle|X(t_k)\right] &= \sum_{t=t_k}^{t_{k+1}-1} \mathbb{E}^{\pi^g}\left[\Delta V(t)\middle|X(t_k)\right] \\ &= \sum_{t=t_k}^{t_{k+1}-1} \mathbb{E}^{\pi^g}\left[\mathbb{E}^{\pi^g}\left[\Delta V(t)|X(t)\right]\middle|X(t_k)\right] \qquad (3.29)\end{aligned}$$

By the assumption of the Theorem and that the system could not serve the queues during reconfiguration, we may write

$$\begin{aligned}\mathbb{E}^{\pi^g}\left[\Delta V(t)\middle|X(t)\right] \leq& \Lambda(\mathbf{Q}(t)) - \mathbb{E}^{\pi^g}\left[W^{\pi^g}(t)\mathbb{1}_{\{r(t)=0\}}\middle|X(t)\right] \\ \leq& -\epsilon W^*(t) + K + \Big(W^*(t) - \mathbb{E}^{\pi^g}\left[W^{\pi^g}(t)\mathbb{1}_{\{r(t)=0\}}\middle|X(t)\right]\Big) \\ \leq& -\epsilon W^*(t) + K + \Big(W^*(t) - \mathbb{E}^{\pi^g}\left[W^{\pi^g}(t)\middle|X(t)\right]\Big) \\ &+ \mathbb{E}^{\pi^g}\left[W^*(t)\mathbb{1}_{\{r(t)>0\}}\middle|X(t)\right], \qquad (3.30)\end{aligned}$$

where the last inequality follows from $W^{\pi^g}(t) \leq W^*(t)$ and $\mathbb{1}_{\{r(t)=0\}} + \mathbb{1}_{\{r(t)>0\}} = 1$.

By the construction of the g-adaptive variant and that π satisfies Condition 3.1, we have

$$\begin{aligned}W^*(t) - \mathbb{E}^{\pi^g}\left[W^{\pi^g}(t)\middle|X(t)\right] =& W^*(t) - \mathbb{E}^{\pi}\left[W^{\pi}(t)\middle|X(t)\right] \\ &+ \mathbb{E}^{\pi^g}\left[W^{\pi}(t) - W^{\pi^g}(t)\middle|X(t)\right] \\ \leq& G + \mathbb{E}^{\pi^g}\left[g(W^{\pi}(t))\middle|X(t)\right] \\ \leq& G + g(W^*(t)) \qquad (3.31)\end{aligned}$$

Since at most one packet could depart and at most $A_{\max}$ packets could arrive at each queue in a time slot, then for any $t \in [t_k, t_{k+1}]$, we have $|Q_{ij}(t) - Q_{ij}(t_k)| \leq A_{\max}(t - t_k) \leq A_{\max}T$, and thus $|\sum_{ij} Q_{ij}(t) - \sum_{ij} Q_{ij}(t_k)| \leq N^2 A_{\max}T$. Hence for any schedule $\mathbf{S}$, we have $|W_{\mathbf{S}}(\mathbf{Q}(t)) - W_{\mathbf{S}}(\mathbf{Q}(t_k))| \leq BA_{\max}N^2T$, where B is the Lipshitz constant. With this relation we derive the following bounds for the maximum weight at time t:

$$\begin{aligned} W^*(t) = W_{\mathbf{\Pi}^*(t)}(\mathbf{Q}(t)) &\le W_{\mathbf{\Pi}^*(t)}(\mathbf{Q}(tQ_{ij}(t)_k)) + BA_{\max}N^2T \\ &\le W^*(t_k) + BA_{\max}N^2T \qquad (3.32) \\ W^*(t) \ge W_{\mathbf{\Pi}^*(t_k)}(\mathbf{Q}(t)) &\ge W_{\mathbf{\Pi}^*(t_k)}(\mathbf{Q}(t_k)) - BA_{\max}N^2T \\ &= W^*(t_k) - BA_{\max}N^2T \qquad (3.33) \end{aligned}$$

Apply (3.30)-(3.33) into (3.29), we obtain

$$\begin{aligned} &\mathbb{E}^{\pi^g}\left[V\left(\mathbf{Q}(t_{k+1})\right) - V(\mathbf{Q}(t_k))\middle|\mathbf{Q}(t_k)\right] \\ \le& -\epsilon T(W^*(t_k) - BA_{\max}N^2T) + Tg(W^*(t_k) + BA_{\max}N^2T) \\ &+ T(G+K) + \sum_{t=t_k}^{t_{k+1}-1}\mathbb{E}^{\pi^g}\left[W^*(t)\mathbb{1}_{\{r(t)>0\}}\middle|\mathbf{Q}(t_k)\right] \qquad (3.34) \end{aligned}$$

In order to achieve the stability, it is necessary to ensure that the reconfiguration does not happen too often. We make this statement formal with the following lemma:

Lemma 3.2. *Given any fixed* $T > 0$ *and a weight function* W *satisfying Assumption 3.1 with Lipschitz constant* B. *Let* $g(x)$ *be a sublinear and strictly increasing function, and let* $h(x) = g(x - BA_{\max}N^2T) - 2BA_{\max}N^2T$. *Suppose the Markov policy* π *satisfies Condition 3.1 under the weight function* W, *then under the* g*-adaptive variant of* π:

$$\mathbb{E}^{\pi^g}\left[\sum_{t=t_k}^{t_{k+1}-1}\mathbb{1}_{\{r(t)>0\}}\middle|X(t_k)\right] \le \Delta_r + \frac{TG}{h(W^*(t_k))} \qquad (3.35)$$

The proof of Lemma 3.2 is given in section 3.8.2. Note that lemma 3.2 gives an upper bound on the frequency of schedule reconfiguration when the queue length is large. Now select T

such that $T > \frac{\Delta_r}{\epsilon}$, and apply Lemma 3.2 to (3.34), we obtain

$$\begin{aligned}\mathbb{E}^{\pi^g}\left[V\left(\mathbf{Q}(t_{k+1})\right) - V(\mathbf{Q}(t_k))\middle|X(t_k)\right] \leq &- T\big(\epsilon - \frac{\Delta_r}{T} - \frac{G}{h(W^*(t_k))}\big)W^*(t_k) \\ &+ Tg\big(W^*(t_k) + BA_{\max}N^2T\big) + T(G+K) \\ &+ T\big(\epsilon + \frac{\Delta_r}{T} + \frac{G}{h(W^*(t_k))}\big)BA_{\max}N^2T \qquad (3.36)\end{aligned}$$

Since $T > \frac{\Delta_r}{\epsilon}$, we may select an arbitrary constant $\epsilon' \in (0, \epsilon - \frac{\Delta_r}{T})$. Then from the sublinearity of g, we obtain

$$\mathbb{E}^{\pi^g}\left[V(\mathbf{Q}(t_{k+1})) - V(\mathbf{Q}(t_k))\middle|X(t_k)\right] < -T\epsilon' W^*(t_k) + K' \qquad (3.37)$$

for some constant $K' < \infty$. It then follows Fact 1 that $\lim\limits_{k\to\infty} W^*(t_k) < \infty$, and hence we have $\lim\limits_{k\to\infty} \sum_{ij} Q_{ij}(t_k) < \infty$, which then implies the strong stability. Since the derived strong stability holds for any admissible traffic load, we conclude that the g-adaptive variant of π achieves throughput optimality.

3.8.2 Proof of Lemma 3.2

Proof. Let $Z_{[s,t]}$ be the number of schedule reconfigurations within the time period $[s,t]$. In the following, we first derive an upper bound on $\Pr\{Z_{[t_k,t_{k+1}]} \leq 1\}$, then use this bound to get an upper bound on the expected length of reconfiguration during $[t_k, t_{k+1}]$.

Let $t'_k = \min\limits_{k:t^S_k \geq t} t^S_k$ denote the first reconfiguration instance after time t_k. Since either there is no reconfiguration within $[t_k, t_{k+1}]$ or $t'_k = \tau$ for some $\tau \in [t_k, t_{k+1}]$, we have

$$\Pr\left\{Z_{[t_k,t_{k+1}]} = 0\right\} + \sum_{\tau=t_k}^{t_{k+1}-1} \Pr\left\{\tau = t'_k\right\} = 1 \qquad (3.38)$$

We may also write the probability that exactly one schedule reconfiguration occurs in the interval $[t_k, t_{k+1}]$ as:

$$\begin{aligned}\Pr\{Z_{[t_k,t_{k+1}]} = 1\} &= \sum_{\tau=t_k}^{t_{k+1}-1} \Pr\left\{Z_{[\tau,t_{k+1}]} = 0, \tau = t'_k\right\} \\ &= \sum_{\tau=t_k}^{t_{k+1}-1} \Pr\left\{Z_{[\tau,t_{k+1}]} = 0 \middle| \tau = t'_k\right\} \Pr\left\{\tau = t'_k\right\} \end{aligned} \tag{3.39}$$

We now show that if at the schedule reconfiguration instance τ, the weight difference is small, then no reconfiguration could occur in the interval $[\tau, t_{k+1}]$. In other words, we need to show that $W^{\pi}(\tau') - W_{\Pi(\tau)}(\tau') \leq g(W^{\pi}(\tau'))$ for any $\tau' \in [\tau, t_{k+1}]$.

Let $h(x) = g(x - BA_{\max}N^2T - G) - 2BA_{\max}N^2T$, then if $W^*(\tau) - W_{\Pi^g(\tau)}(\tau) < h(W^*(t_k))$, we have:

$$\begin{aligned}W^*(\tau') - W_{\Pi(\tau)}(\tau') \leq & W^*(\tau') - W_{\Pi(\tau)}(\tau') \\ \leq & W^*(\tau) - W_{\Pi(\tau)}(\tau) + 2BA_{\max}N^2(\tau' - \tau) \\ < & g(W^*(t_k) - BA_{\max}N^2T - G) \\ \overset{(a)}{\leq} & g(W^{\pi}(\tau'))\end{aligned} \tag{3.40}$$

where (a) follows from $W^{\pi}(\tau') \geq W^*(\tau') - G \geq W^*(t_k) - G - BA_{\max}N^2T$ and g strictly increasing. We thus have:

$$\begin{aligned}& \Pr\left\{Z_{[\tau,t_{k+1}]} = 0 \middle| \tau = t'_k\right\} \\ \geq & \Pr\left\{W^*(\tau) - W_{\Pi(\tau)}(\tau) < h(W^*(t)) \middle| \tau = t'_k\right\} \\ \overset{(b)}{\geq} & 1 - \frac{\mathbb{E}^{\pi}\left[W^*(\tau) - W_{\Pi(\tau)}(\tau)\right]}{h(W^*(t_k))} \overset{(c)}{\geq} 1 - \frac{G}{h(W^*(t_k))}\end{aligned} \tag{3.41}$$

where (b) follows from the Markov's inequality, and (c) follows from Condition 3.1.

Hence by (3.39) and (3.41) we have that

$$\begin{aligned}
&\Pr\left\{Z_{[t_k,t_{k+1}]} \leq 1\right\} \\
\geq &\Pr\left\{Z_{[t_k,t_{k+1}]} = 0\right\} + \sum_{\tau=t_k}^{t_{k+1}-1} \left(1 - \frac{G}{h(W^*(t_k))}\right) \Pr\left\{\tau = t'_k\right\} \\
\geq &1 - \frac{G}{h(W^*(t_k))}
\end{aligned} \tag{3.42}$$

With the following bound which is obvious by definition:

$$\sum_{t=t_k}^{t_{k+1}-1} \mathbb{1}_{\{r(t)>0\}} \leq \begin{cases} \Delta_r, & \text{if } Z_{[t_k,t_{k+1}]} \leq 1 \\ T, & \text{if } Z_{[t_k,t_{k+1}]} > 1 \end{cases}, \tag{3.43}$$

along with (3.42), we then have the bound on the expected schedule reconfiguration delay within the interval $[t_k, t_{k+1}]$:

$$\mathbb{E}^{\pi^g}\left[\sum_{t=t_k}^{t_{k+1}-1} \mathbb{1}_{\{r(t)>0\}} \middle| X(t_k)\right] \leq \Delta_r + \frac{TG}{h(W^*(t_k))} \tag{3.44}$$

□

3.8.3 Proof of Theorem 3.3

Proof. We evaluate the conditional Lyapunov drift for the sampled process $\{X(t_k)\}_{k=0}^{\infty}$ following (3.29). Now consider the Lyapunov function $V(\mathbf{Q}) = \sum_{i,j=1}^{N} F(Q_{ij})$, where $F(x) = \int_0^x f(y)dy$ and let $\boldsymbol{\Delta}(t) = \mathbf{A}(t) - \mathbf{D}(t)$, then following the derivation similar to [22], we have the one-step expected Lyapunov drift given by

$$\begin{aligned}\mathbb{E}^{\pi^g}\left[\Delta V\left(t\right)\Big|X(t)\right] =&\mathbb{E}^{\pi^g}\Big[\sum_{i,j}\Big(F(Q_{ij}(t)+\Delta_{ij}(t))-F(Q_{ij}(t))\Big)\Big|X(t)\Big]\\ \leq&\mathbb{E}^{\pi^g}\left[\left\langle f(\mathbf{Q}(t)),\boldsymbol{\Delta}(t)\right\rangle\middle|X(t)\right]\\ &+\sum_{i,j}\frac{\Delta_{ij}(t)^2}{2}\max_{u\in[-1,A_{\max}]}\left|f'\left(Q_{ij}(t)+u\right)\right| \end{aligned}\tag{3.45}$$

Now note that

$$\begin{aligned}&\mathbb{E}^{\pi^g}\left[\Big\langle f(\mathbf{Q}(t)),\boldsymbol{\Delta}(t)\Big\rangle\Big|X(t)\right]\\ =&\mathbb{E}\left[\Big\langle f(\mathbf{Q}(t)),\mathbf{A}(t)-\boldsymbol{\Pi}^g(t)\mathbb{1}_{\{r(t)=0\}}\Big\rangle\Big|X(t)\right]\\ =&\Big\langle f(\mathbf{Q}(t)),\boldsymbol{\lambda}-\boldsymbol{\Pi}^g(t)\Big\rangle+W^{\pi^g}(t)\mathbb{1}_{\{r(t)>0\}}\\ \leq&\Big\langle f(\mathbf{Q}(t)),\boldsymbol{\lambda}-\boldsymbol{\Pi}^*(t)\Big\rangle+\big(W^*(t)-W^{\pi^g}(t)\big)+W^*(t)\mathbb{1}_{\{r(t)>0\}}\end{aligned}\tag{3.46}$$

By the definition of the traffic load ρ, we may write $\boldsymbol{\lambda}=\rho\sum_{i=1}^{I}\mathbf{P}_i$, where $\mathbf{P}_i\in\mathcal{S}$ for $i=1,\ldots,I$. Then by the definition of the MaxWeight schedule, we have

$$\Big\langle f(\mathbf{Q}(t)),\boldsymbol{\lambda}-\boldsymbol{\Pi}^*(t)\Big\rangle\leq-(1-\rho)W^*(t)\tag{3.47}$$

Also by the construction of the g-adaptive policy and the fact that the scheduling policy satisfies Condition 3.2, we have

$$\begin{aligned}W^*(t)=W^{\pi^g}(t)=&[W^*(t)-W^{\pi}(t)]+[W^{\pi}(t)-W^{\pi^g}(t)]\\ \leq&G+g(W^{\pi}(t))\leq G+g(W^*(t))\end{aligned}\tag{3.48}$$

By Assumption 2.1, we have $\Delta_{ij}^2(t)\leq A_{\max}^2$. Also, by the assumption that $\lim\limits_{x\to\infty}\frac{f'(x)}{g(f(x))}=$

0, we have that $\forall \epsilon > 0$, there exists a constant $K_\epsilon < \infty$ such that

$$\max_{u \in [-T, A_{\max} T]} \Big| f'\left(x+u\right) \Big| \leq \epsilon g(f(x)) + K_\epsilon. \tag{3.49}$$

We then obtain the following:

$$\begin{aligned}\sum_{ij} \frac{\Delta_{ij}^2(t)}{2} \max_{u \in [-1, A_{\max}]} \Big| f'\left(Q_{ij}(t)+u\right) \Big| \leq & \frac{N^2 A_{\max}^2}{2} \left(\epsilon g \left(f \left(\max_{i,j} Q_{ij}(t) \right) \right) + K_\epsilon \right) \\ \leq & \frac{N^2 A_{\max}^2}{2} \Big(\epsilon g(W^*(t)) + K_\epsilon \Big) \end{aligned} \tag{3.50}$$

We thus have

$$\begin{aligned} & \mathbb{E}^{\pi^g} \Big[V(\mathbf{Q}(t_{k+1})) - V(\mathbf{Q}(t_k)) \Big| X(t_k) \Big] \\ \leq & \sum_{t=t_k}^{t_{k+1}-1} \mathbb{E}^{\pi^g} \Big[-(1-\rho) W^*(t) + W^*(t) \mathbb{1}_{\{r(t)>0\}} + g(W^*(t)) \\ & + \frac{N^2 A_{\max}^2}{2} \left(\epsilon g(W^*(t)) + K_\epsilon \right) + G \Big| X(t_k) \Big] \end{aligned} \tag{3.51}$$

We now justify that the reconfiguration overhead is upper bounded when the queue lengths are large. This is characterized by the following lemma:

Lemma 3.3. *Given any fixed $T > 0$ and a scheduling policy π satisfying Condition 3.2 under the weight function $f(\cdot)$. Let $g(\cdot)$ be a sublinear, strictly increasing function satisfying $\lim_{x \to \infty} \frac{f'(x)}{g(f(x))} = 0$. Then there exists a constant $M < \infty$ such that, if a reconfiguration occurs at time t and $W^*(t_k) > M$, then no reconfiguration could occur in $[t+1, t+T]$.*

We postpone the proof of Lemma 3.3 to Appendix 3.8.4. Now by Lemma 3.3, if $W^*(t_k) >$

M, then since at most one schedule reconfiguration could occur within $[t_k, t_{k+1}]$, we have

$$\sum_{t=t_k}^{t_{k+1}-1} \mathbb{1}_{\{r(t)>0\}} \leq \Delta_r \tag{3.52}$$

Similar to (3.33), we derive the following lower and upper bounds on the change of the maximum weight:

$$\begin{aligned} W^*(t) = W_{\Pi^*(t)}(\mathbf{Q}(t)) &\leq \sum_{i,j} \Pi^*_{ij}(t) f(Q_{ij}(t_k) + A_{\max}T) \\ \leq & \sum_{i,j} \Pi^*_{ij}(t)\Big(f(Q_{ij}(t_k)) + A_{\max}T \max_{u\in[0,A_{\max}T]} |f'(Q_{ij}(t_k)+u)|\Big) \\ \leq & W^*(t_k) + NA_{\max}T\left(\epsilon g(W^*(t_k)) + K_\epsilon\right) \end{aligned} \tag{3.53}$$

$$\begin{aligned} W^*(t) \geq W_{\Pi^*(t_k)}(\mathbf{Q}(t)) &\geq \sum_{i,j} \Pi^*_{ij}(t_k) f(Q_{ij}(t_k) - T) \\ \geq & \sum_{i,j} \Pi^*_{ij}(t_k)\Big(f(Q_{ij}(t_k)) - T \max_{u\in[-T,0]} |f'(Q_{ij}(t_k)+u)|\Big) \\ \geq & W^*(t_k) - NT\left(\epsilon g(W^*(t_k)) + K_\epsilon\right) \end{aligned} \tag{3.54}$$

Apply (3.52) - (3.54) into (3.51), then $\forall X(t_k) : W^*(t_k) > M$,

$$\begin{aligned} &\mathbb{E}^{\pi^g}\left[V(\mathbf{Q}(t_{k+1})) - V(\mathbf{Q}(t_k))\middle| X(t_k)\right] \\ \leq & -T\big(1-\rho-\frac{\Delta_r}{T}\big)W^*(t_k) \\ &+ (T(1-\rho) + \Delta_r A_{\max})NT(\epsilon g(W^*(t_k)) + K_\epsilon) \\ &+ Tg(W^*(t_k) + NA_{\max}T(\epsilon g(W^*(t_k)) + K_\epsilon)) + TG \\ &+ \frac{T}{2}N^2A_{\max}^2\big(\epsilon g(W^*(t_k) + NA_{\max}T(\epsilon g(W^*(t_k)) + K_\epsilon)) + K_\epsilon\big) \end{aligned} \tag{3.55}$$

Since $T > 0$ may be selected arbitrarily, we select $T > \frac{\Delta_r}{1-\rho}$ so that $1 - \rho - \frac{\Delta_r}{T} > 0$. Then by Fact 1, we have that $\{X(t_k)\}_{k=0}^{\infty}$ is positive recurrent under the g-adaptive variant of π, and the queue lengths satisfy $\lim_{k\to\infty} \mathbb{E}\left[f(\sum_{ij} Q_{ij}(t_k))\right] < \infty$.

□

3.8.4 Proof of Lemma 3.3

Proof. By the assumption, the schedule is reconfigured to $\mathbf{\Pi}(t)$ at time t, and according to Condition 3.2 we have $W^{\pi}(t) = W_{\mathbf{\Pi}(t)}(\mathbf{Q}(t)) \geq W^*(t) - G$. It now suffices to show that at any time $\tau \in [t+1, t+T]$, the weight of $\mathbf{\Pi}(t)$ is large enough so that no schedule reconfiguration occurs.

First recall that at most one packet could depart from each queue in a time slot, and each schedule has at most N parallel connections. Along with (3.49), we obtain

$$\begin{aligned} &W_{\mathbf{\Pi}(t)}(\tau) = \Big\langle f(\mathbf{Q}(\tau)), \mathbf{\Pi}(t) \Big\rangle \\ \geq &W_{\mathbf{\Pi}(t)}(t) - NT \max_{i,j} \big(\max_{u\in[-T,0]} |f'\left(Q_{ij}(t)+u\right)| \big) \\ \geq &W^*(t) - G - NT \max_{i,j} \left(\epsilon g(f(Q_{ij}(t))) + K_\epsilon\right) \end{aligned} \tag{3.56}$$

On the other hand, since the arrival at each queue is upper bounded by $A_{\max}$ according to Assumption 2.1, we have an upper bound for the maximum weight given by

$$\begin{aligned} W^*(\tau) = W_{\mathbf{\Pi}^*(\tau)}(\tau) \leq &W_{\mathbf{\Pi}^*(\tau)}(t) + NA_{\max}T \max_{i,j} \Big(\max_{u\in[0,A_{\max}T]} |f'\left(Q_{ij}(t)+u\right)| \Big) \\ \leq &W^*(t) + NA_{\max}T \max_{i,j} \Big(\epsilon g(f(Q_{ij}(t))) + K_\epsilon \Big) \end{aligned} \tag{3.57}$$

Combining (3.56) and (3.57) we get an upper bound for the weight difference between

$\mathbf{\Pi}(t)$ and $W^{\pi}(\tau)$:

$$\begin{aligned} W^{\pi}(\tau) - W_{\mathbf{\Pi}(t)}(\tau) \leq & W^{*}(\tau) - W_{\mathbf{\Pi}(t)}(\tau) \\ \leq & N(A_{\max}+1)T\Big(\epsilon g\Big(f\Big(\max_{i,j} Q_{ij}(t)\Big)\Big) + K_{\epsilon}\Big) + G \\ \leq & N(A_{\max}+1)T\left(\epsilon g\left(W^{*}(t)\right) + K_{\epsilon}\right) + G \qquad (3.58) \end{aligned}$$

Similar to (3.56), since at most one packet could depart from each queue in a time slot, and since $g(\cdot)$ is a strictly increasing function, we obtain a lower bound for the threshold:

$$\begin{aligned} & g\left(W^{\pi}(\tau)\right) \geq g\left(W^{*}(\tau) - G\right) \\ \geq & g\Big(W^{*}(t) - NT\max_{i,j}\Big(\max_{u\in[-T,0]}\Big|f'\left(Q_{ij}(t)+u\right)\Big|\Big) - G\Big) \\ \geq & g\Big(W^{*}(t) - NT(\epsilon g(W^{*}(t)) + K_{\epsilon}) - G\Big) \qquad (3.59) \end{aligned}$$

Select ϵ to be sufficiently small, then since $g(\cdot)$ is a sublinear function, there exists a constant $M < \infty$ such that the right hand side of (3.59) is larger than the right hand side of (3.58) whenever $W^{*}(t) > M$. Therefore, if $W^{*}(t) > M$, then no reconfiguration occurs for any $\tau \in [t+1, t+T]$.

□

Chapter 4

Heavy Traffic Delay Analysis of Adaptive MaxWeight

4.1 Heavy Traffic Analysis

Studying queue length or delay performance for a queueing system such as a switch in general is challenging. Therefore, analyses of such systems are mostly considered within certain asymptotic regimes. In this chapter, we focus on the heavy traffic regime, and make use of the drift technique developed in [12]. The outline of the heavy traffic analysis for Adaptive MaxWeight is sketched as follows. We first establish the multi-dimensional state space collapse for the Adaptive MaxWeight. With the state space collapse result, we apply the drift technique to a Lyapunov function proposed in [29], and obtain an steady state queue length upper bound that is dependent on the expected schedule duration. We then characterize the relation between the expected schedule duration and the queue length, and use this relation to derive bounds on asymptotically tight steady state queue length bounds.

In this chapter, we are interested in the queue length behavior of switches with reconfiguration delay in the heavy traffic regime. Unlike some works that focus on arrival rate matrices with a single saturated port, we are interested in arrival rate matrices with all input ports and output ports being saturated. These arrival rate matrices lie on a face of the capacity region $\mathcal{C}$ described as the following:

$$\mathcal{F} = \Big\{\boldsymbol{\lambda} \in \mathbb{R}_+^{n\times n} : \sum_i \lambda_{ij} = 1, \sum_j \lambda_{ij} = 1, \forall i, j \in \{1, 2, \ldots, n\}\Big\}$$

For the heavy traffic analysis in this chapter, we consider a sequence of switch systems indexed by ϵ, where each switch system has an i.i.d. arrival traffic $\mathbf{A}^{(\epsilon)}(t)$ with mean and variance given by

$$\mathbb{E}[\mathbf{A}^{(\epsilon)}(t)] = \boldsymbol{\lambda}^{(\epsilon)} = \boldsymbol{\nu}(1-\epsilon), \quad \mathrm{Var}[\mathbf{A}^{(\epsilon)}(t)] = \big(\boldsymbol{\sigma}^{(\epsilon)}\big)^2 \tag{4.1}$$

where $\boldsymbol{\nu} \in \mathcal{F}$ and $\big(\boldsymbol{\sigma}^{(\epsilon)}\big)^2 \to \boldsymbol{\sigma}^2$ as $\epsilon \to 0$. The traffic load of each switch is $\rho = 1 - \epsilon$. Recall that $\mathcal{F}$ is the set of critically loaded rate matrix with all ports saturated. The sequence of switches considered here have arrival rate matrices that approach $\boldsymbol{\nu}$ as we take $\epsilon \to 0$.

4.1.1 Weak State Space Collapse

It was shown in [29] that for a switch with no reconfiguration delay, the MaxWeight scheduling exhibits a multi-dimensional state space collapse. To be specific, let $\mathbf{e}^{(i)}$ denote the matrix with i^{th} row being all ones and zeros everywhere else, and $\tilde{\mathbf{e}}^{(j}$ denote the matrix with j^{th} row being all ones and zeros everywhere else. As $\epsilon \to 0$, the steady state queue length $\bar{\mathbf{Q}}^{(\epsilon)}$ "concentrates" in the cone spanned by the matrices $\{\mathbf{e}^{(i)}\}_{i=1}^n \cup \{\tilde{\mathbf{e}}^{(j}\}_{j=1}^n$, *i.e.*

$$\mathcal{K} = \Big\{\mathbf{x} \in \mathbb{R}^{n\times n} : \mathbf{x} = \sum_i w_i \mathbf{e}^{(i)} + \sum_j \tilde{w}_j \tilde{\mathbf{e}}^{(j)}, \text{ where } w_i, \tilde{w}_j \in \mathbb{R}_+ \text{ for all } i, j\Big\},$$

in the sense that the projection of $\bar{\mathbf{Q}}^{(\epsilon)}$ onto $\mathcal{K}$ is the domniant component in $\bar{\mathbf{Q}}^{(\epsilon)}$. More specifically, for any $\mathbf{x} \in \mathbb{R}^{n\times n}$, define the projection of $\mathbf{x}$ on to $\mathcal{K}$ as

$$\mathbf{x}_{\|} = \arg\min_{\mathbf{y}\in\mathcal{K}} \|\mathbf{x} - \mathbf{y}\|$$

and define $\mathbf{x}_{\perp} = \mathbf{x} - \mathbf{x}_{\|}$. In the heavy traffic limit ($\epsilon \to 0$), all moments of $\bar{\mathbf{Q}}_{\perp}^{(\epsilon)}$ are bounded by a constant, and hence is a negligible component in $\bar{\mathbf{Q}}$ since it can be shown that $\|\bar{\mathbf{Q}}\|$ is $\Omega(1/\epsilon)$. This is referred as state space collapse (SSC) in [29].

In this paper, we consider a weaker notion of the state space collapse for switches with reconfiguration delay operated under the Adaptive MaxWeight policy.

Definition 4.1 (Weak State Space Collapse)**.** Given a sequence of switch systems $\mathbf{X}^{(\epsilon)}(t)$, parametrized by $0 < \epsilon < 1$. Suppose each switch system is positive recurrent and converges in distribution to a steady state random vector $\bar{\mathbf{X}}^{(\epsilon)} = (\bar{\mathbf{Q}}^{(\epsilon)}, \bar{\mathbf{s}}^{(\epsilon)}, \bar{r}^{(\epsilon)})$ We say that the sequence of switch systems exhbit a weak state space collapse if

$$\lim_{\epsilon\to 0} \frac{\mathbb{E}\big[\|\bar{\mathbf{Q}}_{\perp}^{(\epsilon)}\|\big]}{\mathbb{E}\big[\|\bar{\mathbf{Q}}^{(\epsilon)}\|\big]} = 0.$$

In the rest of this section, we would use Lemma 4.1 below to derive the weak state space collapse (WSSC) for switches with reconfiguration delay operated under the Adaptive MaxWeight policy. Lemma 4.1 is a T-step version of [4, Theorem 1] where T could be any fixed integer. The proof of this lemma could be done by simply replacing the transition probability to T-step transition probability in [4, Theorem 1], hence we omit the proof here.

Lemma 4.1. *Consider an irrdeucible and aperiodic Markov Chain $\{X(t)\}_{t\geq 0}$ over a countable state space $\mathcal{X}$, suppose $Z : \mathcal{X} \to \mathbb{R}_+$ is a nonnegative Lyapunov function. For any fixed integer*

$T > 0$, we define the T-step drift $\Delta^T Z(X)$ of Z at state X as

$$\Delta^T Z(X) = [Z(X(t+T)) - Z(X(t))] \mathbb{1}_{\{X(t)=X\}}.$$

Suppose for some $T > 0$, the T-step drift satisfies the following conditions C.1 and C.2:

C.1 There exists an $\eta > 0$, and a $\kappa < \infty$ such that for any $t = 1, 2, \dots$ and for all $X \in \mathcal{X}$ with $Z(X) \geq \kappa$,

$$\mathbb{E}[\Delta^T Z(X) | X(t) = X] \leq -\eta$$

C.2 There exists a $D < \infty$ such that for all $X \in \mathcal{X}$,

$$\Pr\left\{|\Delta^T Z(X)| \leq D\right\} = 1$$

If the Markov chain $\{X(t)\}_{t\geq 0}$ converges in distribution to a random variable $\bar{X}$, then

$$\mathbb{E}[Z(\bar{X})] \leq \kappa + \frac{2D^2}{\eta}.$$

With Lemma 4.1, we are now able to show the following proposition, which is essential to establish the WSSC result for the Adaptive MaxWeight policy.

Proposition 4.1. *Consider a set of switch systems with a fixed reconfiguration delay $\Delta_r > 0$, parametrized by $0 < \epsilon < 1$, all operated under Adaptive MaxWeight policy with hysteresis function $g(\cdot)$, where $g(\cdot)$ is sublinear and concave. Each system has arrival process $\mathbf{A}^{(\epsilon)}(t)$ as described in (4.1) and satisfying Assumptions 2.1- 2.3. The mean arrival rate vector $\boldsymbol{\lambda}^{(\epsilon)} = (1-\epsilon)\boldsymbol{\nu}$ for some fixed $\boldsymbol{\nu} \in \mathcal{F}$ such that $\nu_{\min} \triangleq \min_{ij} \nu_{ij} > 0$. Let the variance $\left(\boldsymbol{\sigma}^{(\epsilon)}\right)^2$ of the arrival process satisfy that $\|\boldsymbol{\sigma}^{(\epsilon)}\|^2 \leq \tilde{\sigma}^2$ for some $\tilde{\sigma}^2$ not dependent on ϵ.*

Let $\mathbf{X}^{(\epsilon)}(t) \in \mathcal{X}$ denote the process that determines each system, which is positive recurrent and hence converges to a steady state random vector in distribution, denoted as $\bar{\mathbf{X}}^{(\epsilon)} = (\bar{\mathbf{Q}}^{(\epsilon)}, \bar{\mathbf{S}}^{(\epsilon)}, \bar{r}^{(\epsilon)})$. Then for any fixed θ with $0 < \theta < 1/2$, and for each system with $0 < \epsilon \leq \nu_{\min}/4\|\boldsymbol{\nu}\|$, the steady state queue length satisfies

$$\mathbb{E}\Big[\|\bar{\mathbf{Q}}_{\perp}^{(\epsilon)}\| - \theta\|\bar{\mathbf{Q}}_{\|}^{(\epsilon)}\|\Big] \leq M_\theta, \tag{4.2}$$

where M_θ is a function of $\theta, \tilde{\sigma}, a_{\max}$, $\nu_{\min}$ and n, but is independent of ϵ.

The proof of Proposition 4.1 is given in appendix. Comparing Proposition 4.1 with [29, Proposition 2], we may see that we no longer have the guarantee that all moments of $\|\bar{\mathbf{Q}}_{\perp}^{(\epsilon)}\|$ are bounded here. However, we could still show that $\mathbb{E}[\|\bar{\mathbf{Q}}_{\perp}^{(\epsilon)}\|]$ is negligible comparing to $\mathbb{E}[\|\bar{\mathbf{Q}}^{(\epsilon)}\|]$ as $\epsilon \to 0$, hence we consider this as a weak version of SSC. In particular, notice that the constant M_θ is independent of ϵ, and that $\mathbb{E}[\|\bar{\mathbf{Q}}^{(\epsilon)}\|] \to \infty$ as $\epsilon \to 0$. Then since $\mathbb{E}[\|\bar{\mathbf{Q}}_{\perp}^{(\epsilon)}\|] \leq \theta\mathbb{E}[\|\bar{\mathbf{Q}}_{\|}^{(\epsilon)}\|] + M_\theta \leq \theta\mathbb{E}[\|\bar{\mathbf{Q}}^{(\epsilon)}\|] + M_\theta$ for any $\epsilon > 0$, we have $\lim\limits_{\epsilon\to 0} \frac{\mathbb{E}\left[\|\bar{\mathbf{Q}}_{\perp}^{(\epsilon)}\|\right]}{\mathbb{E}\left[\|\bar{\mathbf{Q}}^{(\epsilon)}\|\right]} \leq \theta$ for any $\theta > 0$. Therefore, we may conclude that

$$\lim_{\epsilon\to 0} \frac{\mathbb{E}\big[\|\bar{\mathbf{Q}}_{\perp}^{(\epsilon)}\|\big]}{\mathbb{E}\big[\|\bar{\mathbf{Q}}^{(\epsilon)}\|\big]} = 0. \tag{4.3}$$

4.2 Queue Length Bound of Adaptive MaxWeight in Heavy Traffic Regime

With the WSSC result from the previous subsection, we now utilize Lyapunov drift analysis similar to [29] to derive an upper bound on the steady state queue length $\mathbb{E}[\sum_{ij} \bar{\mathbf{Q}}_{ij}]$.

Theorem 4.1. *Consider a set of switch systems with a fixed reconfiguration delay $\Delta_r > 0$, parametrized by $0 < \epsilon < 1$, all operated under Adaptive MaxWeight policy with hysteresis function $g(\cdot)$, where $g(\cdot)$ is sublinear and concave. Each system has arrival process $\mathbf{a}^{(\epsilon)}(t)$ as* *described in (4.1). The mean arrival rate vector* $\boldsymbol{\lambda}^{(\epsilon)} = (1-\epsilon)\boldsymbol{\nu}$ *for some fixed* $\boldsymbol{\nu} \in \mathcal{F}$ *such that* $\nu_{\min} \overset{\Delta}{=} \min_{ij} \nu_{ij} > 0$*. Let the variance* $\big(\boldsymbol{\sigma}^{(\epsilon)}\big)^2$ *of the arrival process satisfy that* $\|\boldsymbol{\sigma}^{(\epsilon)}\|^2 \leq \tilde{\boldsymbol{\sigma}}^2$ *for some* $\tilde{\boldsymbol{\sigma}}^2$ *not dependent on* ϵ*. For each system with* $0 < \epsilon \leq \nu_{\min}/4\|\boldsymbol{\nu}\|$*, the steady state of the Markov chain* $\bar{\mathbf{X}}^{(\epsilon)} = (\bar{\mathbf{Q}}^{(\epsilon)}, \bar{\mathbf{S}}^{(\epsilon)}, \bar{r}^{(\epsilon)})$ *satisfies*

$$\Big(\epsilon - \Pr{}_{\bar{\mathbf{X}}}\{r(t)>0\}\Big)\Big(\mathbb{E}\Big[\sum_{ij}\bar{Q}_{ij}^{(\epsilon)}(t)\Big] - 2n^3\mathbb{E}\Big[\|\bar{\mathbf{Q}}_{\perp}^{(\epsilon)}(t)\|\Big]\Big) \leq (1-\frac{1}{2n})\|\tilde{\boldsymbol{\sigma}}\|^2 + B(\epsilon, n) \tag{4.4}$$

where $\lim_{\epsilon\to 0} B(\epsilon, n) = 0$*.*

Proof. For simplicity of the notation, we drop the superscript (ϵ) in the following proof. We also use $\mathbb{E}_{\bar{\mathbf{X}}}[\,\cdot\,]$ as a short hand notation for $\mathbb{E}[\,\cdot\,|X = \bar{\mathbf{X}}]$. Now consider the following Lyapunov function from [29]:

$$V(\mathbf{X}) = \sum_i \Big(\sum_j Q_{ij}\Big)^2 + \sum_i \Big(\sum_i Q_{ij}\Big)^2 - \frac{1}{n}\Big(\sum_{ij} Q_{ij}\Big)^2$$

It may be shown using Lemma 4.1 that for steady state $\bar{\mathbf{X}}$, the expectation $\mathbb{E}[V(\bar{\mathbf{X}})]$ is finite. We thus have zero drift for $V(\bar{\mathbf{X}})$ at steady state:

$$\mathbb{E}_{\bar{\mathbf{X}}}\Big[\Delta V(\mathbf{X}(t))\Big] = \mathbb{E}_{\bar{\mathbf{X}}}\Big[V(\mathbf{X}(t+1)) - V(\mathbf{X}(t))\Big] = 0$$

We now evaluate the above drift terms with the queue length dynamics (5.1) and rewrite the expression as

$$\mathcal{T}_1 = \mathcal{T}_2 + \mathcal{T}_3 + \mathcal{T}_4$$

where

$$\begin{aligned}
\mathcal{T}_1 =&\mathbb{E}_{\bar{\mathbf{X}}}\Bigg[2\sum_i\big(\sum_j Q_{ij}(t)\big)\big(\sum_j s_{ij}(t)\mathbb{1}_{\{r(t)=0\}}-a_{ij}(t)\big)\\
&+2\sum_j\Big(\sum_i Q_{ij}(t)\Big)\Big(\sum_i s_{ij}(t)\mathbb{1}_{\{r(t)=0\}}-a_{ij}(t)\Big)\\
&-\frac{2}{n}\Big(\sum_{ij} Q_{ij}(t)\Big)\Big(\sum_{ij} s_{ij}(t)\mathbb{1}_{\{r(t)=0\}}-a_{ij}(t)\Big)\Bigg]\\
\mathcal{T}_2 =&\mathbb{E}_{\bar{\mathbf{X}}}\Bigg[\sum_i\Big(\sum_j a_{ij}(t)-s_{ij}(t)\mathbb{1}_{\{r(t)=0\}}\Big)^2+\sum_j\Big(\sum_i a_{ij}(t)-s_{ij}(t)\mathbb{1}_{\{r(t)=0\}}\Big)^2\\
&-\frac{1}{n}\Big(\sum_{ij} a_{ij}(t)-s_{ij}(t)\mathbb{1}_{\{r(t)=0\}}\Big)^2\Bigg]\\
\mathcal{T}_3 =&\mathbb{E}_{\bar{\mathbf{X}}}\Bigg[-\sum_i\Big(\sum_j u_{ij}(t)\Big)^2-\sum_j\Big(\sum_i u_{ij}(t)\Big)^2+\frac{1}{n}\Big(\sum_{ij} u_{ij}(t)\Big)^2\Bigg]\\
\mathcal{T}_4 =&\mathbb{E}_{\bar{\mathbf{X}}}\Bigg[2\sum_i\Big(\sum_j Q_{ij}(t+1)\Big)\Big(\sum_j u_{ij}(t)\Big)+2\sum_j\Big(\sum_i Q_{ij}(t+1)\Big)\Big(\sum_i u_{ij}(t)\Big)\\
&-\frac{2}{n}\Big(\sum_{ij} Q_{ij}(t+1)\Big)\Big(\sum_{ij} u_{ij}(t)\Big)\Bigg]
\end{aligned}$$

We now simplify each term:

$$\begin{aligned}
\mathcal{T}_1 =&\mathbb{E}_{\bar{\mathbf{X}}}\Big[2\Big(\sum_{ij} Q_{ij}(t)\Big)\Big(\mathbb{1}_{\{r(t)=0\}}-(1-\epsilon)\Big)\Big]=\mathbb{E}_{\bar{\mathbf{X}}}\Big[2\Big(\sum_{ij} Q_{ij}(t)\Big)\Big(\epsilon-\mathbb{1}_{\{r(t)>0\}}\Big)\Big]\\
=&2\epsilon\mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij} Q_{ij}(t)\Big]-2\mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij} Q_{ij}(t)\mathbb{1}_{\{r(t)>0\}}\Big]\\
=&2\epsilon\mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij} Q_{ij}(t)\Big]-2\Big(\mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij} Q_{ij}(t)\Big]-\mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij} Q_{ij}(t)\Big|r(t)=0\Big]\Pr{}_{\bar{\mathbf{X}}}\{r(t)=0\}\Big)\\
=&2\Big(\epsilon-\Pr{}_{\bar{\mathbf{X}}}\{r(t)>0\}\Big)\mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij} Q_{ij}(t)\Big]\\
&-2\Big(\mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij} Q_{ij}(t)\Big]-\mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij} Q_{ij}(t)\Big|r(t)=0\Big]\Big)\Big(1-\Pr{}_{\bar{\mathbf{X}}}\{r(t)>0\}\Big) \qquad (4.5)
\end{aligned}$$

Note that with the ergodicity of the Markov chain $\mathbf{X}(t)$, we have that $\mathbb{E}_{\bar{\mathbf{X}}}[\sum_{ij} Q_{ij}(t)]$ and $\mathbb{E}_{\bar{\mathbf{X}}}[\sum_{ij} Q_{ij}(t)|r(t) = 0]$ equal to the time average of sum queue length and that under $r(t) = 0$, respectively. Also note that the probability of $r(t) = 0$ approaches 1 as $\epsilon \to 0$, which means that the time average under $r(t) = 0$ only excludes a diminishing number of time instances. Then since the change in sum queue length $|\sum_{ij} Q_{ij}(t+1) - \sum_{ij} Q_{ij}(t)| < n^2 a_{\max}$ is bounded, we have the difference $\mathbb{E}_{\bar{\mathbf{X}}}[\sum_{ij} Q_{ij}(t)] - \mathbb{E}_{\bar{\mathbf{X}}}[\sum_{ij} Q_{ij}(t)|r(t) = 0] \to 0$ as $\epsilon \to 0$.

For term $\mathcal{T}_2$, since $\mathbb{E}\Big[\sum_i(\sum_j a_{ij}(t) - 1)^2\Big] = \|\boldsymbol{\sigma}\|^2 + n\epsilon^2$ and $\mathbb{E}\Big[(\sum_{ij} a_{ij}(t) - n)^2\Big] = \|\boldsymbol{\sigma}\|^2 + n^2\epsilon^2$, we have

$$\begin{aligned}
\mathcal{T}_2 =& \mathbb{E}_{\bar{\mathbf{X}}}\Bigg[\Big(\sum_i(\sum_j a_{ij}(t) - 1)^2 + \sum_j(\sum_i a_{ij}(t) - 1)^2 - \frac{1}{n}(\sum_{ij} a_{ij}(t) - n)^2\Big) \\
&+ \Big(\sum_i(2\sum_j a_{ij}(t) - 1) + \sum_j(2\sum_i a_{ij}(t) - 1) - \frac{1}{n}(2n\sum_{ij} a_{ij}(t) - n^2)\Big)\mathbb{1}_{\{r(t)>0\}}\Bigg] \\
=& \mathbb{E}_{\bar{\mathbf{X}}}\Big[\Big(2(\|\boldsymbol{\sigma}\|^2 + n\epsilon^2) - \frac{1}{n}(\|\boldsymbol{\sigma}\|^2 + n^2\epsilon^2)\Big) + \Big(2n(1-\epsilon) - n\Big)\mathbb{1}_{\{r(t)>0\}}\Big] \\
=& \Big((2 - \frac{1}{n})\|\boldsymbol{\sigma}\|^2 + n\epsilon^2\Big) + n(1 - 2\epsilon)\Pr{}_{\bar{\mathbf{X}}}\{r(t) > 0\}
\end{aligned} \tag{4.6}$$

For term $\mathcal{T}_3$, since $u_{ij}(t) \leq s_{ij}(t)$, we have $\sum_i u_{ij} \leq 1, \sum_j u_{ij} \leq 1$ and $\sum_{ij} u_{ij} \leq n$. Therefore,

$$\mathcal{T}_3 \leq \mathbb{E}_{\bar{\mathbf{X}}}\Bigg[\frac{1}{n}\Big(\sum_{ij} u_{ij}(t)\Big)^2\Bigg] \leq \mathbb{E}_{\bar{\mathbf{X}}}\Bigg[\sum_{ij} u_{ij}(t)\Bigg] \tag{4.7}$$

The above expression is the expected sum of unused services between schedule reconfiguration time instance. One way to determine this value is to set the drift of $\sum_{ij} \bar{\mathbf{Q}}_{ij}$ to zero. We may then obtain

$$\mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij} u_{ij}(t)\Big] = n\Big(\epsilon - \Pr{}_{\bar{\mathbf{X}}}\{r(t) > 0\}\Big) \tag{4.8}$$

Hence we have

$$\mathcal{T}_3 \leq n\Big(\epsilon - \operatorname{Pr}_{\bar{\mathbf{X}}}\{r(t) > 0\}\Big) \tag{4.9}$$

We now utilize the relation $u_{ij}(t)Q_{ij}(t+1) = 0$ for any i, j to bound the term $\mathcal{T}_4$. The expression below follows along the same lines in [29]:

$$\begin{aligned}
\mathcal{T}_4 =& 2\mathbb{E}_{\bar{\mathbf{X}}}\left[\sum_{ij} u_{ij}(t)\Big(\sum_{j'} Q_{ij'}(t+1) + \sum_{i'} Q_{i'j}(t+1) - \frac{1}{n}\sum_{i'j'} Q_{i'j'}(t+1)\Big)\right] \\
=& 2\mathbb{E}_{\bar{\mathbf{X}}}\Big[\Big\langle \mathbf{u}(t), -n\mathbf{Q}_{\perp}(t+1) + \sum_i \langle \mathbf{Q}_{\perp}(t+1), \mathbf{e}^{(i)}\rangle \mathbf{e}^{(i)} + \sum_j \langle \mathbf{Q}_{\perp}(t+1), \tilde{\mathbf{e}}^{(j)}\rangle \tilde{\mathbf{e}}^{(j)} \\
& - \frac{1}{n}\langle \mathbf{Q}_{\perp}(t+1), \mathbf{1}\rangle \mathbf{1}\Big\rangle\Big] \\
\leq& 2\mathbb{E}_{\bar{\mathbf{X}}}\Big[\Big\langle \mathbf{u}(t), -n\mathbf{Q}_{\perp}(t+1) - \frac{1}{n}\langle \mathbf{Q}_{\perp}(t+1), \mathbf{1}\rangle \mathbf{1}\Big\rangle\Big] \\
\leq& 2\mathbb{E}_{\bar{\mathbf{X}}}\Big[\big\|\mathbf{u}(t)\big\|\big\| - n\mathbf{Q}_{\perp}(t+1) - \frac{1}{n}\langle \mathbf{Q}_{\perp}(t+1), \mathbf{1}\rangle \mathbf{1}\big\|\Big]
\end{aligned}$$

where the last inequality follows from Cauchy-Schwartz inequality.

Note that

$$\begin{aligned}
\big\| - n\mathbf{Q}_{\perp}(t+1) - \frac{1}{n}\langle \mathbf{Q}_{\perp}(t+1), \mathbf{1}\rangle \mathbf{1}\big\| \overset{(a)}{\leq}& n\|\mathbf{Q}_{\perp}(t+1)\| + \frac{1}{n}|\langle \mathbf{Q}_{\perp}(t+1), \mathbf{1}\rangle|\|\mathbf{1}\| \\
\overset{(b)}{\leq}& n\|\mathbf{Q}_{\perp}(t+1)\| + \frac{\|\mathbf{1}\|\|\mathbf{1}\|}{n}\|\mathbf{Q}_{\perp}(t+1)\| \\
=& 2n\|\mathbf{Q}_{\perp}(t+1)\|
\end{aligned}$$

where (a) follows from triangle inequality, and (b) follows from Cauchy-Schwartz inequality.

We then obtain

$$\begin{aligned}\mathcal{T}_4 \leq& 4n\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{u}(t)\|\|\mathbf{Q}_\perp(t+1)\|\Big]\\ \leq& 4n\mathbb{E}_{\bar{\mathbf{X}}}\Big[\Big(\sum_{ij}u_{ij}(t)\Big)\Big(\|\mathbf{Q}_\perp(t)\|+na_{\max}\Big)\Big]\\ =& 4n\mathbb{E}_{\bar{\mathbf{X}}}\Big[\Big(\sum_{ij}u_{ij}(t)\Big)\|\mathbf{Q}_\perp(t)\|\Big]+4n\mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij}u_{ij}(t)\Big]na_{\max}\end{aligned}$$

Since $\sum_{ij}u_{ij}(t)\leq n$, we may write $\sum_{ij}u_{ij}(t)\leq n\mathbb{1}_{\{\sum_{ij}u_{ij}(t)>0\}}$ and thus

$$\begin{aligned}&\mathbb{E}_{\bar{\mathbf{X}}}\Big[\Big(\sum_{ij}u_{ij}(t)\Big)\|\mathbf{Q}_\perp(t)\|\Big]\\ \leq& n\mathbb{E}_{\bar{\mathbf{X}}}\Big[\mathbb{1}_{\{\sum_{ij}u_{ij}(t)>0\}}\|\mathbf{Q}_\perp(t)\|\Big]\\ =& n\Big(\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{Q}_\perp(t)\|\Big]-\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{Q}_\perp(t)\|\mathbb{1}_{\{\sum_{ij}u_{ij}(t)=0\}}\Big]\Big)\\ =& n\Big(\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{Q}_\perp(t)\|\Big]\mathbb{E}_{\bar{\mathbf{X}}}\Big[\mathbb{1}_{\{\sum_{ij}u_{ij}(t)>0\}}\Big]+\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{Q}_\perp(t)\|\Big]\mathbb{E}_{\bar{\mathbf{X}}}\Big[\mathbb{1}_{\{\sum_{ij}u_{ij}(t)=0\}}\Big]\\ &-\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{Q}_\perp(t)\|\mathbb{1}_{\{\sum_{ij}u_{ij}(t)=0\}}\Big]\Big)\\ \leq& n\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{Q}_\perp(t)\|\Big]\mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij}u_{ij}(t)\Big]\\ &+n\Big(\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{Q}_\perp(t)\|\Big]-\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{Q}_\perp(t)\|\Big|\sum_{ij}u_{ij}(t)=0\Big]\Big)\Pr{}_{\bar{\mathbf{X}}}\Big\{\sum_{ij}u_{ij}(t)=0\Big\}\end{aligned}$$

We then have

$$\begin{aligned}\mathcal{T}_4 \leq& 4n^3\Big(\epsilon-\Pr{}_{\bar{\mathbf{X}}}\{r(t)>0\}\Big)\Big(\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{Q}_\perp(t)\|\Big]+a_{\max}\Big)\\ &+4n^2\Big(\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{Q}_\perp(t)\|\Big]-\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{Q}_\perp(t)\|\Big|\sum_{ij}u_{ij}(t)=0\Big]\Big)\end{aligned}\tag{4.10}$$

Similar to the second term in $\mathcal{T}_1$, by the ergodicity of the Markov chain $\mathbf{X}(t)$, and that the probability of $\sum_{ij}u_{ij}(t)=0$ approaches 1 as $\epsilon\to 0$. Then since the change in $\|\mathbf{Q}_\perp(t)\|$ is bounded, *i.e.* $\big|\|\mathbf{Q}_\perp(t+1)\|-\|\mathbf{Q}_\perp(t)\|\big|<na_{\max}$, we have that $\mathbb{E}_{\bar{\mathbf{X}}}[\|\mathbf{Q}_\perp(t)\|]-$

$\mathbb{E}_{\bar{\mathbf{X}}}[\|\mathbf{Q}_{\perp}(t)\|| \sum_{ij} u_{ij}(t) = 0] \to 0$ as $\epsilon \to 0$.

Combining (4.5)-(4.10), we obtain

$$\Big(\epsilon - \Pr{}_{\bar{\mathbf{X}}}\{r(t) > 0\}\Big)\Big(\mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij} Q_{ij}(t)\Big] - 2n^3\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{Q}_{\perp}(t)\|\Big]\Big) \leq (1 - \frac{1}{2n})\|\boldsymbol{\sigma}\|^2 + B(\epsilon, n) \tag{4.11}$$

where $B = \Big(\mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij} Q_{ij}(t)\Big] - \mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij} Q_{ij}(t)\Big| r(t) = 0\Big]\Big) + \frac{n\epsilon(1+\epsilon)}{2} +$ $2n^2\Big(\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{Q}_{\perp}(t)\|\Big] - \mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\mathbf{Q}_{\perp}(t)\|\Big| \sum_{ij} u_{ij}(t) = 0\Big]\Big)$ and that $\lim_{\epsilon \to 0} B(\epsilon, n) = 0$.

□

4.2.1 Expected Schedule Duration

In the previous subsection, we derived a bound on steady state queue lengths. Note that this bound depends on the probability of the switch in reconfiguration $\Pr{}_{\bar{\mathbf{X}}}\{r(t) > 0\}$. In this subsection, we derive the mean schedule duration in order to evaluate this probability.

Recall that in the previous subsection, we set the drift of $\sum_{ij} \bar{q}_{ij}$ to zero to obtain (4.8), which is an expression of the total unused service. Here we consider the drift of another Lyapunov function to obtain a different expression for the total unused service, and combined the two expressions to derive the expected schedule duration.

In this subsection, we consider the original Markov chain $\mathbf{X}^{(\epsilon)}(t)$ sampled at the reconfiguration times $\{t_k^S\}$ and denote it as $\mathbf{X}_k^{(\epsilon)} = \mathbf{X}^{(\epsilon)}(t_k^S)$. Note that $\{t_k^S\}$ is a stopping time with respect to $\mathbf{X}^{(\epsilon)}(t)$, hence by the strong Markov property, $\mathbf{X}_k^{(\epsilon)}$ is also a Markov chain. Furthermore, the positive recurrence of $\mathbf{X}^{(\epsilon)}(t)$ implies the positive recurrence of $\mathbf{X}_k^{(\epsilon)}$. We then denote the steady state distribution of $\mathbf{X}_k^{(\epsilon)}$ as $\hat{\mathbf{X}}^{(\epsilon)} = (\hat{\mathbf{Q}}^{(\epsilon)}, \hat{\mathbf{S}}^{(\epsilon)}, \hat{r}^{(\epsilon)})$.

Theorem 4.2. *Consider a switch system with a fixed reconfiguration delay $\Delta_r > 0$, and the arrival process $\mathbf{a}(t)$ is as described in (4.1) with the mean arrival rate vector given by $\boldsymbol{\lambda} = (1 - \epsilon)\boldsymbol{\nu}$ for some $\boldsymbol{\nu} \in \mathcal{F}$. Suppose the switch system is operated under Adaptive MaxWeight policy with* *hysteresis function $g(\cdot)$, where $g(\cdot)$ is sublinear and concave. Define the MaxWeight function $W^*(\mathbf{X}) = \max_{\mathbf{S}\in\mathcal{S}}\langle \mathbf{Q}, \mathbf{S}\rangle$ for each state $\mathbf{X} = (\mathbf{Q}, \mathbf{s}, r) \in \mathcal{X}$, and let $\hat{\mathbf{W}}^* = W^*(\hat{\mathbf{X}})$, then the following relation holds:*

$$\mathbb{E}\Big[t_{k+1} - t_k \Big| \mathbf{X}(t_k) = \hat{\mathbf{X}}\Big] = \frac{\mathbb{E}\Big[g(\hat{\mathbf{W}}^*) + \delta_W\Big]}{(n-\alpha)(1-\epsilon)} \tag{4.12}$$

where δ_W satisfies $0 \le \delta_W < na_{\max}$, and $\alpha = \big\langle \boldsymbol{\nu}, \mathbb{E}[\mathbf{s}(t_k)|\mathbf{X}(t_k) = \hat{\mathbf{X}}]\big\rangle$.

Proof. Define the Lyapunov function W on state $\mathbf{X} = (\mathbf{Q}, \mathbf{s}, r)$:

$$W(\mathbf{X}) = \Big\langle \mathbf{Q}, \mathbf{s} \Big\rangle = \sum_{ij} Q_{ij} s_{ij}$$

which is simply the schedule weight function. Note that $W(\mathbf{X}) \le \sum_{ij} Q_{ij}$, hence the steady state mean of $W(\mathbf{X})$ is finite. We may then set the drift of $W(\mathbf{X})$ between two schedule reconfiguration time instance to be zero:

$$\begin{aligned}
&\mathbb{E}\Bigg[W(\mathbf{X}(t_{k+1})) - W(\mathbf{X}(t_k))\Bigg|\mathbf{X}(t_k) = \hat{\mathbf{X}}\Bigg] \\
=&\mathbb{E}\Bigg[\sum_{t=t_k}^{t_{k+1}-1} \Delta W(\mathbf{X}(t))\Bigg|\mathbf{X}(t_k) = \hat{\mathbf{X}}\Bigg] \\
=&\mathbb{E}\Bigg[\sum_{t=t_k}^{t_{k+1}-1} \Big(\sum_{ij} Q_{ij}(t+1)s_{ij}(t+1) - \sum_{ij} Q_{ij}(t)s_{ij}(t)\Big)\Bigg|\mathbf{X}(t_k) = \hat{\mathbf{X}}\Bigg] \\
\overset{(a)}{=}&\mathbb{E}\Bigg[\sum_{ij} Q_{ij}(t_{k+1})\big(s_{ij}(t_{k+1}) - s_{ij}(t_k)\big) + \sum_{t=t_k}^{t_{k+1}-1}\sum_{ij}\big(Q_{ij}(t+1) - Q_{ij}(t)\big)s_{ij}(t_k)\Bigg|\mathbf{X}(t_k) = \hat{\mathbf{X}}\Bigg] \\
\overset{(b)}{=}&\mathbb{E}\Bigg[g(W^*(t_{k+1})) + \delta_W + \sum_{t=t_k}^{t_{k+1}-1}\big(\sum_{ij}\lambda_{ij}s_{ij}(t_k) - n\mathbb{1}_{\{r(t)=0\}}\big) + \sum_{t=t_k}^{t_{k+1}-1}\sum_{ij} u_{ij}(t)\Bigg|\mathbf{X}(t_k) = \hat{\mathbf{X}}\Bigg] \\
=&0
\end{aligned} \tag{4.13}$$

Since the schedule remains $\mathbf{s}(t_k)$ between t_k and t_{k+1}, and changes to $\mathbf{s}(t_{k+1})$ at time t_{k+1}, we have (a). Then by the definition of the Adaptive MaxWeight, the weight difference exceeds $g(W^*(t_{k+1}))$ at time t_{k+1} and thus we may write $\sum_{ij} Q_{ij}(t_{k+1})\Big(s_{ij}(t_{k+1}) - s_{ij}(t_k)\Big) = g(W^*(t_{k+1})) + \delta_W$, where $0 \leq \delta_W < na_{\max}$, we have (b).

Since the arrival processes are independent from the scheduling decisions, we have $\mathbb{E}\big[\sum_{ij} \lambda_{ij} s_{ij}(t_k) \big| \mathbf{X}(t_k) = \hat{\mathbf{X}}\big] = (1-\epsilon)\big\langle \boldsymbol{\nu}, \mathbb{E}[\mathbf{s}(t_k)|\mathbf{X}(t_k) = \hat{\mathbf{X}}]\big\rangle$. Now define

$$\alpha = \big\langle \boldsymbol{\nu}, \mathbb{E}[\mathbf{s}(t_k)|\mathbf{X}(t_k) = \hat{\mathbf{X}}]\big\rangle, \tag{4.14}$$

and we may then write (4.13) as

$$\begin{aligned}\mathbb{E}\Bigg[\sum_{t=t_k}^{t_{k+1}-1} \sum_{ij} u_{ij}(t) \Bigg| \mathbf{X}(t_k) = \hat{\mathbf{X}}\Bigg] =& \Big(n - \alpha(1-\epsilon)\Big)\mathbb{E}\Big[t_{k+1} - t_k \Big| \mathbf{X}(t_k) = \hat{\mathbf{X}}\Big] \\ & - n\Delta_r - \mathbb{E}\Big[g(\hat{\mathbf{W}}^*) + \delta_W\Big]\end{aligned} \tag{4.15}$$

On the other hand, we may set the drift of $\sum\limits_{ij} Q_{ij}$ to zero and obtain

$$\mathbb{E}\Bigg[\sum_{t=t_k}^{t_{k+1}-1} \sum_{ij} u_{ij}(t) \Bigg| \mathbf{X}(t_k) = \hat{\mathbf{X}}\Bigg] = n\Big(\epsilon\mathbb{E}\Big[t_{k+1} - t_k \Big| \mathbf{X}(t_k) = \hat{\mathbf{X}}\Big] - \Delta_r\Big) \tag{4.16}$$

Combining (4.15) and (4.16), we then have

$$\mathbb{E}\Big[t_{k+1} - t_k \Big| \mathbf{X}(t_k) = \hat{\mathbf{X}}\Big] = \frac{\mathbb{E}\Big[g(\hat{\mathbf{W}}^*) + \delta_W\Big]}{(n-\alpha)(1-\epsilon)} \tag{4.17}$$

□

In the rest of this section, we utilize Theorems 4.1 and 4.2 to derive bounds on the expected sum of queue lengths for switches with reconfiguration operating under the Adaptive MaxWeight policy.

We first start with a lower bound on the sum of queue lengths. By the non-negativity of the unused service $u_{ij}(t)$, we derive the following from (4.16):

$$\epsilon\mathbb{E}\Big[t_{k+1}-t_k\Big|\mathbf{X}(t_k)=\hat{\mathbf{X}}\Big]-\Delta_r=\frac{1}{n}\mathbb{E}_{\hat{\mathbf{X}}}\Bigg[\sum_{t=t_k}^{t_{k+1}-1}\sum_{ij}u_{ij}(t)\Bigg|\mathbf{X}(t_k)=\hat{\mathbf{X}}\Bigg]\geq 0$$

Using Theorem 4.2, we thus have a lower bound on the expected schedule duration.

$$\mathbb{E}\Big[g(\hat{\mathbf{W}}^*)+\delta_W\Big]=(n-\alpha)(1-\epsilon)\mathbb{E}_{\hat{\mathbf{X}}}\Big[t_{k+1}-t_k\Big|\mathbf{X}(t_k)=\hat{\mathbf{X}}\Big]\geq\frac{(n-\alpha)(1-\epsilon)\Delta_r}{\epsilon}\quad(4.18)$$

The lower bound on the expected schedule duration then implies a lower bound on the expected maximum weight through Jensen's inequality since $g(\cdot)$ is a concave function:

$$\mathbb{E}\Big[\sum_{ij}\hat{\mathbf{Q}}_{ij}\Big]\geq\mathbb{E}\Big[\hat{\mathbf{W}}^*\Big]\geq g^{-1}\Big(\mathbb{E}\Big[g(\hat{\mathbf{W}}^*)\Big]\Big)\geq g^{-1}\bigg(\frac{(n-\alpha)(1-\epsilon)\Delta_r}{\epsilon}-\mathbb{E}\Big[\delta_W\Big]\bigg)\quad(4.19)$$

Since $\mathbb{E}\big[\delta_W\big]<na_{\max}$ by definition, hence in the heavy traffic regime ($\epsilon\downarrow 0$), we have $\mathbb{E}\big[\sum_{ij}\hat{\mathbf{Q}}_{ij}\big]\sim\Omega(g^{-1}(1/\epsilon))$.

By the ergodicity of the Markov chain, we may use the expected schedule duration to derive the probability that the switch is in reconfiguration delay as

$$\Pr{}_{\bar{X}}\{r(t)>0\}=\mathbb{E}_{\bar{\mathbf{X}}}\Big[\mathbb{1}_{\{r(t)>0\}}\Big]=\frac{\Delta_r}{\mathbb{E}[t_{k+1}-t_k|\mathbf{X}(t_k)=\hat{\mathbf{X}}]}=\frac{(n-\alpha)(1-\epsilon)\Delta_r}{\mathbb{E}[g(\hat{\mathbf{W}}^*)+\delta_W]}\quad(4.20)$$

We may then apply (4.20) into (4.4). For the simplicity of notation, we denote $\hat{\mathbf{g}}=\mathbb{E}\Big[g(\hat{W}^*)+\delta_W\Big]$, $\bar{\mathbf{Q}}_s=\mathbb{E}_{\bar{\mathbf{X}}}\Big[\sum_{ij}\bar{q}_{ij}\Big]-4n^3\mathbb{E}_{\bar{\mathbf{X}}}\Big[\|\bar{q}_\perp\|\Big]$, $\beta=(1-\frac{1}{2n})\tilde{\sigma}^2+B(\epsilon,n)$ and obtain

$$\begin{aligned}&\Big(\epsilon\hat{\mathbf{g}}-(n-\alpha)(1-\epsilon)\Delta_r\Big)\bar{\mathbf{Q}}_s\leq\beta\hat{\mathbf{g}}\\ \Rightarrow&\Big(\epsilon-\frac{\beta}{\bar{\mathbf{Q}}_s}\Big)\hat{\mathbf{g}}\leq(n-\alpha)(1-\epsilon)\Delta_r\end{aligned}$$

We thus have

$$\hat{\mathbf{g}} \leq (n-\alpha)(1-\epsilon)\Delta_r \frac{\bar{\mathbf{Q}}_s}{\epsilon\bar{\mathbf{Q}}_s - \beta} = \frac{(n-\alpha)(1-\epsilon)\Delta_r}{\epsilon}\Big(1 + \frac{\beta}{\epsilon\bar{\mathbf{Q}}_s - \beta}\Big)$$

Note that from the lower bound (4.19) and the WSSC result (4.3), we have $\bar{\mathbf{Q}}_s \sim \Omega\Big(g^{-1}(1/\epsilon)\Big)$. Therefore $\frac{\beta}{\epsilon\bar{\mathbf{Q}}_s - \beta} \to 0$ as $\epsilon \downarrow 0$, and thus

$$\limsup_{\epsilon\downarrow 0}\ \epsilon\hat{\mathbf{g}} = (n-\alpha)\Delta_r \tag{4.21}$$

Along with the lower bound from (4.19), and that $\mathbb{E}[\delta_W] < na_{\max}$, we have

$$\lim_{\epsilon\downarrow 0} \epsilon\mathbb{E}\Big[g(\hat{\mathbf{W}}^*)\Big] = (n-\alpha)\Delta_r \tag{4.22}$$

With the relation between the total queue length and the maximum weight $\sum_{ij} Q_{ij}(t) \leq nW^*(t)$, we then have the following upper bound:

$$\limsup_{\epsilon\downarrow 0}\ \epsilon\mathbb{E}\Big[g\Big(\sum_{ij}\hat{\mathbf{Q}}_{ij}\Big)\Big] \leq \lim_{\epsilon\downarrow 0}\epsilon\mathbb{E}\Big[g(n\hat{\mathbf{W}}^*)\Big] \leq \lim_{\epsilon\downarrow 0} n\epsilon\mathbb{E}\Big[g(\hat{\mathbf{W}}^*)\Big] = n(n-\alpha)\Delta_r \tag{4.23}$$

While the upper bound does not give an exact bound on the expected sum of queue length $\mathbb{E}[\sum_{ij}\hat{\mathbf{Q}}_{ij}]$, we may still consider $g^{-1}(n(n-\alpha)\Delta_r/\epsilon))$ as an approximate upper bound, which becomes more accurate as the hysteresis function g is close to a linear function.

In fact, this is also the regime of interest when pursuing an optimal queue length bound, since $g^{-1}(n(n-\alpha)\Delta_r/\epsilon))$ becomes lower as g gets closer to a linear function. In other words, if we consider $g(x) = (1-\gamma)x^{1-\delta}$ as in [50] and take $\delta \to 0$, we not only get a tighter asymptotic bounds but also a better scaling. The only caveat here is that selecting g as exactly a linear function does not fit the analysis in this paper. In fact, it is even unclear whether throughput optimal could be guaranteed if g is linear.

4.3 Benchmark Queue Length Behavior under Reconfiguration Delay

In this section, we derive some benchmark queue length behavior of switches with reconfiguration delay for the Adaptive MaxWeight policy to compare with. We start with a queue length lower bound for switch systems with reconfiguration delay, which determines a limit on the performance for any scheduling policy. In a latter subsection, we then derive a queue length upper bound for a benchmark policy known as the Fixed Frame MaxWeight (FFMW) [25] policy. Although it is shown that the FFMW policy may achieve the optimal queue length scaling in the heavy traffic regime, the optimality would require the perfect knowledge of the traffic load, which restricts its feasibility in practice.

4.3.1 Queue Length Lower Bound with Reconfiguration Delay

The first proposition extends the analysis from [29, Proposition 1], which gives a universal lower bound on the expected queue length for switch systems without reconfiguration delay. The proof of the proposition couples the queue length process $\mathbf{Q}(t)$ to that of a queueing system with less restricted schedule constraint, and is given in appendix 4.6.2.

Proposition 4.2. *Consider a switch system with the arrival process $\mathbf{a}(t)$, which has the mean $\boldsymbol{\lambda} = (1-\epsilon)\boldsymbol{\nu}$ for some $\boldsymbol{\nu} \in \mathcal{F}$ and variance $\boldsymbol{\sigma}^2$. Let $\mathbf{Q}(t)$ denote the queue lengths process of the switch system. Suppose the switch system is stable under its scheduling policy, where the queue lengths process $\mathbf{Q}(t)$ converges in distribution to a steady state random vector $\bar{\mathbf{Q}}$. The expected sum of queue lengths is lower bounded by*

$$\mathbb{E}\Big[\sum_{ij} \bar{q}_{ij}\Big] \geq \frac{\|\boldsymbol{\sigma}\|^2}{2(\epsilon-p)} - \frac{n(1-\epsilon)(\epsilon-2p)}{2(\epsilon-p)} \tag{4.24}$$

where $p = \mathbb{E}[\mathbb{1}_{\{r(t)>0\}}]$ is the probability that the switch system is in reconfiguration under the *given scheduling policy.*

Note that the lower bound in Proposition 4.2 coincides with the lower bound in [29] when $p = 0$, and monotonically increases as p increases. This result is not surprising since the probability of reconfiguration p represents the portion of overhead caused by reconfiguration delay, and should degrade the performance when p increases. However, the minimal of the lower bound occurring at $p = 0$ contradicts the intuition that not switching the schedule also hurts the performance. In fact, for $p = 0$, the switch is always stuck at one schedule and any queues that are not served by the schedule would increase without a bound. In other words, the lower bound in Proposition 4.2 does not capture the effect of infrequent schedule reconfiguration.

To capture the effect of infrequent schedule reconfiguration, the following proposition lower bounds the expected queue lengths by examining the unserved queues when the switch is fixed at one schedule between two reconfiguration times. The proof of Proposition 4.3 is given in appendix 4.6.3.

Proposition 4.3. *Given a switch system with the arrival process* $\mathbf{a}(t)$*, which has the mean* $\boldsymbol{\lambda} = (1-\epsilon)\boldsymbol{\nu}$ *for some* $\boldsymbol{\nu} \in \mathcal{F}$ *and variance* $\boldsymbol{\sigma}^2$*. For any scheduling policy under which the switch system is stable, and the queue lengths process* $\mathbf{Q}(t)$ *converges in distribution to a steady state random vector* $\bar{\mathbf{Q}}$*, the average sum of queue lengths is lower bounded by*

$$\mathbb{E}\bigg[\sum_{ij} \bar{q}_{ij}\bigg] \geq \frac{\Delta_r}{2p}(1-\epsilon)(n-\bar{\alpha}) \tag{4.25}$$

where $\bar{\alpha} = \max_{\mathbf{S}\in\mathcal{S}} \big\langle \boldsymbol{\nu}, \mathbf{S} \big\rangle$*, and* $p = \mathbb{E}[\mathbb{1}_{\{r(t)>0\}}]$ *is the probability that the switch system is in reconfiguration under the given scheduling policy.*

With the two propositions above, we may then derive an optimal lower bound for a given switch system. In particular, for each reconfiguration probability p, the lower bound is given by the maximum of eqs. (4.24) and (4.25). Due to the monotonicity with p of eqs. (4.24) and (4.25), the reconfiguration probability p that minimizes the joint lower bound could be easily solved by equating eqs. (4.24) and (4.25).

In this paper, we are particularly interested in the queue length scaling in the heavy traffic regime, where ϵ approaches 0. The following corollary provides a queue length lower bound in the heavy traffic regime.

Corollary 4.1. *Given a sequence of switch systems with a fixed reconfiguration delay $\Delta_r > 0$, parametrized by $0 < \epsilon < 1$. For each switch system, let $p^{(\epsilon)}$ be the reconfiguration probability that minimizes the joint lower bounds of eqs. (4.24) and (4.25), then in the heavy traffic regime, the reconfiguration probability satisfies:*

$$\lim_{\epsilon \downarrow 0} \frac{p^{(\epsilon)}}{\epsilon} = \frac{\Delta_r(n-\bar{\alpha})}{\|\boldsymbol{\sigma}\|^2 + \Delta_r(n-\bar{\alpha})} \tag{4.26}$$

where $\bar{\alpha} = \max_{\mathbf{S}\in\mathcal{S}} \big\langle \boldsymbol{\nu}, \mathbf{S} \big\rangle$.

Therefore, we have the following queue length lower bound in the heavy traffic limit:

$$\liminf_{\epsilon \downarrow 0} \epsilon \mathbb{E}\Bigg[\sum_{ij} \bar{q}_{ij}\Bigg] \geq \frac{\|\boldsymbol{\sigma}\|^2 + \Delta_r(n-\bar{\alpha})}{2} \tag{4.27}$$

Proof. Since eq. (4.24) is monotonically increasing in p and eq. (4.25) is monotonically decreasing in p, the minimizer $p^{(\epsilon)}$ could be solved by equating (4.24) and (4.25):

$$p^{(\epsilon)2} + \bigg(\frac{\|\boldsymbol{\sigma}\|^2}{2n(1-\epsilon)} - \frac{\epsilon}{2} + \frac{\Delta_r(n-\bar{\alpha})}{2n}\bigg)p^{(\epsilon)} - \frac{\epsilon\Delta_r(n-\bar{\alpha})}{2n} = 0$$

Since $p^{(\epsilon)} \geq 0$, the only feasible solution for $p^{(\epsilon)}$ is given by

$$p^{(\epsilon)} = \frac{\sqrt{C^2 + x} - C}{2} = \frac{C}{2}\left(\sqrt{1 + \frac{x}{C^2}} - 1\right)$$

where $C = \frac{\|\boldsymbol{\sigma}\|^2}{2n(1-\epsilon)} - \frac{\epsilon}{2} + \frac{\Delta_r(n-\bar{\alpha})}{2n}$ and $x = 2\epsilon\Delta_r\frac{n-\bar{\alpha}}{n}$. Note that when $\frac{x}{C^2} << 1$, we have $p^{(\epsilon)} \approx \frac{x}{4C}$. Also, since $\frac{x}{C^2} \to 0$ as $\epsilon \downarrow 0$, we thus obtain

$$\lim_{\epsilon\downarrow 0} \frac{p^{(\epsilon)}}{\epsilon} = \lim_{\epsilon\downarrow 0} \frac{x}{4C} = \frac{\Delta_r(n-\bar{\alpha})}{\|\boldsymbol{\sigma}\|^2 + \Delta_r(n-\bar{\alpha})}$$

□

Corollary 4.1 generalizes the lower bound from [29, Proposition 1] and characterizes the effect of the reconfiguration delay Δ_r on the delay performance. Note that $\bar{\alpha}$ in corollary 4.1 is different from α defined in eq. (4.14), and by definition $\alpha \leq \bar{\alpha} \leq n$. Compare the queue length bound of the Adaptive MaxWeight in eq. (4.23) with corollary 4.1, and we may see that when the hysteresis function $g(\cdot)$ approaches a linear function, the queue length behavior of the Adaptive MaxWeight approximates the optimal scaling with respect to ϵ as well as to the reconfiguration delay Δ_r in the heavy traffic limit $\epsilon \to 0$.

4.3.2 Queue Length Behavior of Fixed Frame MaxWeight

In this subsection, we analyze the queue length behavior of the Fixed Frame MaxWeight (FFMW) as a benchmark policy and compare it with that of the Adaptive MaxWeight policy.

The FFMW policy is a simple extension of the MaxWeight policy, which sets a fixed parameter T, and periodically reconfigures to the MaxWeight schedule every T time slots. It is shown in [25] that given the traffic load $\rho = 1 - \epsilon$, then the switch system is stabilized by the FFMW policy with any period $T > \frac{\Delta_r}{\epsilon}$. Note that the FFMW policy requires the knowledge of the traffic load, which limits the applicability of the policy in practice.

The following proposition extends the heavy traffic queue length analysis of the MaxWeight policy in [29] and gives an upper bound on the expected sum of queue lengths for switches with reconfiguration delay.

Proposition 4.4. *Given a switch system with a fixed reconfiguration delay* $\Delta_r > 0$*, and the arrival process* $\mathbf{a}(t)$ *as described in Section 4.1. Suppose the mean arrival rate vector is given by* $\boldsymbol{\lambda} = (1-\epsilon)\boldsymbol{\nu}$ *where* $\boldsymbol{\nu} \in \mathcal{F}$ *such that* $\nu_{\min} \triangleq \min\limits_{ij} \nu_{ij} > 0$*, and for some* $\epsilon > 0$*. The variance of* $\mathbf{a}(t)$ *is* $\boldsymbol{\sigma}^2$*. For any* ϵ *that satisfies* $\epsilon < \frac{\nu_{\min}}{4n}$*, suppose that the switch system is operated under the Fixed Frame MaxWeight policy with schedule duration* $T > \frac{\Delta_r}{\epsilon}$*, then the expected queue length sum satisfies:*

$$\mathbb{E}\Big[\sum_{ij} \bar{q}_{ij}\Big] \leq (1 - \frac{1}{2n})\frac{T}{\epsilon T - \Delta_r}\|\boldsymbol{\sigma}\|^2 + T\Big(\frac{n(1+\epsilon)}{2} + n^2(a_{\max} + 2M)\Big) \tag{4.28}$$

where $M = \frac{4n(a_{\max}+1)+4(\|\boldsymbol{\lambda}\|^2+\|\boldsymbol{\sigma}\|^2+n)+16\sqrt{2}n^2a_{\max}^2}{\nu_{\min}} + 2n(2\sqrt{2}a_{\max}+1)$.

We may then further minimize the upper bound over T*, and derive the minimizing schedule duration as* $T^* = \frac{\Delta_r}{\epsilon}(1 + \sqrt{\frac{(1-\frac{1}{2n})\|\boldsymbol{\sigma}\|^2}{\Delta_r M'}})$ *and the corresponding heavy traffic queue length upper bound is given by:*

$$\limsup_{\epsilon \downarrow 0} \epsilon \mathbb{E}\Big[\sum_{ij} \bar{q}_{ij}\Big] \leq \Big(\sqrt{(1-\frac{1}{2n})\|\boldsymbol{\sigma}\|^2} + \sqrt{\Delta_r M'}\Big)^2 \tag{4.29}$$

where $M' = \frac{n}{2} + n^2(a_{\max} + 2M)$.

From proposition 4.4, we could see that given the traffic load information, the FFMW policy may achieve the optimal scaling with respect to ϵ and Δ_r in the heavy traffic limit $\epsilon \to 0$. Note that since eq. (4.28) is not necessarily a tight bound, hence the derived minimizing schedule duration T^* may not be the true minimizer of the expected queue length. However, it does guarantee the optimal scaling, and may be a good estimate for the true minimizer. From the expression of T^* we could see that the minimizer is close to the boundary of stability $T = \frac{\Delta_r}{\epsilon}$, this also implies that the optimal delay scaling of the FFMW policy is rather intolerant to estimation error of the traffic load. Moreover, this issue becomes more prominent for large Δ_r, since it decreases the distance between T^* and the boundary of stability.

In the next section, we compare the average queue length performance between the Adaptive MaxWeight policy and the FFMW policy. We show that with the hysteresis function $g(\cdot)$ that is close to a linear function, the Adaptive MaxWeight policy has a comparable performance to the FFMW policy, and does not require any knowledge of traffic load.

4.4 Simulations

In this section, we show simulation results for switches with reconfiguration delay operated under the Adaptive MaxWeight policy, with hysteresis function $g(x) = (1-\gamma)x^{1-\delta}$. We first compare the simulation result with the Fixed Frame MaxWeight policy, and then determine the scaling of the average queue length with respect to different system parameters and compare with the queue length scaling derived in section 4.2.

We now briefly describe the simulation setup. The arrival processes are assumed to be Poisson processes, all with the same arrival rate, which is also known as the uniform traffic. More specifically, the matrix $\boldsymbol{\nu} \in \mathcal{F}$ satisfies $\nu_{ij} = \frac{1}{n}, \forall i, j \in \{1, \ldots, n\}$. Under the uniform traffic, we have $\|\boldsymbol{\nu}\|^2 = 1$, and thus $\alpha = n - 1$ in eq. (4.21). For the parameter of the hysteresis function g, since we are only interested in the scaling, we fix $\gamma = 0.1$, and consider $\delta \in \{0.01, 0.05, 0.1, 0.2\}$ for average queue length comparison.

Fig. 4.1 shows the average queue length under various traffic loads $\rho \in [0.1, 1]$. From Fig. 4.1 we could see that the average queue length is with smaller δ, in other words, the delay performance improves when the hysteresis function $g(x)$ approaches a linear function. The result implies that while the analysis in this work focuses on the heavy traffic regime, the conclusion that δ close to zero gives better delay performance also applies for lower traffic loads.

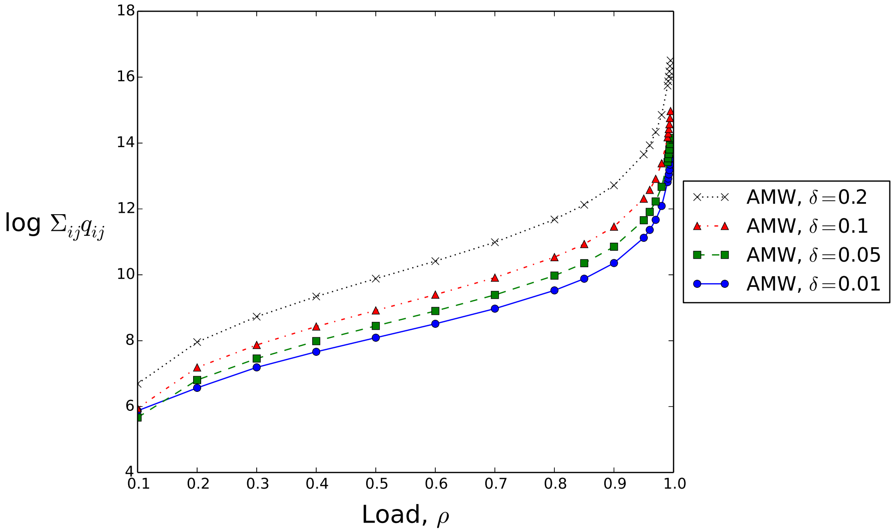

Figure 4.1: Mean total queue length versus traffic load ρ under uniform traffic. Number of ports is $n = 16$, and reconfiguration delay $\Delta_r = 20$.

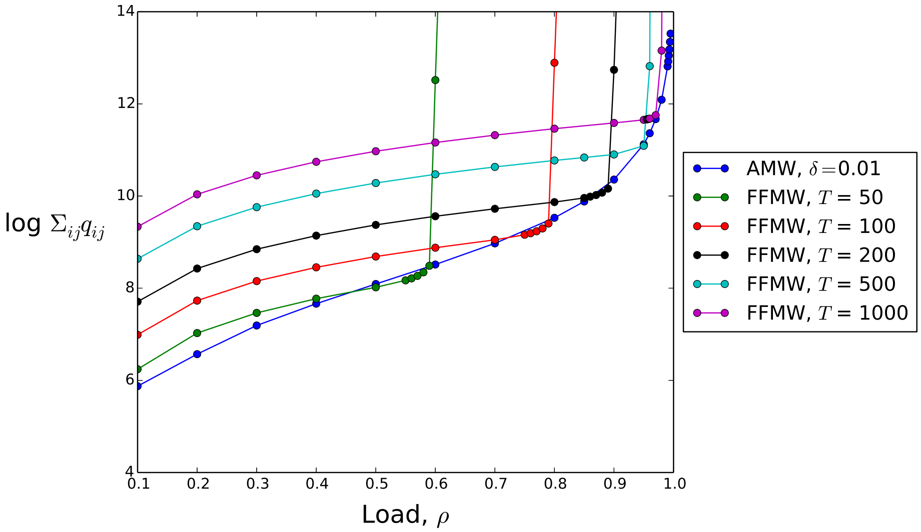

Figure 4.2: Mean total queue length versus traffic load ρ under uniform traffic. Number of ports is $n = 16$, and reconfiguration delay $\Delta_r = 20$.

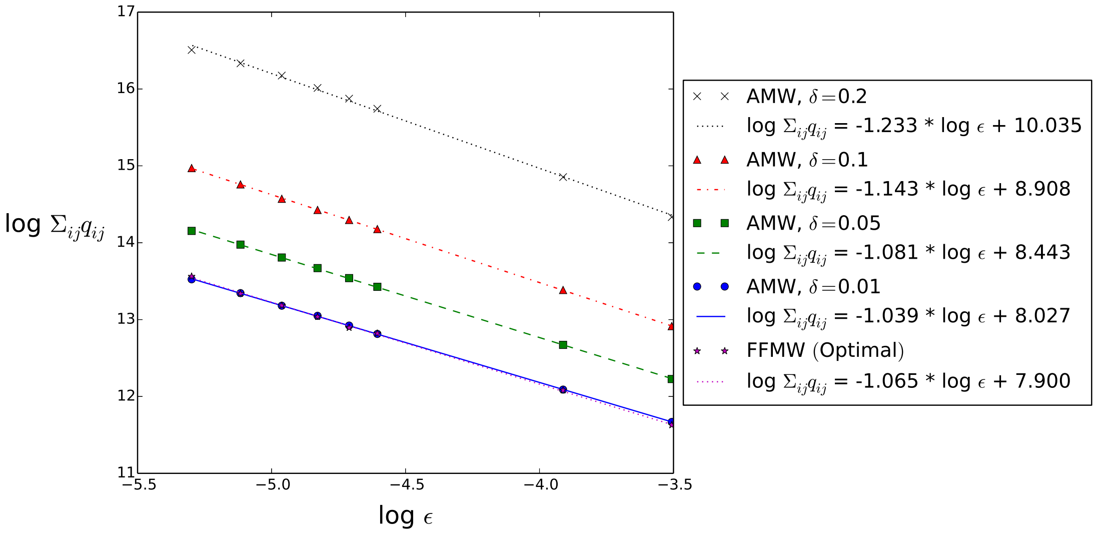

Figure 4.3: Mean total queue length versus traffic load near the capacity region. Number of ports is $n = 16$, and reconfiguration delay $\Delta_r = 20$.

In Fig. 4.2, we keep the Adaptive MaxWeight with $\delta = 0.01$, which has the optimal performance, and compare the average queue length to the FFMW policy with various schedule durations $T \in \{50, 100, 200, 500, 1000\}$. We first note that for each schedule duration T, the average queue length grows quickly when ρ is approaching $1 - \frac{\Delta_r}{T}$.

We now focus on simulations in the heavy traffic regime. In Fig. 4.3, we plot the average total queue length of the Adaptive MaxWeight for various $\epsilon \in [0.005, 0.03]$ (corresponding to $\rho \in [0.97, 0.995]$), and take log scale for both axes. For the FFMW policy, we consider the optimal queue length performance over schedule durations. In other words, for each ϵ, we consider the FFMW policy with different schedule durations and take the one that minimizes the average queue length for comparison. We may see that the Adaptive MaxWeight with $\delta = 0.01$ closely follows the optimal performance of the FFMW policy. We then use linear regression to determine the scaling (*i.e.* the exponent) of the average queue length with respect to ϵ in the heavy traffic regime. With the scaling result from (4.21), the scaling with respect to ϵ is close to $g^{-1}(1/\epsilon)$, hence the theoretical exponent should be $-1/(1-\delta)$, which would be $\{-1.010, -1.053, -1.111, -1.250\}$ for $\delta = \{0.01, 0.05, 0.1, 0.2\}$, respectively.

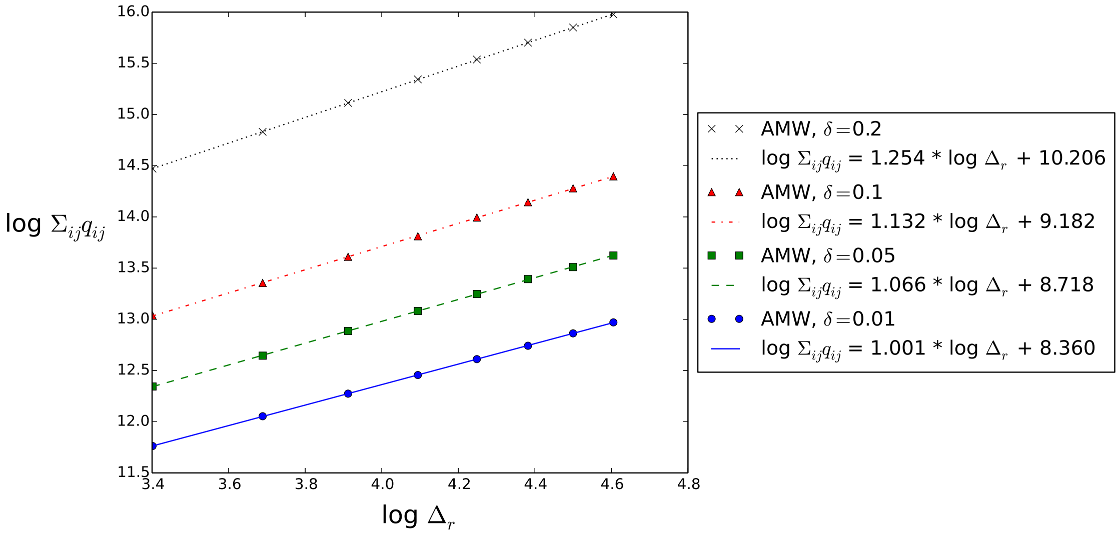

Figure 4.4: Mean total queue length versus reconfiguration delay Δ_r. Number of ports is $n = 16$, and traffic load is $\rho = 0.96$.

Figs. 4.4 and 4.5 show the queue length scaling behavior under varying reconfiguration delay Δ_r and varying number of ports n, while the traffic load is fixed as $\rho = 0.96$ (or $\epsilon = 0.04$). For the reconfiguration delay, the scaling is $g^{-1}(\Delta_r)$, hence the theoretical exponents are $\{1.010, 1.053, 1.111, 1.250\}$ for $\delta \in \{0.01, 0.05, 0.1, 0.2\}$, respectively. We may see from Fig. 4.4 that the exponents obtained from the simulation result are close to our derived scaling. On the other hand, the scaling with respect to n is $ng^{-1}(n)$, hence the theoretical exponents should be $\{2.010, 2.053, 2.111, 2.250\}$. We could see that the exponents derived from the simulation result are slightly larger than the our derived scaling.

4.5 Concluding Remarks

We consider the heavy traffic queue length behavior in an input-queued switch with reconfiguration delay, operating under the Adaptive MaxWeight policy. It is shown that the Adaptive MaxWeight exhibits weak state space collapse behavior, which could be considered as an inheritance from the MaxWeight policy in the regime of zero reconfiguration delay. Utilizing

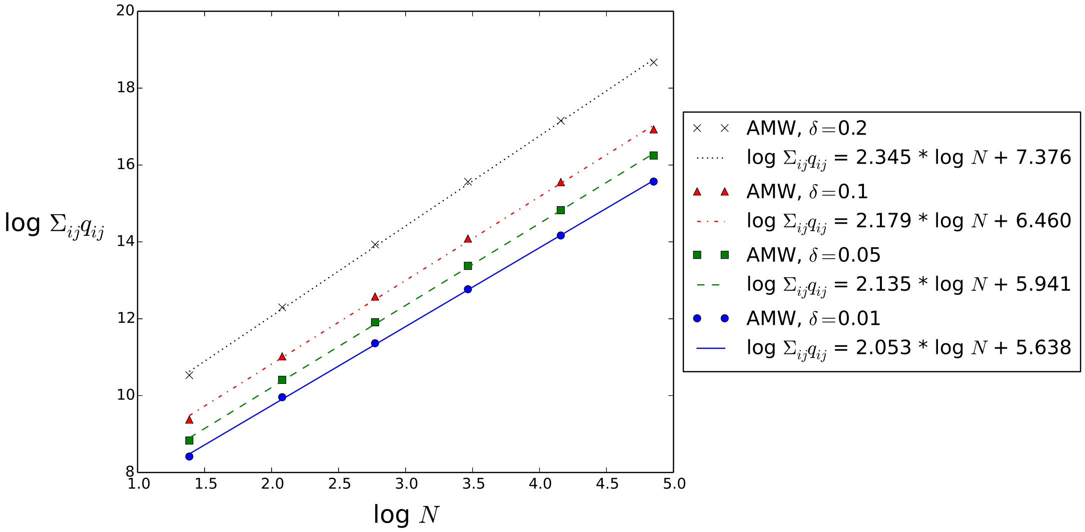

Figure 4.5: Mean total queue length versus number of ports n. Reconfiguration delay is $\Delta_r = 20$, and traffic load is $\rho = 0.96$.

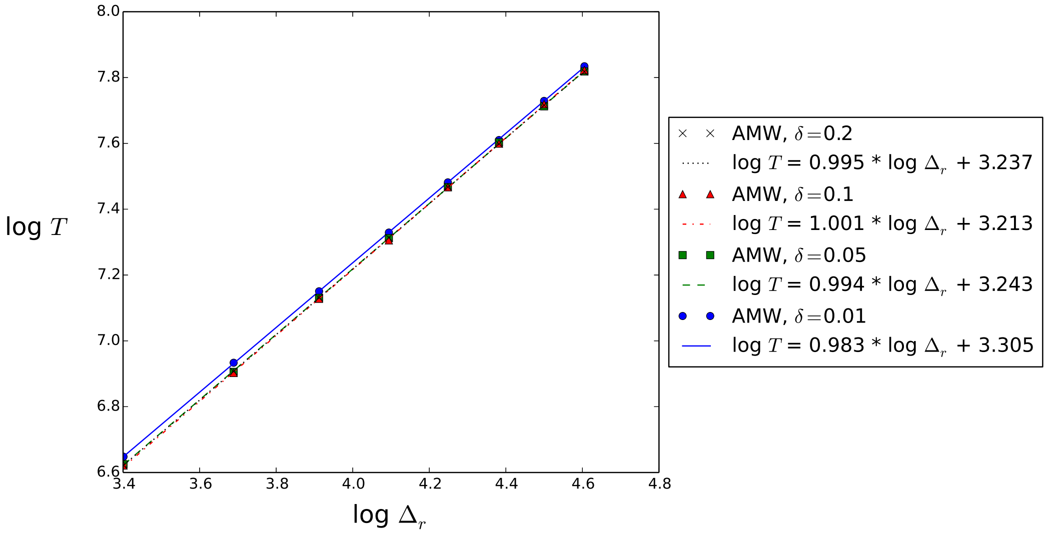

Figure 4.6: Mean schedule duration versus Mean total queue length. Number of ports is $n = 16$, and reconfiguration delay $\Delta_r = 20$.

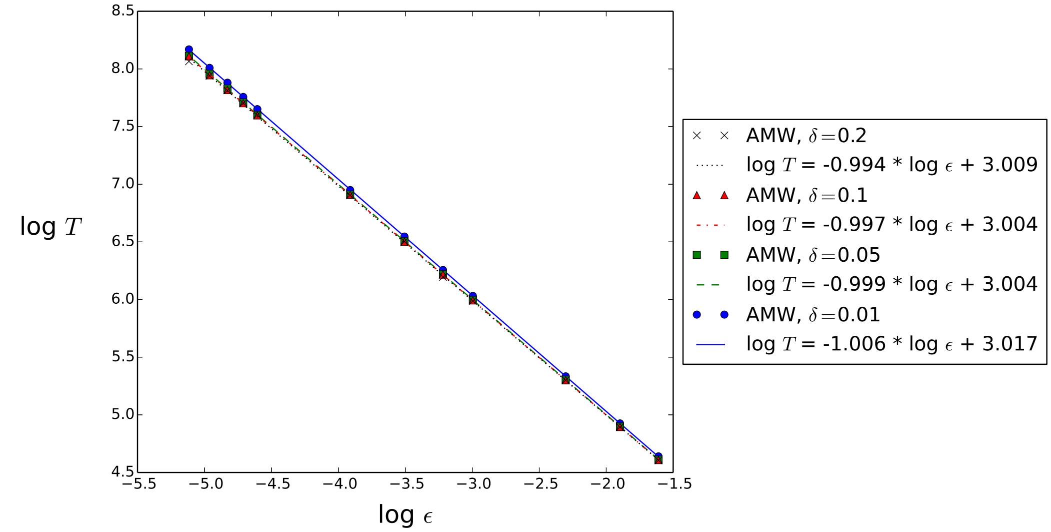

Figure 4.7: Mean schedule duration versus Mean total queue length. Number of ports is $n = 16$, and reconfiguration delay $\Delta_r = 20$.

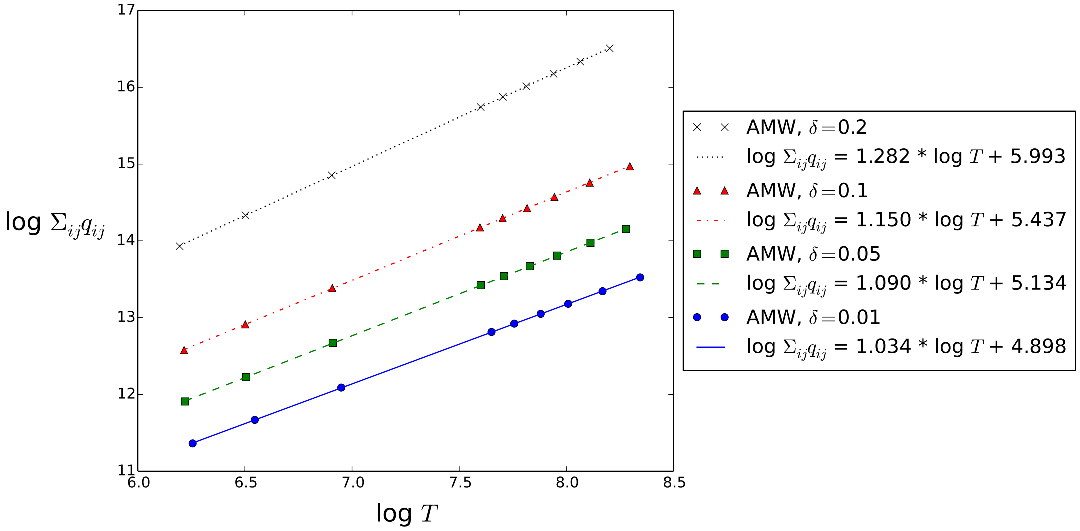

Figure 4.8: Mean schedule duration versus Mean total queue length. Number of ports is $n = 16$, and reconfiguration delay $\Delta_r = 20$.

the Lyapunov drift technique introduced in [29], we obtain a queue length upper bound in heavy traffic, which depends on the expected schedule duration. We then discover a relation between the expected schedule duration and the expected queue length, which then implies asymptotically tight bounds for the expected schedule duration in heavy traffic limit, thus determining its scaling. The scaling of the expected schedule duration then implies the dependence of the queue length scaling with the selection of the hysteresis function g, and that this scaling improves as g becomes closer to linear. Simulation results are also presented to illustrate the queue length scaling with respect to several system parameters (e.g. traffic load, number of ports, reconfiguration delay) and for comparison to the derived queue length scaling in heavy traffic.

The results obtained in this paper apply to traffic patterns that all input and output ports are saturated. It would be interested to consider the queue length behavior of the Adaptive MaxWeight under incompletely saturated traffic, for example, traffic conditions considered in [28]. The state space collapse result might be similar due to the inheritance from MaxWeight policy, but the characterization for the expected schedule duration remains unclear at this point.

Chapter 4, in part, is a reprint of the material as it appears in the paper: C.-H. Wang, S. T. Maguluri, T. Javidi, “Heavy Traffic Queue Length Behavior in Switches with Reconfiguration Delay”, IEEE Conference on Computer Communications (INFOCOM), pp 1-9, 2017. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in full, is currently being prepared for submission for publication as: C.-H. Wang, S. T. Maguluri, T. Javidi, ”Toward Optimal Heavy Traffic Queue Length Behavior for Switches with Reconfiguration Delay.” The dissertation author was the primary investigator and author of this material.

4.6 Supplementary Proofs

4.6.1 Proof of Proposition 4.1

Proof. For ease of notation, we drop the superscript (ϵ) in the following derivation. For each state $\mathbf{X} = (\mathbf{Q}, \mathrm{s}, r)$, we define the Lyapunov function $Z(\mathbf{X}) = \max\{\|\mathbf{Q}_\perp\| - \theta\|\mathbf{Q}_{\|}\|, 0\}$. We then apply Lemma 4.1 with the Lyapunov function Z to obtain the result. Note that the selection of the Lyapunov function is such that Z is a nonnegative function. Since $\|\mathbf{Q}_\perp\| - \theta\|\mathbf{Q}_{\|}\| \leq Z(\mathbf{X})$ for any state $\mathbf{X} = (\mathbf{Q}, \mathrm{s}, r)$, the result follows a bound on $\mathbb{E}[Z(\bar{\mathbf{X}})]$.

We first verify Condition C.2 for $Z(\mathbf{X})$:

$$\begin{aligned}
|\Delta^T Z(\mathbf{X})| =& \Big|\Big(\|\mathbf{Q}_\perp(t+T)\| - \theta\|\mathbf{Q}_{\|}(t+T)\|\Big) - \Big(\|\mathbf{Q}_\perp(t)\| - \theta\|\mathbf{Q}_{\|}(t)\|\Big)\Big| \\
\leq& \Big|\|\mathbf{Q}_\perp(t+T)\| - \|\mathbf{Q}_\perp(t)\|\Big| + \theta\Big|\|\mathbf{Q}_{\|}(t+T)\| - \|\mathbf{Q}_{\|}(t)\|\Big| \\
\leq& \|\mathbf{Q}_\perp(t+T) - \mathbf{Q}_\perp(t)\| + \theta\|\mathbf{Q}_{\|}(t+T) - \mathbf{Q}_{\|}(t)\| \\
\leq& (1+\theta)\,\|\mathbf{Q}(t+T) - \mathbf{Q}(t)\| \\
\leq& (1+\theta) n a_{\max} T \triangleq D
\end{aligned} \tag{4.30}$$

Here we use the fact that $\mathbf{Q}_\perp$ is a projection onto $\mathcal{K}^\circ = \{\mathbf{x} \in \mathbb{R}^{n^2} : \langle \mathbf{x}, \mathbf{y}\rangle \leq 0, \forall \mathbf{y} \in \mathcal{K}\}$, the polar cone of $\mathcal{K}$. Since the projection onto a cone is non-expansive, we have $\|\mathbf{x}_\perp - \mathbf{y}_\perp\| \leq \|\mathbf{x} - \mathbf{y}\|$ and $\|\mathbf{x}_{\|} - \mathbf{y}_{\|}\| \leq \|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y}$.

To verify condition C.1, we need to bound the T-step drift for $Z(\mathbf{X})$. For ease of notation, we denote $\mathbb{E}[\,\cdot\,|\mathbf{X}(t) = \mathbf{X}]$ as $\mathbb{E}_{\mathbf{X}}[\,\cdot\,]$. From (4.30), it is not hard to see that $\forall \mathbf{X} : Z(\mathbf{X}) > D$,

$$\mathbb{E}_{\mathbf{X}}\Big[\Delta^T Z(\mathbf{X})\Big] = \mathbb{E}_{\mathbf{X}}\Big[\Big(\|\mathbf{Q}_\perp(t+T)\| - \|\mathbf{Q}_\perp(t)\|\Big) - \theta\Big(\|\mathbf{Q}_{\|}(t+T)\| - \|\mathbf{Q}_{\|}(t)\|\Big)\Big]. \tag{4.31}$$

Therefore, we need only to consider the T-step expected drift of $\|\mathbf{Q}_\perp\|$ and $\|\mathbf{Q}_{\|}\|$.

We first consider the drift of $\|\mathbf{Q}_\perp\|$. The derivation follows the line in [29] where the relation in [29, Lemma 4] is used: Let $V(\mathbf{X}) = ||\mathbf{Q}||^2$, $V_{\|}(\mathbf{X}) = ||\mathbf{Q}_{\|}||^2$, and ΔV, $\Delta V_{\|}$ denote the one-step drift of V, $V_{\|}$ (respectively), then $\|\mathbf{Q}_{\perp}(t+1)\| - \|\mathbf{Q}_{\perp}(t)\| \leq \frac{1}{2\|\mathbf{Q}_{\perp}(t)\|}\big(\Delta V(\mathbf{X}(t)) - \Delta V_{\|}(\mathbf{X}(t))\big)$. With the relation, we have

$$\begin{aligned}
\mathbb{E}_{\mathbf{X}}\Big[\|\mathbf{Q}_{\perp}(t+T)\| - \|\mathbf{Q}_{\perp}(t)\|\Big] =&\mathbb{E}_{\mathbf{X}}\Big[\sum_{\tau=t}^{t+T-1}\big(\|\mathbf{Q}_{\perp}(\tau+1)\| - \|\mathbf{Q}_{\perp}(\tau)\|\big)\Big] \\
\leq&\mathbb{E}_{\mathbf{X}}\Big[\sum_{\tau=t}^{t+T-1}\frac{\Delta V(\mathbf{X}(\tau)) - \Delta V_{\|}(\mathbf{X}(\tau))}{2\|\mathbf{Q}_{\perp}(\tau)\|}\Big] \\
=&\mathbb{E}_{\mathbf{X}}\Big[\sum_{\tau=t}^{t+T-1}\mathbb{E}\Big[\frac{\Delta V(\mathbf{X}(\tau)) - \Delta V_{\|}(\mathbf{X}(\tau))}{2\|\mathbf{Q}_{\perp}(\tau)\|}\Big|\mathbf{X}(\tau)\Big]\Big] \qquad (4.32)
\end{aligned}$$

We now derive bounds for ΔV and $\Delta V_{\|}$:

$$\begin{aligned}
&\mathbb{E}\Big[\Delta V(\mathbf{X}(\tau))\Big|\mathbf{X}(\tau)\Big] \\
=&\mathbb{E}\Big[\|\mathbf{Q}(\tau+1)\|^2 - \|\mathbf{Q}(\tau)\|^2\Big|\mathbf{X}(\tau)\Big] \\
=&\mathbb{E}\Big[\|\mathbf{Q}(\tau) + \mathbf{a}(\tau) - \mathbf{s}(\tau)\mathbb{1}_{\{r(\tau)=0\}}\|^2 + \|\mathbf{u}(\tau)\|^2 \\
&+ 2\Big\langle\mathbf{Q}(\tau+1) - \mathbf{u}(\tau), \mathbf{u}(\tau)\Big\rangle - \|\mathbf{Q}(\tau)\|^2\Big|\mathbf{X}(\tau)\Big] \\
\leq&\mathbb{E}\Big[\|\mathbf{Q}(\tau) + \mathbf{a}(\tau) - \mathbf{s}(\tau)\mathbb{1}_{\{r(\tau)=0\}}\|^2 - \|\mathbf{Q}(\tau)\|^2\Big|\mathbf{X}(\tau)\Big] \\
=&\sum_{i,j}\mathbb{E}\Big[a_{ij}^2(\tau) + s_{ij}(\tau)\mathbb{1}_{\{r(\tau)=0\}} - 2a_{ij}(\tau)s_{ij}(\tau)\mathbb{1}_{\{r(\tau)=0\}}\Big|\mathbf{X}(\tau)\Big] \\
&+ \mathbb{E}\Big[2\Big\langle\mathbf{Q}(\tau), \boldsymbol{\lambda} - \mathbf{s}(\tau)\mathbb{1}_{\{r(\tau)=0\}}\Big\rangle\Big|\mathbf{X}(\tau)\Big] \\
\overset{(a)}{\leq}&\sum_{ij}(\lambda_{ij}^2 + \sigma_{ij}^2) + n + 2\Big\langle\mathbf{Q}(\tau), (1-\epsilon)\boldsymbol{\nu} - \mathbf{s}(\tau)\Big\rangle + 2\Big\langle\mathbf{Q}(\tau), \mathbf{s}(\tau)\Big\rangle\mathbb{1}_{\{r(\tau)>0\}} \\
=&\|\boldsymbol{\lambda}\|^2 + \|\boldsymbol{\sigma}\|^2 + n - 2\epsilon\Big\langle\mathbf{Q}(\tau), \boldsymbol{\nu}\Big\rangle + 2\Big\langle\mathbf{Q}(\tau), \boldsymbol{\nu} - \mathbf{s}(\tau)\Big\rangle + 2\Big\langle\mathbf{Q}(\tau), \mathbf{s}(\tau)\Big\rangle\mathbb{1}_{\{r(\tau)>0\}} \qquad (4.33)
\end{aligned}$$

where (a) follows from $\mathbb{E}[a_{ij}^2] = \lambda_{ij}^2 + \sigma_{ij}^2$, $a_{ij}(t)s_{ij}(t) \geq 0$ for all i, j, and $\sum_{ij} s_{ij}(t) = 1$ for all t.

Suppose g is the sublinear hysteresis function for the Adaptive MaxWeight, then by the sublinearity, there exists a constant K_θ such that $g(x) < \frac{\theta}{\alpha}x$ for any $x > K_\theta$, where $\alpha = \frac{8\|\boldsymbol{\nu}\|}{\nu_{\min}}$. Hence by the definition of the Adaptive MaxWeight, we have for any $\mathbf{X}(\tau)$ such that $\langle \mathbf{Q}(\tau), \mathbf{s}^*(\tau)\rangle > K_\theta$:

$$\begin{aligned}
\Big\langle \mathbf{Q}(\tau), \boldsymbol{\nu} - \mathbf{s}(\tau)\Big\rangle =& \Big\langle \mathbf{Q}(\tau), \boldsymbol{\nu} - \mathbf{s}^*(\tau)\Big\rangle + \Big\langle \mathbf{Q}(\tau), \mathbf{s}^*(\tau) - \mathbf{s}(\tau)\Big\rangle \\
\leq& \Big\langle \mathbf{Q}(\tau), \boldsymbol{\nu} - \mathbf{s}^*(\tau)\Big\rangle + g\left(\Big\langle \mathbf{Q}(\tau), \mathbf{s}^*(\tau)\Big\rangle\right) \\
\leq& \Big\langle \mathbf{Q}(\tau), \boldsymbol{\nu} - \mathbf{s}^*(\tau)\Big\rangle + \frac{\theta}{\alpha}\Big\langle \mathbf{Q}(\tau), \mathbf{s}^*(\tau)\Big\rangle \\
=& \Big(1 - \frac{\theta}{\alpha}\Big)\Big\langle \mathbf{Q}(\tau), \boldsymbol{\nu} - \mathbf{s}^*(\tau)\Big\rangle + \frac{\theta}{\alpha}\Big\langle \mathbf{Q}(\tau), \boldsymbol{\nu}\Big\rangle
\end{aligned}$$

From [29, Claim 2], we have $\Big\langle \mathbf{Q}(\tau), \boldsymbol{\nu} - \mathbf{s}^*(\tau)\Big\rangle \leq -\nu_{\min}\|\mathbf{Q}_\perp(\tau)\|$. Therefore,

$$\begin{aligned}
\mathbb{E}\Big[\Delta V(\mathbf{X}(\tau))\Big|\mathbf{X}(\tau)\Big] \leq& \|\boldsymbol{\lambda}\|^2 + \|\boldsymbol{\sigma}\|^2 + n - 2\epsilon\Big\langle \mathbf{Q}(\tau), \boldsymbol{\nu}\Big\rangle - 2\Big(1 - \frac{\theta}{\alpha}\Big)\nu_{\min}\|\mathbf{Q}_\perp(\tau)\| \\
&+ 2\frac{\theta}{\alpha}\Big\langle \mathbf{Q}(\tau), \boldsymbol{\nu}\Big\rangle + 2\Big\langle \mathbf{Q}(\tau), \mathbf{s}(\tau)\Big\rangle \mathbb{1}_{\{r(\tau)>0\}} \qquad (4.34)
\end{aligned}$$

For $\Delta V_{\|}$, we have

$$\begin{aligned}
&\mathbb{E}\Big[\Delta V_{\|}(\mathbf{X}(\tau))\Big|\mathbf{X}(\tau)\Big] = \mathbb{E}\Big[\|\mathbf{Q}_{\|}(\tau+1)\|^2 - \|\mathbf{Q}_{\|}(\tau)\|^2\Big|\mathbf{X}(\tau)\Big]\\
=&\mathbb{E}\Big[\Big\langle \mathbf{Q}_{\|}(\tau+1)+\mathbf{Q}_{\|}(\tau), \mathbf{Q}_{\|}(\tau+1)-\mathbf{Q}_{\|}(\tau)\Big\rangle\Big|\mathbf{X}(\tau)\Big]\\
=&\mathbb{E}\Big[\|\mathbf{Q}_{\|}(\tau+1)-\mathbf{Q}_{\|}(\tau)\|^2 + 2\Big\langle \mathbf{Q}_{\|}(\tau), \mathbf{Q}_{\|}(\tau+1)-\mathbf{Q}_{\|}(\tau)\Big\rangle\Big|\mathbf{X}(\tau)\Big]\\
\geq&2\mathbb{E}\Big[\Big\langle \mathbf{Q}_{\|}(\tau), \mathbf{Q}_{\|}(\tau+1)-\mathbf{Q}_{\|}(\tau)\Big\rangle\Big|\mathbf{X}(\tau)\Big]\\
=&2\mathbb{E}\Big[\Big\langle \mathbf{Q}_{\|}(\tau), \mathbf{Q}(\tau+1)-\mathbf{Q}(\tau)\Big\rangle - \Big\langle \mathbf{Q}_{\|}(\tau), \mathbf{Q}_{\perp}(\tau+1)-\mathbf{Q}_{\perp}(\tau)\Big\rangle\Big|\mathbf{X}(\tau)\Big]\\
\overset{(b)}{\geq}&2\mathbb{E}\Big[\Big\langle \mathbf{Q}_{\|}(\tau), \mathbf{a}(\tau)-\mathbf{s}(\tau)\mathbb{1}_{\{r(\tau)=0\}}+\mathbf{u}(\tau)\Big\rangle\Big|\mathbf{X}(\tau)\Big]\\
\geq&2\Big\langle \mathbf{Q}_{\|}(\tau), \boldsymbol{\lambda}\Big\rangle - 2\Big\langle \mathbf{Q}_{\|}(\tau), \mathbf{s}(\tau)\mathbb{1}_{\{r(\tau)=0\}}\Big\rangle\\
=&-2\epsilon\Big\langle \mathbf{Q}_{\|}(\tau), \boldsymbol{\nu}\Big\rangle + 2\Big\langle \mathbf{Q}_{\|}(\tau), \boldsymbol{\nu}-\mathbf{s}(\tau)\Big\rangle + 2\Big\langle \mathbf{Q}_{\|}(\tau), \mathbf{s}(\tau)\Big\rangle\mathbb{1}_{\{r(\tau)>0\}}\\
=&-2\epsilon\Big\langle \mathbf{Q}_{\|}(\tau), \boldsymbol{\nu}\Big\rangle + 2\Big\langle \mathbf{Q}_{\|}(\tau), \mathbf{s}(\tau)\Big\rangle\mathbb{1}_{\{r(\tau)>0\}} \qquad (4.35)
\end{aligned}$$

For (b), we use the following properties of the projection onto cone $\mathcal{K}$. For $\mathbf{Q} \in \mathbb{R}^{n^2}$, $\langle \mathbf{Q}_{\|}, \mathbf{Q}_{\perp}\rangle = 0$, and $\mathbf{Q}_{\perp} \in \mathcal{K}^{\circ}$. Therefore, $\langle \mathbf{Q}_{\|}(t), \mathbf{Q}_{\perp}(t)\rangle = 0$, and $\langle \mathbf{Q}_{\|}(t), \mathbf{Q}_{\perp}(t+1)\rangle \leq 0$.

Apply (4.34) and (4.35) into (4.32), we obtain

$$\begin{aligned}
&\mathbb{E}_{\mathbf{X}}\Big[\|\mathbf{Q}_{\perp}(t+T)\| - \|\mathbf{Q}_{\perp}(t)\|\Big]\\
\leq\ &\mathbb{E}_{\mathbf{X}}\Big[\sum_{\tau=t}^{t+T-1}\Big(\frac{\|\boldsymbol{\lambda}\|^2+\|\boldsymbol{\sigma}\|^2+n}{2\|\mathbf{Q}_{\perp}(\tau)\|} - \epsilon\Big\langle\frac{\mathbf{Q}_{\perp}(\tau)}{\|\mathbf{Q}_{\perp}(\tau)\|}, \boldsymbol{\nu}\Big\rangle - \big(1-\frac{\theta}{\alpha}\big)\nu_{\min}\\
&\qquad + \frac{\theta\langle \mathbf{Q}(\tau), \boldsymbol{\nu}\rangle}{\alpha\|\mathbf{Q}_{\perp}(\tau)\|} + \Big\langle\frac{\mathbf{Q}_{\perp}(\tau)}{\|\mathbf{Q}_{\perp}(\tau)\|}, \mathbf{s}(\tau)\Big\rangle\mathbb{1}_{\{r(\tau)>0\}}\Big)\Big]\\
\leq\ &\mathbb{E}_{\mathbf{X}}\Big[T\Big(\frac{\|\boldsymbol{\lambda}\|^2+\|\boldsymbol{\sigma}\|^2+n}{\min\limits_{\tau\in[t,t+T]} 2\|\mathbf{Q}_{\perp}(\tau)\|} + \epsilon\|\boldsymbol{\nu}\| - \big(1-\theta\big)\nu_{\min} + \frac{1+\theta}{\alpha}\|\boldsymbol{\nu}\|\Big)\\
&\qquad + \sqrt{n}\sum_{\tau=t}^{t+T-1}\mathbb{1}_{\{r(\tau)>0\}}\Big] \qquad (4.36)
\end{aligned}$$

where we have used the fact that $\|\mathbf{Q}_{\perp}\| \geq \theta\|\mathbf{Q}_{\|}\|$ implies $\theta\|\mathbf{Q}\| \leq \theta(\|\mathbf{Q}_{\|}\| + \|\mathbf{Q}_{\perp}\|) \leq$

$(1+\theta)\|\mathbf{Q}_\perp\|$, and $\|\mathbf{s}(\tau)\| \le \sqrt{n}$ for any schedule $\mathbf{s}(\tau) \in \mathcal{S}$.

On the other hand, the drift of $\|\mathbf{Q}_\|\|$ could be obtained following (4.35):

$$\begin{aligned}
\mathbb{E}_{\mathbf{X}}\Big[\|\mathbf{Q}_\|(t+T)\| - \|\mathbf{Q}_\|(t)\|\Big] =& \mathbb{E}_{\mathbf{X}}\Big[\sum_{\tau=t}^{t+T-1} \mathbb{E}\Big[\|\mathbf{Q}_\|(\tau+1)\| - \|\mathbf{Q}_\|(\tau)\|\Big|\mathbf{X}(\tau)\Big]\Big] \\
\ge& \mathbb{E}_{\mathbf{X}}\Big[\sum_{\tau=t}^{t+T-1} \mathbb{E}\Big[\frac{\|\mathbf{Q}_\|(\tau+1)\|^2 - \|\mathbf{Q}_\|(\tau)\|^2}{\|\mathbf{Q}_\|(\tau+1)\| + \|\mathbf{Q}_\|(\tau)\|}\Big|\mathbf{X}(\tau)\Big]\Big] \\
\ge& \mathbb{E}_{\mathbf{X}}\Big[\sum_{\tau=t}^{t+T-1} \mathbb{E}\Big[\frac{-2\epsilon\langle\mathbf{Q}_\|(\tau), \boldsymbol{\nu}\rangle}{\|\mathbf{Q}_\|(\tau+1)\| + \|\mathbf{Q}_\|(\tau)\|}\Big|\mathbf{X}(\tau)\Big]\Big] \\
\ge& \mathbb{E}_{\mathbf{X}}\Big[\sum_{\tau=t}^{t+T-1} \frac{-2\epsilon\langle\mathbf{Q}_\|(\tau), \boldsymbol{\nu}\rangle}{\|\mathbf{Q}_\|(\tau)\|}\Big] \ge -2T\epsilon\|\boldsymbol{\nu}\| \qquad (4.37)
\end{aligned}$$

where the last inequality follows the Cauchy-Schwartz inequality.

Now apply (4.36) and (4.37) into (4.31), we obtain

$$\begin{aligned}
\mathbb{E}_{\mathbf{X}}\Big[\Delta^T Z(\mathbf{X})\Big] \le \mathbb{E}_{\mathbf{X}}\Big[&T\Big(\frac{\|\boldsymbol{\lambda}\|^2 + \|\boldsymbol{\sigma}\|^2 + n}{\min\limits_{\tau\in[t,t+T]} 2\|\mathbf{Q}_\perp(\tau)\|} + (1+2\theta)\epsilon\|\boldsymbol{\nu}\| - \big(1-\theta\big)\nu_{\min} + \frac{1+\theta}{\alpha}\|\boldsymbol{\nu}\|\Big) \\
&+ \sqrt{n}\sum_{\tau=t}^{t+T-1} \mathbb{1}_{\{r(\tau)>0\}}\Big] \qquad (4.38)
\end{aligned}$$

From [49, Lemma 1], we know that for any fixed $T > 0$, if $W^*(t) > g^{-1}\Big(nT(a_{\max}+1)\Big) + nT$, then at most one reconfiguration could occur within $[t, t+T]$, which gives $\sum_{\tau=t}^{t+T-1} \mathbb{1}_{r(\tau)>0} \le \Delta_r$.

Select $T = \frac{8\sqrt{n}\Delta_r}{\nu_{\min}}$, then set $D = \frac{3}{2}na_{\max}T = \frac{12n^{3/2}a_{\max}\Delta_r}{\nu_{\min}}$. Then $\forall\mathbf{X} : Z(\mathbf{X}) > \kappa = \Big\{D, na_{\max}T + \frac{4(\|\boldsymbol{\lambda}\|^2+\|\boldsymbol{\sigma}\|^2+n)}{\nu_{\min}}, nK_\theta, ng^{-1}\Big(nT(a_{\max}+1)\Big) + n^2T\Big\}$, and $\forall\epsilon : 0 < \epsilon \le \frac{\nu_{\min}}{16\|\boldsymbol{\nu}\|}$, we have $\mathbb{E}_{\mathbf{X}}\Big[\Delta^T Z(\mathbf{X})\Big] \le -\frac{(1-\theta)\nu_{\min}}{4} \le -\frac{\nu_{\min}}{8}$.

Hence by Lemma 4.1, we have $\forall\epsilon : 0 < \epsilon \le \frac{\nu_{\min}}{16\|\boldsymbol{\nu}\|}$:

$$\mathbb{E}\left[\|\bar{\mathbf{Q}}_\perp\| - \theta\|\bar{\mathbf{Q}}_\|\|\right] \le \mathbb{E}\left[Z(\bar{\mathbf{X}})\right] \le \kappa + \frac{16D^2}{\nu_{\min}}.$$

Let $M_\theta = \kappa + \frac{16D^2}{\nu_{\min}}$, we then have the result.

□

4.6.2 Proof of Proposition 4.2

Proof. We will derive the lower bound by bounding the expected queue length sum at each input port, in particular, $\mathbb{E}\big[\sum_j \bar{q}_{ij}(t)\big]$ for each input port i.

The queue length dynamics of the coupled queue $\phi_i(t)$ is given by:

$$\begin{aligned} \phi_i(t+1) &= \big[\phi_i(t) + b_i(t) - \mathbb{1}_{\{r(t)=0\}}\big]^+ \\ &= \phi_i(t) + b_i(t) - \mathbb{1}_{\{r(t)=0\}} + v_i(t) \end{aligned}$$

where $v_i(t)$ is the unused service and satisfies $\phi_i(t+1)v_i(t) = 0$.

We may show by induction that $\mathbb{E}\big[\sum_j \bar{q}_{ij}(t)\big] \geq \mathbb{E}\big[\bar{\phi}_i(t)\big]$. It then remains to lower bound $\mathbb{E}\big[\bar{\phi}_i(t)\big]$. We consider the expected drift of $\big(\bar{\phi}_i(t)\big)^2$ as follows:

$$\begin{aligned} &\mathbb{E}\big[\big(\bar{\phi}_i(t+1)\big)^2 - \big(\bar{\phi}_i(t)\big)^2\big] \\ =&\mathbb{E}\big[\big(\bar{\phi}_i(t) + b_i(t) - \mathbb{1}_{\{r(t)=0\}}\big)^2 - \big(v_i(t)\big)^2 - \big(\bar{\phi}_i(t)\big)^2\big] \\ =&\mathbb{E}\big[2\bar{\phi}_i(t)\big(b_i(t) - \mathbb{1}_{\{r(t)=0\}}\big) + \big(b_i(t) - \mathbb{1}_{\{r(t)=0\}}\big)^2 - \big(v_i(t)\big)^2\big] \\ =&\mathbb{E}\big[2\bar{\phi}_i(t)\big((1-\epsilon) - (1 - \mathbb{1}_{\{r(t)>0\}})\big) + \big(b_i(t) - (1-\epsilon) + (\mathbb{1}_{\{r(t)>0\}} - \epsilon)\big)^2 - v_i(t)\big] \\ =&-2\epsilon\mathbb{E}\big[\bar{\phi}_i(t)\big] + 2\mathbb{E}\big[\bar{\phi}_i(t)\big]\mathbb{E}\big[\mathbb{1}_{\{r(t)>0\}}\big] + \mathrm{Var}(b_i(t)) + \mathbb{E}\big[(\mathbb{1}_{\{r(t)>0\}} - \epsilon)^2\big] - \mathbb{E}\big[v_i(t)\big] \end{aligned}$$

where the last equality follows from the independence between the queue length process $\phi_i(t)$ and the schedule reconfiguration decision.

$$2(\epsilon - p)\mathbb{E}\big[\bar{\phi}_i(t)\big] = \sum_j \sigma_{ij}^2 + p - 2p\epsilon + \epsilon^2 - \mathbb{E}\big[v_i(t)\big]$$

Considering the drift of $\bar{\phi}_i(t)$, we could derive $\mathbb{E}\big[v_i(t)\big]$ as follows:

$$\begin{aligned}
&\mathbb{E}\big[\bar{\phi}_t(t+1) - \bar{\phi}_i(t)\big] = (1-\epsilon) - (1 - \mathbb{E}\big[\mathbb{1}_{\{r(t)>0\}}\big]) + \mathbb{E}\big[v_i(t)\big] = 0 \\
\Rightarrow &\mathbb{E}\big[v_i(t)\big] = \epsilon - \mathbb{E}\big[\mathbb{1}_{\{r(t)>0\}}\big] = \epsilon - p
\end{aligned}$$

We thus have

$$\mathbb{E}\big[\bar{\phi}_i(t)\big] = \frac{\sum\limits_j \sigma_{ij}^2}{2(\epsilon - p)} - \frac{(1-\epsilon)(\epsilon - 2p)}{2(\epsilon - p)}$$

Using $\mathbb{E}\big[\sum_j \bar{q}_{ij}(t)\big] \geq \mathbb{E}\big[\bar{\phi}_i(t)\big]$, and summing over each input port, we obtain

$$\mathbb{E}\bigg[\sum_{ij} \bar{q}_{ij}(t)\bigg] \geq \mathbb{E}\bigg[\sum_i \bar{\phi}_i(t)\bigg] \geq \frac{\sum\limits_{ij} \sigma_{ij}^2}{2(\epsilon - p)} - \frac{(1-\epsilon)(\epsilon - 2p)}{2(\epsilon - p)}$$

□

4.6.3 Proof of Proposition 4.3

Proof. From the renewal reward theory, we have

$$\mathbb{E}\bigg[\sum_{ij} Q_{ij}(t)\bigg] = \lim_{T\to\infty} \frac{1}{T} \mathbb{E}\bigg[\sum_{t=0}^{T} Q_{ij}(t)\bigg] = \frac{\mathbb{E}\bigg[\sum\limits_{t=t_k}^{t_{k+1}-1} \sum\limits_{ij} Q_{ij}(t)\bigg]}{\mathbb{E}[t_{k+1} - t_k]}$$

For any (i,j) such that $S_{ij}(t_k) = 0$, we have $Q_{ij}(t) = Q_{ij}(t_k) + \sum_{\tau=t_k}^{t-1} a_{ij}(\tau) \geq$

$\sum_{\tau=t_k}^{t-1} a_{ij}(\tau)$, while for any (i,j) such that $S_{ij}(t_k)=1$, we have $Q_{ij}(t) \geq 0$.

$$\begin{aligned}
\mathbb{E}\Big[\sum_{t=t_k}^{t_{k+1}-1}\sum_{ij} Q_{ij}(t)\Big] =& \mathbb{E}\Big[\sum_{t=t_k}^{t_{k+1}-1} \mathbb{E}\Big[\sum_{ij} Q_{ij}(t)\Big|\mathbf{Q}(t_k)\Big]\Big] \\
\geq& \mathbb{E}\Big[\sum_{t=t_k}^{t_{k+1}-1} \sum_{(i,j):S_{ij}(t_k)=0} \lambda_{ij}(t-t_k)\Big] \\
\geq& \mathbb{E}\Big[\sum_{t=t_k}^{t_{k+1}-1} \Big(t-t_k\Big)\Big(n(1-\epsilon)-\max_{\mathbf{S}}\langle \mathbf{S}, \Lambda\rangle\Big)\Big] \\
=& \mathbb{E}\Big[\frac{(t_{k+1}-t_k)^2}{2}\Big](n-\bar{\alpha})(1-\epsilon)
\end{aligned}$$

We then have a lower bound of the expected queue length sum as follows:

$$\begin{aligned}
\mathbb{E}\Big[\sum_{ij} Q_{ij}(t)\Big] =& \frac{\mathbb{E}[(t_{k+1}-t_k)^2]}{2\mathbb{E}[t_{k+1}-t_k]}(n-\bar{\alpha})(1-\epsilon) \\
\geq& \frac{\mathbb{E}[t_{k+1}-t_k]}{2}(n-\bar{\alpha})(1-\epsilon) \\
=& \frac{\Delta_r}{2p}(n-\bar{\alpha})(1-\epsilon)
\end{aligned}$$

□

4.6.4 Proof of Proposition 4.4

Proof. Given the period T, we consider the Markov chain $\mathbf{X}(t)$ being sampled at times $t_k = kT, k = 0, 1, \ldots$. Since the Fixed Frame MaxWeight policy stabilizes the system if $T > \frac{\Delta_r}{\epsilon}$, we know that $\mathbf{X}(t)$ converges to a steady state distribution, and so does $\mathbf{X}(t_k)$. Let $\bar{\mathbf{X}} = (\bar{\mathbf{Q}}, \bar{\mathrm{s}}, \bar{r})$ and $\hat{\mathbf{X}} = (\hat{\mathbf{Q}}, \hat{\mathrm{s}}, \hat{r})$ denote the steady state distribution of $\mathbf{X}(t)$ and $\mathbf{X}(t_k)$, respectively. By the assumption on the maximum arrival, we immediately have that $\mathbb{E}[\sum_{ij} \bar{q}_{ij}] \leq \mathbb{E}[\sum_{ij} \hat{q}_{ij}] + n^2 a_{\max} T$. It then remains to bound $\mathbb{E}[\sum_{ij} \hat{q}_{ij}]$ following the similar procedures in [29]:

(1) Derive an upper bound on $\mathbb{E}[\|\mathbf{Q}_{\perp}(t_k)\|^2]$

(2) Derive the queue length upper bound which depends on $\mathbb{E}[\|\mathbf{Q}_\perp(t_k)\|^2]$

Consider the Lyapunov function $Z(\mathbf{X}) = \|\mathbf{Q}_\perp\|$. By the assumption on the maximum arrival, we have

$$\begin{aligned}
|\Delta Z(\mathbf{X})| =& \Big| \|\mathbf{Q}_\perp(t_{k+1})\| - \|\mathbf{Q}(t_k)\| \Big| \\
\leq& \|\mathbf{Q}_\perp(t_{k+1}) - \mathbf{Q}(t_k)\| \\
=& \sqrt{\sum_{ij} |Q_{ij}(t_{k+1} - Q_{ij}(t_k)|^2} \\
\leq& n a_{\max} T
\end{aligned} \tag{4.39}$$

For the expected drift at steady state, we have

$$\mathbb{E}\Big[\|\mathbf{Q}_\perp(t_{k+1})\| - \|\mathbf{Q}(t_k)\| \Big] \leq \mathbb{E}\Big[\sum_{\tau=t_k}^{t_{k+1}-1} \mathbb{E}\Big[\frac{\Delta V(\mathbf{X}(\tau)) - \Delta V_{\|}(\mathbf{X}(\tau))}{2\|\mathbf{Q}_\perp\|} \Big| \mathbf{X}(\tau) \Big] \Big] \tag{4.40}$$

where $\Delta V(\mathbf{X})$ and $\Delta V_{\|}(\mathbf{X})$ are the drift of Lyapunov functions $V(\mathbf{X}) = \|\mathbf{Q}\|^2$ and $V_{\|}(\mathbf{X}) = \|\mathbf{Q}_{\|}\|^2$, respectively. Now for each $\tau \in [t_k, t_{k+1} - 1]$ we have

$$\begin{aligned}
\mathbb{E}\Big[\Delta V(\mathbf{X}(\tau)) \Big| \mathbf{X}(\tau) \Big] \leq& \|\boldsymbol{\lambda}\|^2 + \|\boldsymbol{\sigma}\|^2 + n - 2\epsilon\langle \mathbf{Q}(\tau), \boldsymbol{\nu} \rangle + 2\langle \mathbf{Q}(\tau), \boldsymbol{\nu} - \mathbf{S}^*(\tau) \rangle \\
&+ 2\langle \mathbf{Q}(\tau), \mathbf{S}^*(\tau) - \mathbf{S}(\tau) \rangle + 2\langle \mathbf{Q}(\tau), \mathrm{s}(\tau) \rangle \mathbb{1}_{\{r(\tau)>0\}} \\
\leq& \|\boldsymbol{\lambda}\|^2 + \|\boldsymbol{\sigma}\|^2 + n - 2\epsilon\langle \mathbf{Q}(\tau), \boldsymbol{\nu} \rangle - 2\nu_{\min}\|\mathbf{Q}_\perp(\tau)\| \\
&+ 2n(a_{\max} + 1)\tau + 2\langle \mathbf{Q}(\tau), \mathrm{s}(\tau) \rangle \mathbb{1}_{\{r(\tau)>0\}}
\end{aligned} \tag{4.41}$$

and

$$\mathbb{E}\Big[\Delta V_{\|}(\mathbf{X}(\tau)) \Big| \mathbf{X}(\tau) \Big] \geq -2\epsilon\langle \mathbf{Q}_{\|}(\tau), \boldsymbol{\nu} \rangle + 2\langle \mathbf{Q}_{\|}(\tau), \mathrm{s}(\tau) \rangle. \tag{4.42}$$

We then have the expected drift of $Z(\mathbf{X})$ at steady state given by

$$\begin{aligned}
&\mathbb{E}\Big[\|\mathbf{Q}_{\perp}(t_{k+1})\| - \|\mathbf{Q}(t_k)\|\Big] \\
\leq &\mathbb{E}\Bigg[\sum_{\tau=t_k}^{t_{k+1}-1}\Big(\frac{\|\boldsymbol{\lambda}\|^2+\|\boldsymbol{\sigma}\|^2+n}{2\|\mathbf{Q}(\tau)\|} - \epsilon\langle\frac{\mathbf{Q}_{\perp}(\tau)}{\|\mathbf{Q}_{\perp}(\tau)\|},\boldsymbol{\nu}\rangle - \nu_{\min} + \frac{n(a_{\max}+1)\tau}{\|\mathbf{Q}(\tau)\|} \\
&+ 2\langle\frac{\mathbf{Q}_{\perp}(\tau)}{\|\mathbf{Q}_{\perp}(\tau)\|},\mathrm{s}(\tau)\rangle\mathbb{1}_{\{r(\tau)>0\}}\Big)\Bigg] \\
\leq &T\Big(\frac{\|\boldsymbol{\lambda}\|^2+\|\boldsymbol{\sigma}\|^2+n+n(a_{\max}+1)T}{2(\|\mathbf{Q}(t_k)\|-nT)} + \epsilon\|\nu\| - \nu_{\min}\Big) + \sqrt{n}\Delta_r \\
\leq &T\Big(\frac{\|\boldsymbol{\lambda}\|^2+\|\boldsymbol{\sigma}\|^2+n+n(a_{\max}+1)T}{2(\|\mathbf{Q}(t_k)\|-nT)} - \frac{\nu_{\min}}{4}\Big) \qquad (4.43)
\end{aligned}$$

where the last inequality follows from that $\epsilon \leq \frac{\nu_{\min}}{4\|\boldsymbol{\nu}\|}$ and $T > \frac{\Delta_r}{\epsilon} \geq \frac{4\|\boldsymbol{\nu}\|\Delta_r}{\nu_{\min}} \geq \frac{4\sqrt{n}\Delta_r}{\nu_{\min}}$.

Let $\kappa = \frac{2(\|\boldsymbol{\lambda}\|^2+\|\boldsymbol{\sigma}\|^2+n)}{\nu_{\min}} + nT(\frac{2(a_{\max}+1)}{\nu_{\min}}+1)$ and we have that $\|\mathbf{Q}(t_k)\| > \kappa$ implies

$$\mathbb{E}\Big[\|\mathbf{Q}_{\perp}(t_{k+1})\| - \|\mathbf{Q}(t_k)\|\Big] \leq -\frac{\nu_{\min}}{4}T \qquad (4.44)$$

then using [29, lemma 3] with $D = na_{\max}T$ and $\eta = \frac{\nu_{\min}T}{4}$, we have

$$\begin{aligned}
\mathbb{E}\Big[\|\mathbf{Q}_{\perp}(t_k)\|^2\Big] \leq &4\kappa^2 + 32D^2\Big(1+\frac{D}{\eta}\Big)^2 \leq \Big(2\kappa + 4\sqrt{2}D(1+\frac{D}{\eta})\Big)^2 \\
\leq &T^2\Big(\frac{4na_{\max}+4(\|\boldsymbol{\lambda}\|^2+\|\boldsymbol{\sigma}\|^2+2n)+16\sqrt{2}n^2a_{\max}^2}{\nu_{\min}}\Big)^2 \\
&+ T^2\Big(2n+4\sqrt{2}na_{\max}\Big)^2 \qquad (4.45)
\end{aligned}$$

Let $M = \frac{4na_{\max}+4(\|\boldsymbol{\lambda}\|^2+\|\boldsymbol{\sigma}\|^2+2n)+16\sqrt{2}n^2a_{\max}^2}{\nu_{\min}} + 2n + 4\sqrt{2}na_{\max}$, we have $\mathbb{E}\Big[\|\mathbf{Q}_{\perp}(t_k)\|^2\Big] \leq T^2M^2$.

Consider the Lyapunov function $W(\mathbf{X}) = \sum_i\big(\sum_j Q_{ij}\big)^2 + \sum_i\big(\sum_i Q_{ij}\big)^2 - \frac{1}{n}\big(\sum_{ij} Q_{ij}\big)^2$, and set the corresponding Lyapunov drift at steady state to zero, $\mathbb{E}\Big[W(\mathbf{X}(t_{k+1})) - W(\mathbf{X}(t_k))\Big] =$

0. We then have $T_1 = T_2 + T_3 + T_4$ where

$$\begin{aligned}
T_1 =& 2\mathbb{E}\Big[\sum_i \Big(\sum_j Q_{ij}(t_k)\Big)\Big(\sum_j \sum_{\tau=t_k}^{t_{k+1}-1} (s_{ij}(\tau)\mathbb{1}_{r(\tau)=0} - a_{ij}(\tau))\Big) \\
&+ \sum_j \Big(\sum_i Q_{ij}(t_k)\Big)\Big(\sum_i \sum_{\tau=t_k}^{t_{k+1}-1} (s_{ij}(\tau)\mathbb{1}_{r(\tau)=0} - a_{ij}(\tau))\Big) \\
&- \frac{1}{n}\Big(\sum_{ij} Q_{ij}(t_k)\Big)\Big(\sum_{ij} \sum_{\tau=t_k}^{t_{k+1}-1} (s_{ij}(\tau)\mathbb{1}_{r(\tau)=0} - a_{ij}(\tau))\Big)\Big] \\
=& 2(\epsilon T - \Delta_r)\mathbb{E}\Big[\sum_{ij} Q_{ij}(t_k)\Big] \qquad (4.46)
\end{aligned}$$

$$\begin{aligned}
T_2 =& \mathbb{E}\Big[\sum_i \Big(\sum_j \sum_{\tau=t_k}^{t_{k+1}-1} (a_{ij}(\tau) - s_{ij}(\tau)\mathbb{1}_{\{r(\tau)=0\}})\Big)^2 \\
&+ \sum_j \Big(\sum_i \sum_{\tau=t_k}^{t_{k+1}-1} (a_{ij}(\tau) - s_{ij}(\tau)\mathbb{1}_{\{r(\tau)=0\}})\Big)^2 \\
&- \frac{1}{n}\Big(\sum_{ij} \sum_{\tau=t_k}^{t_{k+1}-1} (a_{ij}(\tau) - s_{ij}(\tau)\mathbb{1}_{\{r(\tau)=0\}})\Big)^2\Big] \\
=& \Big(2 - \frac{1}{n}\Big)T\sum_{ij}\sigma_{ij}^2 + n(\epsilon T - \Delta_r)^2 \qquad (4.47)
\end{aligned}$$

$$\begin{aligned}
T_3 =& \mathbb{E}\Big[-\sum_i \Big(\sum_j \sum_{\tau=t_k}^{t_{k+1}-1} u_{ij}(\tau)\Big)^2 - \sum_j \Big(\sum_i \sum_{\tau=t_k}^{t_{k+1}-1} u_{ij}(\tau)\Big)^2 \\
&+ \frac{1}{n}\Big(\sum_{ij} \sum_{\tau=t_k}^{t_{k+1}-1} u_{ij}(\tau)\Big)^2\Big] \\
\leq& nT(\epsilon T - \Delta_r) \qquad (4.48)
\end{aligned}$$

$$
\begin{aligned}
T_4 =& 2\mathbb{E}\Big[\sum_i \Big(\sum_j Q_{ij}(t_{k+1})\Big)\Big(\sum_j \sum_{\tau=t_k}^{t_{k+1}-1} u_{ij}(\tau)\Big) \\
&+ \sum_j \Big(\sum_i Q_{ij}(t_{k+1})\Big)\Big(\sum_i \sum_{\tau=t_k}^{t_{k+1}-1} u_{ij}(\tau)\Big) \\
&- \frac{1}{n}\Big(\sum_{ij} Q_{ij}(t_{k+1})\Big)\Big(\sum_{ij} \sum_{\tau=t_k}^{t_{k+1}-1} u_{ij}(\tau)\Big)\Big] \\
\le& 2na_{\max}T\mathbb{E}\Big[\sum_{ij} \sum_{\tau=t_k}^{t_{k+1}-1} u_{ij}(\tau)\Big] + 4n \sum_{\tau=t_k}^{t_{k+1}-1} \mathbb{E}\Big[\sum_{ij} u_{ij}(\tau)\Big] \sqrt{\mathbb{E}[\|\mathbf{Q}_\perp(t_k+\tau)\|^2]} \\
\le& 2n^2(\epsilon T - \Delta_r)T(a_{\max} + 2M)
\end{aligned}
\tag{4.49}
$$

Hence we have the upper bound

$$
\begin{aligned}
\mathbb{E}\Big[\sum_{ij} \bar{q}_{ij}\Big] \le& \Big(1 - \frac{1}{2n}\Big)\frac{T}{\epsilon T - \Delta_r}\|\boldsymbol{\sigma}\|^2 + \frac{n((1+\epsilon)T - \Delta_r)}{2} + n^2T(a_{\max} + 2M) \\
\le& (1 - \frac{1}{2n})\frac{T}{\epsilon T - \Delta_r}\|\tilde{\boldsymbol{\sigma}}\|^2 + T\Big(\frac{n(1+\epsilon)}{2} + n^2(a_{\max} + 2M)\Big)
\end{aligned}
\tag{4.50}
$$

□

Chapter 5

Distributed Computing Networks with Reconfiguration Delay and Cost

In this chapter, we address the scheduling problem of distributed computing networks subject to reconfiguration overhead.

The emergence of network function virtualization (NFV) and software defined networking (SDN) enables network services to be deployed in the form of interconnected software functions instantiated over commercial off-the-shelf servers at multiple cloud locations and interconnected via a programmable network fabric. This allows cloud network operators to host a large variety of services over a common general purpose infrastructure and dynamically allocate resources according to changing demands, reducing both capital and operational expenses.

The unprecedented flexibility of the cloud networking paradigm provides exciting opportunities for future service scenarios and stimulates research in key technical areas such as optimal function placement, service flow routing, and joint cloud/network resource allocation. One line of research addressed the virtual network functions placement problem from a static global optimization point of view, in which the goal is to find the placement of virtual functions and the routing of network flows that meet service demands with minimum cost [3, 9, 15, 51].

However, requirements for prior knowledge of global system information and service demands restrict the use of such centralized policies to relatively small-scale scenarios with relatively static demands. In contrast, recent works have leveraged ideas from dynamic network control to design distributed control policies for computation networks, in which nodes make local decisions on processing and transmission flow scheduling [11], as well as associated compute and network resource allocation [14, 16, 18], with global system guarantees. The work in [11] proposes a backpressure-based algorithm for maximizing the rate of queries for a computation operation on remote data, while [14, 16, 18] present cloud network control policies for service function chains that guarantee throughput-optimality and minimun average cloud network cost. While the dynamic cloud network control (DCNC) algorithm presented in [18] shows promise in serving varying workloads with minimum cost by dynamically adjusting resource allocation and scheduling decisions, it overlooks the fact that the reconfiguration of compute and network resources takes a non-negligible amount of time and may incur additional cost. As an example, starting up a virtual machine (VM) can take up to 2 minutes [38]. A control policy that is unaware of the reconfiguration delay and cost associated with the cloud and network resources, may perform excessive reconfigurations that can lead to increased congestion and overall operational cost.

The reconfiguration delay associated with flow scheduling has been studied in the context of the switch model [5, 7, 49], multi-hop networks [21, 44], and signal control in transportation systems [21]. In these works, throughput optimal scheduling policies under any finite reconfiguration delay have been proposed. However, resource allocation, and thus cost minimization, is not considered in the settings of these works. Regarding reconfiguration cost, [36] addressed the cost of flow reconfigurations in SDN by designing a control policy that minimizes total flow allocation cost subject to a given reconfiguration cost budget. In [23], the reconfiguration cost associated with switching base stations on and off in a dynamic wireless network setting was considered. The proposed approach requires arrival and channel statistics for activation decisions, and leverages an explore-exploit policy in the case that this information is not available.

5.1 System Model

A cloud network is modeled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $|\mathcal{V}| = V$ vertices and $|\mathcal{E}| = E$ edges representing cloud nodes and network links, respectively. Cloud and network resources can be allocated in terms of elemental resource units (e.g. VMs, wavelengths) and are characterized by their processing and transmission capacities and costs, respectively. We describe these parameters in the following, and provide an example illustrating how these parameters may correspond to physical parameters associated with the deployment of NFV service chains in cloud networks.

- $\mathcal{K}_i = \{0, 1, \ldots, K_i\}$: the set of possible processing resource units at node $i \in \mathcal{V}$
- $\mathcal{K}_{ij} = \{0, 1, \ldots, K_{ij}\}$: the set of possible transmission resource units at link $(i, j) \in \mathcal{E}$
- $C_i(k)$: the processing capacity resulting from the allocation of k processing resource units at node $i \in \mathcal{V}$
- $C_{ij}(k)$: the transmission capacity resulting from the allocation of k transmission resource units at link $(i, j) \in \mathcal{E}$
- $w_i(k)$: the cost of maintaining k active processing resource units at node $i \in \mathcal{V}$
- $w_{ij}(k)$: the cost of maintaining k active transmission resource units at link $(i, j) \in \mathcal{E}$
- e_i: the cost per processing flow unit at node $i \in \mathcal{V}$
- e_{ij}: the cost per transmission flow unit at link $(i, j) \in \mathcal{E}$

Fig. 5.1(a) depicts an example cloud network that hosts a service chain. Each service function in the chain may be instantiated as a VM within a cloud node. For each cloud node, K_i would then represent the total number of VMs available, and $C_i(k)$ the number of vCPUs associated with the instantiation of k VMs. The activation and load-proportional processing resource

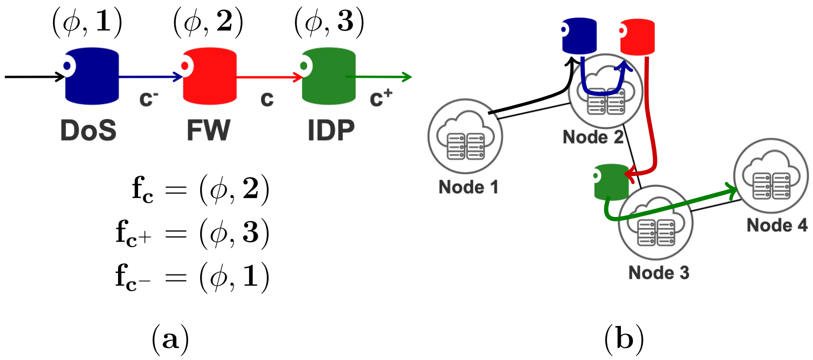
$(\phi, \mathbf{1})$
$(\phi, \mathbf{2})$
$(\phi, \mathbf{3})$
c⁻
c
c⁺
DoS
FW
IDP
$\mathbf{f_c} = (\phi, \mathbf{2})$
$\mathbf{f_{c^+}} = (\phi, \mathbf{3})$
$\mathbf{f_{c^-}} = (\phi, \mathbf{1})$
(a)
Node 1
Node 2
Node 3
Node 4
(b)

Figure 5.1: (a) A service function chain composed of Denial of Service, Firewall, and Deep Packet Inspections functions, and (b) its instantiation in a 4-node cloud network.

cost, $w_i(k)$ and e_i, may either represent power consumption or price paid to the cloud provider for the activation and usage of the allocated VMs. Regarding the allocation of transmission resources, K_{ij} may represent the total number of available bandwidth resource blocks and $C_{ij}(k)$ the transmission rate associated with the allocation of k resource blocks. The corresponding transmission costs, $w_{ij}(k)$ and e_{ij}, may again represent power consumption or prices paid to a cloud and/or network provider for the activation and usage of bandwidth resources. Depending on the application, an operator may also include propagation delay as an additional penalty term in the transmission resource cost to favor shorter propagation delay.

Throughout the rest of the discussion, we make the following assumption on the capacities and costs of cloud and network resources:

Assumption 5.1. *For any node $i \in \mathcal{V}$ and any link $(i, j) \in \mathcal{E}$, we assume that both the capacity and the cost are strictly increasing with the amount of resource assigned. In other words, given any node i, for any k such that $0 \le k \le K_i - 1$, we have $C_i(k) < C_i(k+1)$ and $w_i(k) < w_i(k+1)$; similarly, given any link (i, j), for any k such that $0 \le k \le K_{ij} - 1$, we have $C_{ij}(k) < C_{ij}(k+1)$ and $w_{ij}(k) < w_{ij}(k+1)$.*

5.1.1 Service model

Cloud network $\mathcal{G}$ offers a set of services Φ. Each service $\phi \in \Phi$ is described by a chain of service functions. We let $\mathcal{M}_\phi = \{1, 2, \ldots, M_\phi\}$ denote the ordered set of functions of service ϕ, hence the tuple (ϕ, m) represents the m-th function of service ϕ. Fig. 5.1(b) illustrates an example of a network security service [32] consisting of denial-of-service (DoS) protection, firewall (FW), and intrusion detection/protection (IDP). Denote the security service as ϕ, and the DoS service function is then denoted as $(\phi, 1)$, etc.

In order to describe the flow of packets through a service chain, we adopt a multi-commodity-chain flow model as in [3, 16, 17], in which a commodity represents the flow of packets at a given stage of a service chain. In particular, a commodity-c flow is specified by source node s_c, destination node d_c, and function $f_c = (\phi, m)$, indicating the flow of packets with origin at s_c and destination at d_c that have been processed by the first m functions of service ϕ. For ease of exposition, we let c^+ and c^- denote the commodities that succeed and precede commodity c in its service chain, respectively.

Each service function has potentially distinct processing requirement, which may also vary between cloud locations. We let $\rho_i^{(c)}$ denote the processing-transmission flow ratio of function f_c at node i. That is, when one transmission flow unit of commodity c goes through function f_c at node i, it occupies $\rho_i^{(c)}$ processing flow units. In addition, our service model also captures the possibility of flow scaling. We denote by $\xi^{(c)} > 0$ the scaling factor of function f_c, indicating that function f_c generates an average of $\xi^{(c)}$ output packets of commodity c per input packet of commodity c^-.

5.1.2 Reconfiguration Delay and Cost

We consider cloud network control policies that adjust the schedule of commodity flows, as well as the allocation of cloud and network resources, according to changing demands. We assume that such reconfiguration may incur the following two types of overhead:

- Reconfiguration delay (time): This is the time duration for the reconfiguration process to complete. We assume that during the reconfiguration process, the associated function (transmission or processing of commodity flows) is not available. We denote by δ_i the reconfiguration delay for node $i \in \mathcal{V}$, and by δ_{ij} the reconfiguration delay for link $(i, j) \in \mathcal{E}$.

- Reconfiguration cost: This is the cost/penalty associated with each reconfiguration operation. Let η_i denote the reconfiguration cost for node $i \in \mathcal{V}$, and η_{ij} denote the reconfiguration cost for link $(i, j) \in \mathcal{E}$.

It is important to note that reconfiguration delay indicates the time during which traffic is not being served, and hence has a direct impact on throughput performance. For example, the more time it takes to start a VM or to redirect a traffic flow in a software-defined switch, the more traffic is blocked during the reconfiguration process. On the other hand, reconfiguration cost accounts for any other cost penalty associated with the reconfiguration process that is not causing a blockage of traffic flow, e.g., power consumption and/or hardware degradation when switching on/off physical resources. While reconfiguration cost does not affect throughput performance, it impacts overall resource cost (e.g., total power consumption).

In the rest of the paper, we use Δ to denote the reconfiguration delay and cost structure of a cloud network, where $\Delta = \Big\{\{\delta_i\}_{i\in\mathcal{V}}, \{\delta_{ij}\}_{(i,j)\in\mathcal{E}}, \{\eta_i\}_{i\in\mathcal{V}}, \{\eta_{ij}\}_{(i,j)\in\mathcal{E}}\Big\}$.

We consider a time slotted system with slots normalized to integral units $t \in \{0, 1, 2, \dots\}$. Suppose that node i reconfigures the processing resource allocation or the commodity being processed at time t. Then, flow processing at node i becomes unavailable during time period $[t, t + \delta_i]$, and a reconfiguration cost η_i is incurred at time t. Similarly, suppose that link (i, j) reconfigures the transmission resource allocation or the commodity being transmitted at time t. Then, flow tranmission is unavailable during $[t, t + \delta_{ij}]$, and a reconfiguration cost η_{ij} is incurred at time t.

Note that we consider a worst-case reconfiguration delay model in that we assume complete unavailability of packet processing or transmission functionality at a node or link undergoing reconfiguration. Importantly, a throughput-optimal policy for this worst-case reconfiguration delay model will guarantee throughput-optimality for any other less restrictive model.

For ease of discussion in the following, we also define $r_i(t)$ and $r_{ij}(t)$ to denote the reconfiguration status:

- $r_i(t)$: the time remaining in the reconfiguration process at node $i \in \mathcal{V}$
- $r_{ij}(t)$: the time remaining in the reconfiguration process at link $(i, j) \in \mathcal{E}$, where $i, j \in \mathcal{V}$

By definition, these processes evolve as follows: At any time t, if node i (or link (i, j)) reconfigures, then set $r_i(t) = \delta_i$ (or $r_{ij}(t) = \delta_{ij}$, respectively); otherwise, set $r_i(t) = [r_i(t-1) - 1]^+$ (or $r_{ij}(t) = [r_{ij}(t-1) - 1]^+$, respectively).

5.1.3 Queueing Model

Let $Q_i^{(c)}(t)$ denote the queue backlog of commodity-c packets at node i at the beginning of time slot t. We denote by $a_i^{(c)}(t)$ the exogenous arrivals of commodity-c packets at node i during time slot t. Throughout this paper, we make the following assumptions for the exogenous arrival processes.

Assumption 5.2. *Each exogenous arrival process is independent and identically distributed (i.i.d.) over time, with $\mathbb{E}[a_i^{(c)}(t)] = \lambda_i^{(c)}$. Furthermore, each exogenous arrival process has bounded support. In other words, there exist $a_{\max} < \infty$ such that $a_i^{(c)}(t) \leq a_{\max}$, $\forall i \in \mathcal{V}, c \in \mathcal{C}$, $\forall t$.*

At each time slot t, each node makes the following transmission and processing scheduling and resource allocation decisions:

- $\mu_i^{(c)}(t)$: the flow rate of commodity c being processed at node i at time t

- $\mu_{ij}^{(c)}(t)$: the flow rate of commodity c on link (i,j) at time t
- $k_i(t)$: the number of processing resource units allocated to node i at time t
- $k_{ij}(t)$: the number of transmission resource units allocated to link (i,j) at time t

With the aforementioned setup, we may write the queue dynamics for each commodity $c \in \mathcal{C}$ at each node i:

$$\begin{aligned} Q_i^{(c)}(t+1) = &\Bigg[Q_i^{(c)}(t) - \sum_{j \in \mathcal{V}^+(i)} \mu_{ij}^{(c)}(t)\mathbb{1}_{\{r_{ij}(t)=0\}} - \mu_i^{(c)}(t)\mathbb{1}_{\{r_i(t)=0\}}\Bigg]^+ \\ &+ \sum_{j \in \mathcal{V}^-(i)} \mu_{ji}^{(c)}(t)\mathbb{1}_{\{r_{ji}(t)=0\}} + \xi^{(c)}\mu_i^{(c^-)}(t)\mathbb{1}_{\{r_i(t)=0\}} + a_i^{(c)}(t), \end{aligned} \tag{5.1}$$

where $\mathcal{V}^+(i)$ and $\mathcal{V}^-(i)$ denote the set of outgoing and incoming neighbors of node i, respectively.

Observe from (5.1) that the serving rate of the queue of commodity c at node i is composed of the transmission rate of commodity c of all outgoing links and the local processing rate of commodity c. On the other hand, the arrival rate is composed of the transmission rate of commodity c of all incoming links and the local processing rate of the preceeding commodity in the service chain c^-. It is important to note that there is no contribution to both serving and arrival rates from those transmission and processing resources that are undergoing reconfiguration (i.e., $r_{ij}(t) > 0$ or $r_i(t) > 0$), indicating the inability to transmit or process packets during the reconfiguration process.

5.1.4 Problem Formulation

Given a set of service demands with average input rate matrix $\boldsymbol{\lambda} = \{\lambda_i^{(c)}\}_{i \in \mathcal{V}, c \in \mathcal{C}}$, the goal is to support the demand while minimizing the average cloud network cost.

We assume that when a processing/transmission resource is undergoing reconfiguration, processing/transmission allocation cost is not incurred since the resource is not operative until the reconfiguration process completes. Hence, we can write the total cloud network cost at time t as

$$\begin{aligned} h(t) = \sum_{i \in \mathcal{V}} \Big(&(e_i \mu_i(t) + w_i(k_i(t))) \mathbb{1}_{\{r_i(t)=0\}} + \eta_i \mathbb{1}_{\{(\boldsymbol{\mu}_i(t), k_i(t)) \neq (\boldsymbol{\mu}_i(t-1), k_i(t-1))\}} \Big) \\ + \sum_{(i,j) \in \mathcal{E}} \Big(&(e_{ij} \mu_{ij}(t) + w_{ij}(k_{ij}(t))) \mathbb{1}_{\{r_{ij}(t)=0\}} + \eta_{ij} \mathbb{1}_{\{(\boldsymbol{\mu}_{ij}(t), k_{ij}(t)) \neq (\boldsymbol{\mu}_{ij}(t-1), k_{ij}(t-1))\}} \Big) \end{aligned} \tag{5.2}$$

We can then formulate the dynamic cloud network control problem under reconfiguration delay/cost as follows. Given an input rate matrix $\boldsymbol{\lambda}$ in the capacity region:

$$\min \ \limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[h(\tau)] \tag{5.3a}$$

$$\text{s.t. The cloud network is rate stable with input rate } \boldsymbol{\lambda}$$

$$\text{and under queue length dynamics (5.1)} \tag{5.3b}$$

$$\sum_{c \in \mathcal{C}} \mu_i^{(c)}(\tau) \leq \mu_i(\tau) \leq C_i(k_i(\tau)), \qquad \forall i, \tau \tag{5.3c}$$

$$\sum_{c \in \mathcal{C}} \mu_{ij}^{(c)}(\tau) \leq \mu_{ij}(\tau) \leq C_{ij}(k_{ij}(\tau)), \quad \forall (i,j), \tau \tag{5.3d}$$

$$\mu_i^{(c)}(\tau) \geq 0, \ \mu_{ij}^{(c)}(\tau) \geq 0, \qquad \forall i, (i,j), c, \tau \tag{5.3e}$$

$$0 \leq k_i(\tau) \leq K_i, \ 0 \leq k_{ij}(\tau) \leq K_{ij}, \ \forall i, (i,j), \tau \tag{5.3f}$$

5.2 Impact of Reconfiguration Delay/Cost

In this section, we discuss the impact of reconfiguration delay and cost on the performance of a cloud network control policy that is unaware of such reconfiguration delay/cost.

We start with the characterization of the cloud network capacity region and the minimum average cloud network cost required for network stability. The cloud network capacity region $\boldsymbol{\Lambda}_\Delta$ is defined as the closure of all input rate matrices that can be stabilized by some cloud network control policy, given the cloud network structure $(\mathcal{G}, \Phi, \Delta)$. For each rate matrix $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_\Delta$, we denote by $h^*_\Delta(\boldsymbol{\lambda})$ the minimum average cost required for network stability.

The following theorem establishes that the capacity region and the minimum average cost for each arrival rate in the capacity region remains the same for any finite reconfiguration delay and cost. The proof of Theorem 5.1 is given in Appendix D.

Theorem 5.1. *Given any finite reconfiguration delay/cost structure Δ, the capacity region $\boldsymbol{\Lambda}_\Delta$ remains the same. In particular, $\boldsymbol{\Lambda}_\Delta = \boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ is the capacity region of the cloud network without reconfiguration delay, as characterized in [16, Theorem 1]. Furthermore, given any exogenous arrival rate matrix $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_\Delta$, we have $h^*_\Delta(\boldsymbol{\lambda}) = h^*(\boldsymbol{\lambda})$.*

While it was shown in [17] that under the setup of no reconfiguration delay and cost, DCNC policy is throughput optimal and achieves a $[O(1/V), O(V)]$ cost-delay tradeoff, the result does not hold for the case in which reconfiguration delay or cost exists. In fact, as it will be shown in section 5.5 (Figs. 5.3 and 5.8), DCNC policy loses throughput optimality and the ability to achieve minimum average cost under the presence of reconfiguration delay or cost.

In the next section, we propose Adaptive DCNC (ADCNC) policy, which is an online distributed policy for cloud network control under reconfiguration delay and cost. We then establish theoretical performance guarantees of ADCNC policy, specifically throughput optimality and the $[O(1/V), O(V)]$ cost-delay tradeoff. In other words, ADCNC policy recovers the performance guarantees that DCNC policy loses when reconfiguration overhead exists.

5.3 Adaptive Dynamic Cloud Network Control Policy

At each time slot t, each cloud network node makes local processing and transmission decisions on its corresponding outgoing interfaces.

The ADCNC policy requires the selection of a function g satisfying the following condition and a parameter $V \in \mathbb{R}^+$.

Condition 5.1. *Function $g : \mathbb{R}^+ \to \mathbb{R}^+$ satisfies:*

(1) $g(0) = 0$*,*

(2) g *is strictly increasing, and*

(3) g *is a sublinear function, i.e.* $\lim_{x \to \infty} \frac{g(x)}{x} = 0$.

With the given function g and parameter V, node $i \in \mathcal{V}$ makes the following transmission and processing decisions at each time slot t:

- **Transmission decisions:** For each neighbor $j \in \mathcal{V}^+(i)$

  1. Compute the *transmission max-utility-weight* as

  $$W_{ij}^*(t) = \max_{\substack{k \in \mathcal{K}_{ij} \\ c \in \mathcal{C}}} \left\{ C_{ij}(k) \left[Q_i^{(c)}(t) - Q_j^{(c)}(t) - Ve_{ij} \right]^+ - Vw_{ij}(k) \right\} \tag{5.4}$$

  with k^*, c^* being its maximizers.

  2. Let $(\bar{k}, \bar{c})$ denote the schedule at time $t - 1$. Compute the *transmission weight* as

  $$W_{ij}(t) = C_{ij}(\bar{k}) \left[Q_i^{(\bar{c})}(t) - Q_j^{(\bar{c})}(t) - Ve_{ij} \right]^+ - Vw_{ij}(\bar{k}) \tag{5.5}$$

  and the *transmission weight differential* as

  $$\Delta W_{ij}(t) = W_{ij}^*(t) - W_{ij}(t) \tag{5.6}$$

  3. Define the *transmission weight differential threshold* at time t as

  $$\theta_{ij}(t) = g\Big(C_{ij}(\bar{k}) (Q_i^{(c^*)}(t) - Q_j^{(c^*)}(t)) \Big) \tag{5.7}$$

and set the transmission resource-commodity schedule at time t as

$$(k(t), c(t)) = \begin{cases} (k^*, c^*) & \text{if } \Delta W_{ij}(t) > \theta_{ij}(t) \\ (\bar{k}, \bar{c}) & \text{otherwise} \end{cases} \tag{5.8}$$

4. Allocate $k(t)$ transmission resource units and set the transmission flow rates as:

$$\mu_{ij}^{(c)}(t) = C_{ij}(k(t))\mathbb{1}_{\{c=c(t)\}}, \forall c \in \mathcal{C}$$

- **Processing decisions:**

1. Compute the *processing max-utility weight* as

$$W_i^*(t) = \max_{\substack{k \in \mathcal{K}_i \\ c \in \mathcal{C}}} \left\{ \frac{C_i(k)}{\rho_i^{(c)}} \left[Q_i^{(c)}(t) - \xi^{(c^+)} Q_i^{(c^+)}(t) - Ve_i \right]^+ - Vw_i(k) \right\} \tag{5.9}$$

with k^*, c^* being its maximizers.

2. Let $(\bar{k}, \bar{c})$ denote the schedule at time $t-1$. Compute the *processing weight* as

$$W_i(t) = \frac{C_i(\bar{k})}{\rho_i^{(\bar{c})}} \left[Q_i^{(\bar{c})}(t) - \xi^{(\bar{c}^+)} Q_i^{(\bar{c}^+)}(t) - Ve_i \right]^+ - Vw_i(\bar{k}) \tag{5.10}$$

and the *processing weight differential* as

$$\Delta W_i(t) = W_i^*(t) - W_i(t) \tag{5.11}$$

3. Define the *processing weight differential threshold* at time t as

$$\theta_i(t) = g\Big(C_i(\bar{k})(Q_i^{(c^*)}(t) - Q_i^{({c^*}^+)}(t)) \Big) \tag{5.12}$$

and set the processing resource-commodity schedule at time t as

$$(k(t), c(t)) = \begin{cases} (k^*, c^*) & \text{if } \Delta W_i(t) > \theta_i(t) \\ (\bar{k}, \bar{c}) & \text{otherwise} \end{cases} \tag{5.13}$$

4. Allocate $k(t)$ processing resource units and set the processing flow rates as:

$$\mu_i^{(c)}(t) = C_i(k(t))\mathbb{1}_{\{c=c(t)\}}, \forall c \in \mathcal{C} \tag{5.14}$$

5.4 Throughput Optimality and Cost Minimization

In this section, we extend the drift-plus-penalty analysis of [34] to show that Adaptive DCNC is throughput-optimal and achieves $[O(1/V), O(V)]$ average cost-delay tradeoff with probability 1 (w.p. 1) under any finite reconfiguration delay/cost.

The stability of Adaptive DCNC relies on the fact that it allows each node and link to implicitly adjust the frequency of reconfiguration according to its maximal queue length differential. In particular, under Adaptive DCNC, the frequency of reconfiguration will decrease if the maximal queue length differential increases. This behavior may be characterized by the following lemma.

Lemma 5.1. *Suppose Assumption 5.1 and 5.2 hold, and the cloud network is operated under Adaptive DCNC with parameter V and sublinear function g. Given any fixed integer T, if the maximal queue length differential at a link $(i, j) \in \mathcal{E}$ at time t,* i.e. $\max_c \left\{Q_i^{(c)}(t) - Q_j^{(c)}(t)\right\}$*, is greater than a constant M_{ij} as defined below in (5.15) , then link (i, j) reconfigures at most once during $[t, t + T]$.*

Similarly, if the maximal queue length differential at a node $i \in \mathcal{V}$ at time t, i.e. $\max_c \left\{Q_i^{(c)}(t) - \xi^{(c^+)}Q_i^{(c^+)}(t)\right\}$*, is greater than a constant M_i as defined below in (5.16) , then* *node i reconfigures at most once during $[t, t+T]$.*

$$M_{ij} = \max\left\{V\Big(\min_{k>0}\frac{w_{ij}(k)}{C_{ij}(k)} + e_{ij}\Big) + T\gamma_{\max}, \frac{1}{C_{ij}(1)}g^{-1}\left(2C_{ij,\max}T\gamma_{\max}\right) + T\gamma_{\max}\right\} \quad (5.15)$$

$$M_{i} = \max\left\{V\Big(\min_{k>0}\frac{w_{ij}(k)}{C_{ij}(k)} + e_{ij}\Big) + T\gamma_{\max}, \frac{1}{C_{i}(1)}g^{-1}\left(2C_{i,\max}T\gamma_{\max}\right) + T\gamma_{\max}\right\} \quad (5.16)$$

where $C_{ij,\max} = \max_k C_{ij}(k)$, $C_{i,\max} = \max_k C_i(k)$, $\gamma_{\max} = 2a_{\max} + 2C_{\max}(v_{\max}+1)$, $v_{\max} = \max_i\{\max\{|\mathcal{V}^+(i)|, |\mathcal{V}^-(i)|\}\}$, and $C_{\max} = \max\{\max_{(i,j)\in\mathcal{E}} C_{ij,\max}, \max_{i\in\mathcal{V}} C_{i,\max}\}$.

With Lemma 5.1 limiting the frequency of reconfiguration, and the weight differentials $\{\Delta W_i(t)\}_{i\in\mathcal{V}}$, $\{\Delta W_{ij}(t)\}_{(i,j)\in\mathcal{E}}$ being bounded by local thresholds that are growing sublinearly with the local maximal queue length differential, we then extend the drift-plus-penalty analysis of [34] to prove the following performance guarantee.

Theorem 5.2. *Suppose the arrival rate matrix $\boldsymbol{\lambda} = (\lambda_i^{(c)})$ is strictly interior to the capacity region $\boldsymbol{\Lambda}$, i.e. there exists a positive constant ϵ satisfying $(\boldsymbol{\lambda} + \epsilon\mathbf{1}) \in \boldsymbol{\Lambda}$, where $\mathbf{1}$ is a matrix of all ones. If all reconfiguration delays and costs in Δ are finite, and function g satisfies Condition 5.1, then Adaptive DCNC stabilizes the cloud network, while achieving arbitrarily close to minimum average cost $h^*(\boldsymbol{\lambda})$ w.p. 1, i.e.*

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t} h(\tau) \le h^*(\boldsymbol{\lambda}) + \frac{B}{V} \quad (5.17)$$

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t}\sum_{\substack{i\in\mathcal{V},\\ c\in\mathcal{C}}} Q_i^{(c)}(\tau) \le \frac{B + V[h^*(\boldsymbol{\lambda}+\epsilon\mathbf{1}) - h^*(\boldsymbol{\lambda})]}{\epsilon} \quad (5.18)$$

where B is a constant that is dependent on the system parameters $(\mathcal{G}, \Phi, \Delta)$, $C_i(k)$, $C_{ij}(k)$, $w_i(k)$, $w_{ij}(k)$, $a_{\max}$.

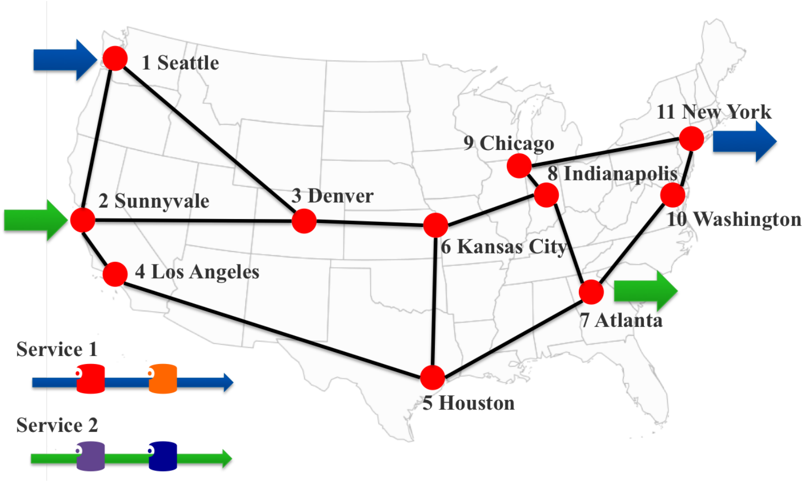

Figure 5.2: Abilene US network topology. The numbers for each cloud node corresponds to that in Tables 5.1 and 5.2.

5.5 Simulations

In this section, we present simulation results for the proposed Adaptive DCNC policy and compare with benchmark policies. The sublinear function g for Adaptive DCNC is selected as $g(x) = (1-\gamma)x^{1-\delta}$ for all simulations, and unless otherwise stated, $\delta = 0.01$ and $\gamma = 0.01$. For this time slotted cloud network, we assume each time slot to be of 10 ms.

We consider a cloud network with network topology based on the Abilene US Network, as shown in Fig. 5.2. We assume that all nodes are cloud nodes that can host all service functions.

For the processing resource parameters, we assume that each cloud node i can instantiate a maximum of $K_i = 5$ VMs, with one vCPU per VM, i.e. $C_i(k) = k$ vCPUs, $k \in \{1, \dots, 5\}$. The associated processing resource costs are estimated from power consumption and local energy cost data from [33] and [30], and the resulting per-timeslot processing resource costs are listed in Table 5.1. We assume a linear cost structure such that $w_i(k) = k \cdot w_i(1)$.

On the other hand, for the transmission resource parameters, we assume that each link (i, j) can allocate a maximum of $K_{ij} = 5$ bandwidth channels of 1 Gbps each, i.e. $C_{ij}(k) = k$ Gbps, $k \in \{1, \dots, K_{ij}\}$. The transmission resource costs are estimated based on AWS Direct Connect pricing [1], and modified to take link distance into account. The resulting per-timeslot transmission resource costs are listed in Table 5.2. We assume a linear cost structure such that $w_{ij}(k) = k \cdot w_{ij}(1)$.

Table 5.1: Processing resource cost per time slot

Node	1	2	3	4	5	6	7	8	9	10	11
$w_i(1)$ ($\times 10^{-8}$ \$)	3.59	6.50	4.21	6.50	3.43	4.19	4.01	4.25	3.69	4.84	5.72

Table 5.2: Transmission resource cost per time slot

Link	(1,2)	(1,3)	(2,3)	(2,4)	(3,6)	(4,5)	(5,6)	(5,7)	(6,8)	(7,8)	(7,10)	(8,9)	(9,11)	(10,11)
$w_{ij}(1)$ ($\times 10^{-8}$ \$)	9.4	12.4	12.3	6.3	6.5	14.0	8.8	9.4	8.56	7.5	8.1	5.2	9.2	5.4

We consider 2 network service chains, each composed of 2 distinct functions. The scaling factors of all functions are assumed to be $\xi^{(c)} = 1$, and the processing-transmission flow ratios are $\rho_i^{(c)} = 1$ vCPU / Gbps for all nodes and service functions. We assume each service is requested by one source-destination pair. For Service 1, the source is in Seattle and the destination is in New York; while for Service 2, the source is in Sunnyvale and the destination is in Atlanta. The arrival processes for both flows are i.i.d. Poisson with arrival rates denoted by λ_1 and λ_2, respectively. Throughout the simulation, we set both arrival rates to the same value, denoted by λ, i.e. $\lambda_1 = \lambda_2 = \lambda$.

For ease of discussion, we separate cases to illustrate different effects of reconfiguration delay and reconfiguration cost. In the following subsections, we vary the reconfiguration delay while fixing the reconfiguration cost, and vice versa.

5.5.1 Reconfiguration Delay

We first consider the case of cloud networks with reconfiguration delay only, in other words, the reconfiguration costs are set to zero. In this subsection, the reconfiguration delay of all the processing and transmission resources are set to the same value, denoted by δ_r.

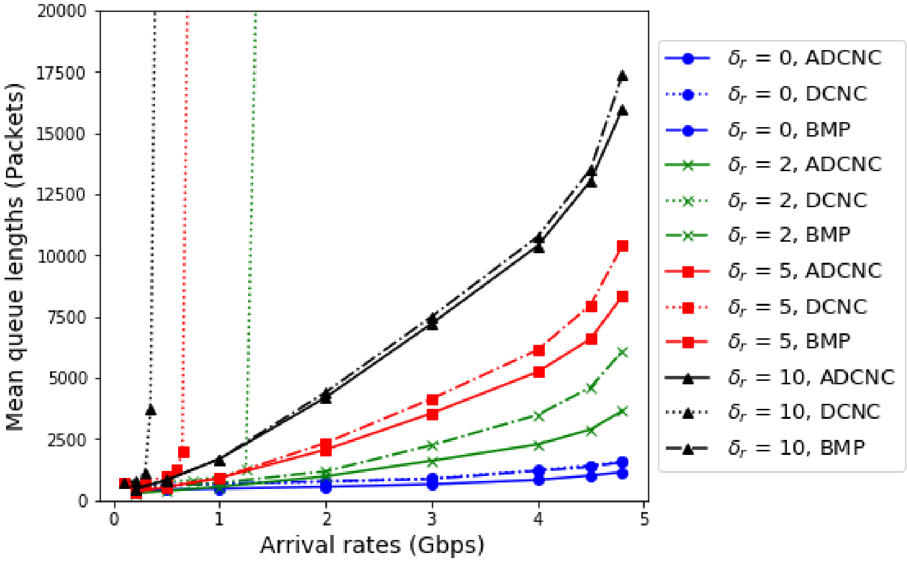

Figure 5.3: Mean total queue length for BMP, DCNC, and ADCNC under various flow arrival rates. Parameter $V = 0$.

Fig. 5.3 compares the mean (time average) total queue length between ADCNC and benchmark policies, DCNC [17] and BMP [21], under various flow arrival rates λ. Note that the BMP policy is proposed in the context where the cost minimization is not addressed, hence we make the parameter V fixed as $V = 0.0$ for all policies to ignore the cost minimization aspect, and focus only on the throughput performance. Given the topology and the processing and transmission capacity setting, the rate pair $(\lambda_1, \lambda_2) = (5.0, 5.0)$ Gbps is at the boundary of the capacity region, hence we consider the interval $\lambda \in [0, 5.0)$. It is clear from Fig. 5.3 that when the reconfiguration delay δ_r is nonzero, DCNC loses throughput-optimality, and the maximum arrival rate it can stabilize reduces as the reconfiguration delay δ_r increases. In contrast, ADCNC and BMP guarantee the throughput-optimality irrespective of the finite reconfiguration delay value. We also see that while ADCNC and BMP have comparable delay performance, ADCNC has slightly shorter queue lengths especially for small reconfiguration delays. In the following, we further look into the cost-delay tradeoff performance for DCNC and ADCNC.

In Fig. 5.4, we plot the mean (time average) network cost versus the mean total queue

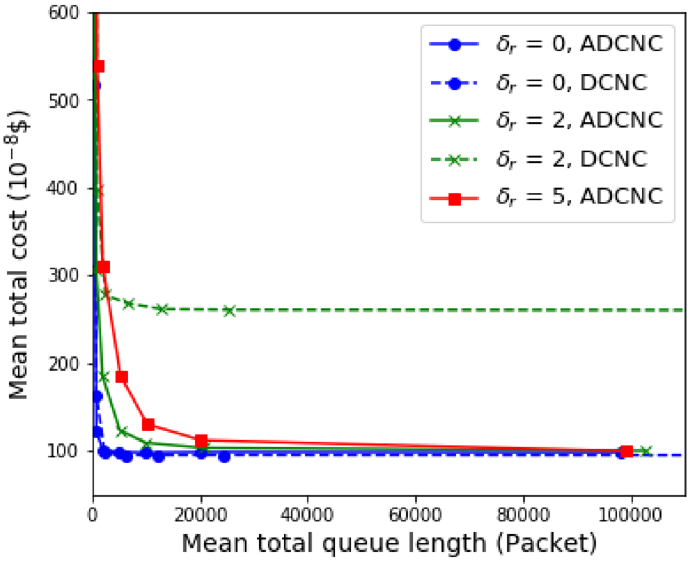

Figure 5.4: Mean cost versus mean queue length for DCNC and ADCNC under various reconfiguration delays. Arrival rate $\lambda = 1.0$ Gbps.

length for DCNC and ADCNC under various reconfiguration delays. The arrival rate is fixed to $\lambda = 1.0$ Gbps. Note that for each curve, the control parameter V tunes the tradeoff between network cost and total queue length. The closer a curve is to the lower-left corner, the better the performance (cost-delay tradeoff). Note that without reconfiguration delay, $\delta_r = 0$, DCNC and ADCNC have similar performance. As δ_r increases, the performance of the two policies starts to degrade. Nevertheless, ADCNC always guarantees throughput-optimality, and it is able to push the mean network cost arbitrarily close to minimum at the expense of increased mean queue length. In contrast, DCNC has significantly larger performance degradation as δ_r increases, and does not guarantee throughput-optimality. In fact, for $\delta_r = 5$, DCNC cannot even stabilize the arrival rate of $\lambda = 1.0$ Gbps, hence the absence of the associated cost-delay curve.

In Fig. 5.5, we further look into the reconfiguration behavior of both DCNC and ADCNC under various values of the control parameter V, with reconfiguration delay fixed to $\delta_r = 2$, and arrival rate $\lambda = 1.0$ Gbps. The vertical axis represents the fraction of time that a given transmission/processing resource is under reconfiguration, averaged over all resources, i.e., the time overhead caused by the reconfiguration delay. We first notice that ADCNC spends much less

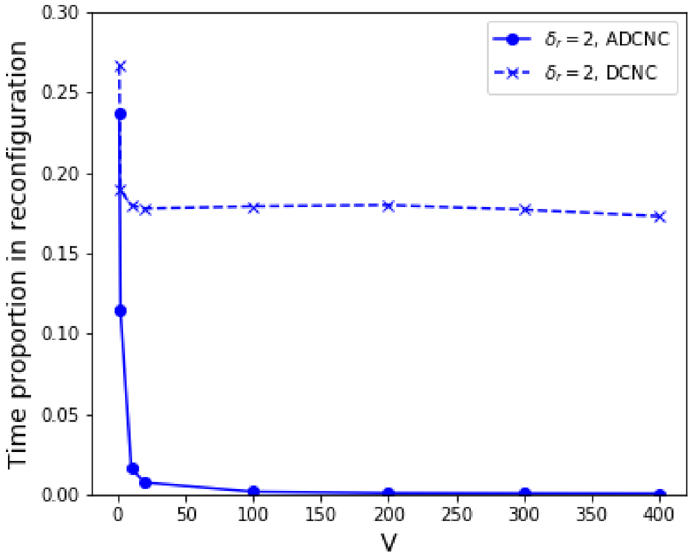

Figure 5.5: Mean fraction of time under reconfiguration for various parameter V.

time under reconfiguration, which is one of the key reasons for ADCNC to preserve throughput-optimality under finite reconfiguration delay. We then notice that while increasing the parameter V helps reducing reconfiguration overhead for both policies, DCNC spends a significantly higher fraction of time under reconfiguration even for large V.

We now consider the selection of the function $g(x)$ used in ADCNC. As specified in the simulation settings, we fix the form of $g(x) = (1-\gamma)x^{1-\delta}$ and look at the effect of varying the parameters δ and γ. In Fig. 5.6, we fix $\gamma = 0.01$, i.e., $g(x) = 0.99x^{1-\delta}$, and vary δ in the exponent. We can see that the performance improves as δ approaches 0, suggesting that the performance is better when $g(x)$ approaches a linear function. Notice a caveat here is that the linear function does not satisfy Condition 5.1 for $g(x)$, and hence does not guarantee the throughput optimality and average cost upper bounds from Theorem 5.2. On the other hand, in Fig. 5.7, we fix $\delta = 0.01$, i.e., $g(x) = (1-\gamma)x^{0.99}$, and vary γ in the exponent.

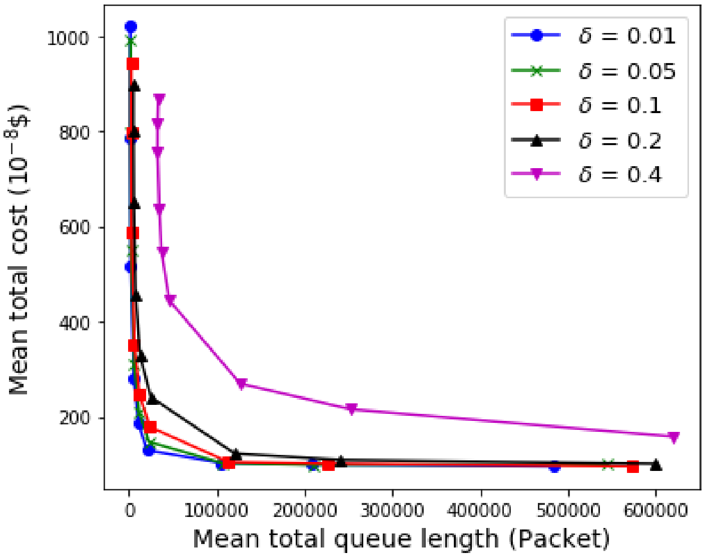

Figure 5.6: Mean cost versus mean queue length for ADCNC with different hysteresis function $g(x) = 0.99x^{1-\delta}$. Arrival rate $\lambda = 1.0$.

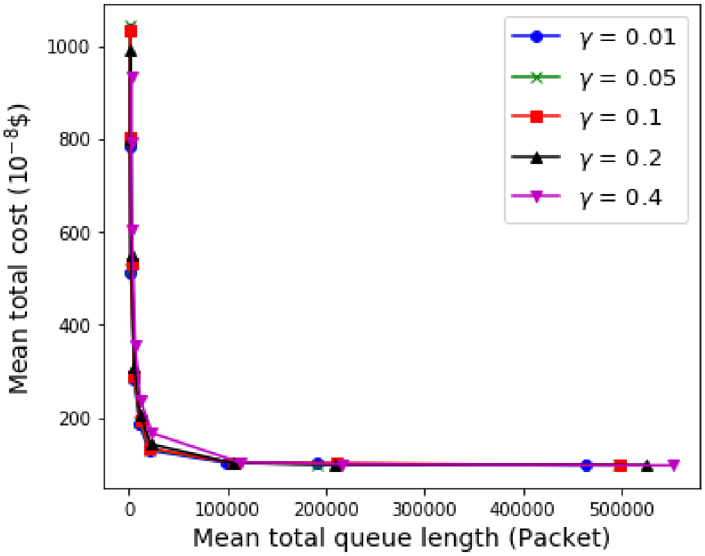

Figure 5.7: Mean cost versus mean queue length for ADCNC with different hysteresis function $g(x) = (1-\gamma)x^{0.99}$. Arrival rate $\lambda = 1.0$.

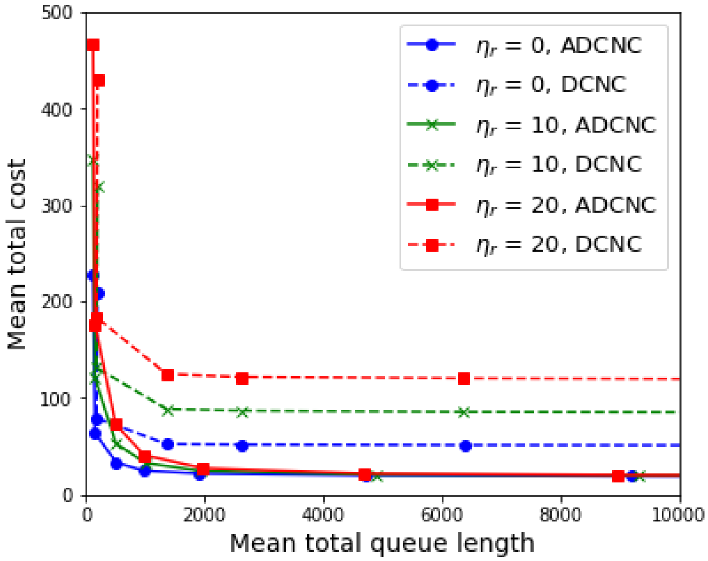

Figure 5.8: Mean cost versus mean queue length for DCNC and ADCNC under various reconfiguration costs. Reconfiguration delay $\delta_r = 2$.

5.5.2 Reconfiguration Cost

In this subsection, we set the reconfiguration delay to a fixed value ($\delta_r = 2$), and set the reconfiguration cost of all the processing and transmission resources to be the same value, denoted by η_r. Given the reconfiguration delay, we select the arrival rate low enough such that DCNC is rate stable, and compare the cost-delay tradeoff performance between DCNC and ADCNC.

Fig. 5.8 shows the cost-delay tradeoff achieved by DCNC and ADCNC under various reconfiguration costs η_r. We first notice that as the reconfiguration cost increases, DCNC can no longer achieve arbitrarily close to the minimum average cost, even when the parameter V is tuned to endure large mean total queue length. On the other hand, Adaptive DCNC is able to achieve arbitrarily close to the minimum cost under any reconfiguration cost η_r.

5.6 Heuristic Variants of ADCNC Policy

5.6.1 ADCNC in Generalized Settings

In this subsection, we briefly discuss interesting extensions to the current model that can be captured with a slight modification of the analysis.

(1) *Different reconfiguration delay/cost for resource allocation and commodity allocation*: In this paper, we have assumed that the same reconfiguration delay and cost are incurred upon any change in either the allocation of resources or the commodity being processed/transmitted. In practice, different delays and costs can be associated with different reconfiguration operations. It is rather straightforward to show that ADCNC would preserve throughput and cost optimality for any finite values of such heterogeneous delays and costs by treating any change as incurring the maximum of such delays/costs. However, improved policies (in terms of cost-delay tradeoff) could be designed in such settings. In the next subsection, we provide an example heuristic variant of the ADCNC policy to improve the cost-delay performance under this setting.

(2) *Partial reconfiguration:* In this paper, we consider a worst-case reconfiguration delay model in the sense that we assume complete unavailability of packet processing or transmission functionality at a node or link undergoing reconfiguration. In practice, there may be cases in which adding or removing resources without changing the allocated commodity only reduces the available processing or transmission rate to the minimum between the available rates before and after reconfiguration. Importantly, a throughput-optimal policy for this worst-case reconfiguration delay model will guarantee throughput optimality for any other less restrictive model. Improved policies (in terms of cost-delay tradeoff) for this setting are of interest for future work.

5.6.2 A Heuristic Variant of ADCNC

In the previous subsection, we discussed a more generic setting of the cloud network control problem, where different reconfiguration overheads are associated with different reconfiguration operations, i.e., resource reconfiguration and commodity reconfiguration. While, as mentioned earlier, the throughput optimality guarantee for Adaptive DCNC extends to this case, it is possible to improve the performance, i.e., the cost-delay tradeoff, by exploiting the unequal reconfiguration overhead associated with the different operations. We now introduce a heuristic variant of Adaptive DCNC as an example for improving the cost-delay performance.

Recall that Adaptive DCNC reconfigures both resource allocation and scheduled commodity at the same time. This approach is reasonable when the reconfiguration overhead is the same for both operations, since when one reconfiguration operation is performed, the other could be performed at the same time without incurring additional overhead. However, when the reconfiguration overhead is different for different operations, intuitively one may benefit from performing the reconfiguration operation with smaller overhead more frequently. For this reason, we modify the reconfiguration criterion in Adaptive DCNC to a two-stage criterion. The first stage is the same as the reconfiguration criterion in Adaptive DCNC, while the additional stage is used to decide whether to perform the reconfiguration operation with smaller overhead. With the additional stage of reconfiguration criterion, we would expect the reconfiguration operation with smaller overhead to be performed more frequently.

To be more specific, consider an example where the resource reconfiguration overhead is larger than the commodity reconfiguration overhead. The local processing decisions at node $i \in \mathcal{V}$ at time t first follow steps 1) and 2) as described in section IV.A to compute $W_i^*(t)$ and $\Delta W_i(t)$. Then, at step 3), node i first checks if the criterion $\Delta W_i(t) > \theta_i(t)$ is met. If so, then it reconfigures both resource and commodity allocation; otherwise, it further checks the following. Node i computes $\Delta Q_i(t) = [Q_i^{(c^*)}(t) - Q_i^{(c^{*+})}(t)]^+ - [Q_i^{(\bar{c})}(t) - Q_i^{(\bar{c}+)}(t)]^+$, and a threshold defined as $\phi_i(t) = g([Q_i^{(c^*)}(t) - Q_i^{(c^{*+})}(t)]^+)$. If $\Delta Q_i(t) > \phi_i(t)$, then it reconfigures the commodity (while the resource allocation remains the same), otherwise no reconfiguration is performed. We refer to this policy as ADCNC-2stage, as it is a variant of ADCNC where the reconfiguration criterion becomes a 2-stage decision.

In Fig. 5.9, we show the simulation result for ADCNC and ADCNC-2stage under different commodity reconfiguration delay, while the resource reconfiguration delay is fixed as $\delta_{r,resource} = 20$. Again, for simplicity, we set all the reconfiguration costs to zero. We note that while the performance of ADCNC (solid lines) improves slightly as the commodity reconfiguration delay decreases, ADCNC-2stage (dashed lines) further exploits the smaller commodity reconfiguration delay and improves the cost-delay performance for each commodity reconfiguration delay.

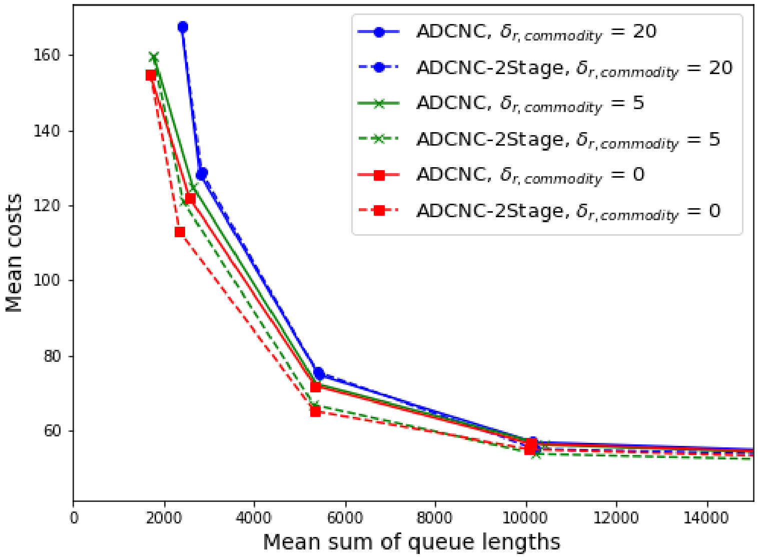

Figure 5.9: Mean cost versus mean queue length for ADCNC and ADCNC-2stage under various commodity reconfiguration delay. Resource reconfiguration delay is fixed as $\delta_{r,resource} = 20$. Arrival rate $\lambda = 2.5$ Gbps.

5.7 Concluding Remarks

This chapter addresses the dynamic control of network service chains in distributed computing networks (e.g., cloud networks) with non-negligible resource reconfiguration delay and cost. We show that while the capacity region and the minimum achievable time average cost remains unchanged regardless of the value of reconfiguration delay or cost, the throughput and cost optimality of existing policies (in the regime without reconfiguration delay/cost) is compromised when reconfiguration delay/cost exists. We then propose Adaptive DCNC, a distributed flow scheduling and resource allocation policy that addresses the reconfiguration overhead. In particular, we show that ADCNC is throughput optimal and achieves a $[O(1/V), O(V)]$ cost-delay tradeoff through an extension of the drift-plus-penalty analysis, and further validate the results via numerical simulations.

ADCNC exhibits important practical features, among which we highlight its distributed nature and the fact that it does not require neither prior knowledge of the traffic demands nor the exact values of reconfiguration overhead. These properties benefit the applicability of ADCNC to the control of large scale cloud networks and relieves the burden of demand estimation/prediction or the measurement of reconfiguration overhead. In addition, we show that ADNC can be easily extended to account for the heterogeneity of reconfiguration delay and cost to further improve the cost-delay performance in such practical settings.

With the recent rapid advances in network virtualization technologies, we expect the cloud network architecture considered in this work to be widely adopted for emerging large scale applications. The model and the control policy described in this chapter address two important practical aspects regarding the deployment and efficient operation of distributed cloud networks, namely the dynamic traffic demand and the reconfiguration overhead. We claim that awareness of reconfiguration overhead is essential toward accurately evaluating the performance of dynamic cloud network control policies, which should further facilitate the adoption of cloud network architectures in practice.

Chapter 5, in full, is a reprint of the material as it appears in the paper: C.-H. Wang, J. Llorca, A. M. Tulino, T. Javidi, "Dynamic Cloud Network Control under Reconfiguration Delay and Cost", IEEE/ACM Transactions on Networking (TON), vol 27, pp 491-504, 2019. The dissertation author was the primary investigator and author of this paper.

5.8 Supplementary Proofs

In the following, given a time t', we denote by $(k_{ij}(t'), c_{ij}(t'))$ the transmission resource-commodity pair scheduled at time t' on link (i,j), and by $(k_i(t'), c_i(t'))$ the processing resource-commodity pair scheduled at time t' at node i. In addition, we denote by $k^*_{ij}(t'), c^*_{ij}(t'))$ the transmission resource-commodity pair that maximizes the weight, $W^*_{ij}(t')$ at time t' on link (i,j), and by $(k^*_i(t'), c^*_i(t'))$ the processing resource-commodity pair that maximizes the weight, $W^*_i(t')$, at time t' at node i.

5.8.1 Proof of Lemma 5.1

Proof. We prove the result by contradiction. Suppose that under the assumption of Lemma 5.1, there are two or more reconfigurations within the time period $[t, t+T]$. Therefore we may select two consecutive reconfiguration instances t_1, t_2 with $t \leq t_1 < t_2 < t+T$.

Before we proceed with the proof, we state the following lemmas that will be handy in the following. The proofs of these lemmas are given in Appendix C.

Lemma 5.2. *Given Assumption 5.2, for any commodities $c_1, c_2 \in \mathcal{C}$, any nodes $i, j \in \mathcal{V}$, and any $\xi < \xi_{\max}$, $\rho > \rho_{\min}$, the (weighted) queue length differential between $Q_i^{(c_1)}$ and $Q_j^{(c_2)}$ can change only by a finite amount over one time slot, given as*

$$\begin{aligned}
&\Big|\frac{1}{\rho}\Big(Q_i^{(c_1)}(\tau+1) - \xi Q_j^{(c_2)}(\tau+1)\Big) - \frac{1}{\rho}\Big(Q_i^{(c_1)}(\tau) - \xi Q_j^{(c_2)}(\tau)\Big)\Big| \\
\leq &\frac{1}{\rho_{\min}}(1+\xi_{\max})\big(a_{\max} + C_{\max}(v_{\max}+1)\big) \triangleq \gamma_{\max} \qquad (5.19)
\end{aligned}$$

where $C_{\max} = \max\{\max\limits_{(i,j)\in\mathcal{E}} C_{ij}(K_{ij}), \max\limits_{i\in\mathcal{V}} C_i(K_i)\}$ is the maximum transmission or processing rate, and $v_{\max} = \max\limits_i\{\max\{|\mathcal{V}^+(i)|, |\mathcal{V}^-(i)|\}\}$ is the maximum number of incoming or outgoing links over all nodes. We also take $\rho_{\min} = \min\{1, \min\limits_{i,c} \rho_i^{(c)}\}$ and $\xi_{\max} = \max\{1, \max\limits_c \xi^{(c)}\}$.

Similarly, the change in the maximal queue length differential for transmission on link

(i,j) over one time slot is bounded as

$$\left|\max_{c\in\mathcal{C}}\left\{Q_i^{(c)}(\tau+1)-Q_j^{(c)}(\tau+1)\right\}-\max_{c\in\mathcal{C}}\left\{Q_i^{(c)}(\tau)-Q_j^{(c)}(\tau)\right\}\right|\leq\gamma_{\max}\tag{5.20}$$

and the change in the maximal queue length differential for processing in node i over one time slot is bounded as

$$\begin{aligned}\Bigg|\max_{c\in\mathcal{C}}&\left\{\frac{1}{\rho_i^{(c)}}\big[Q_i^{(c)}(\tau+1)-\xi^{(c^+)}Q_i^{(c^+)}(\tau+1)-Ve_i\big]^+\right\}\\&-\max_{c\in\mathcal{C}}\left\{\frac{1}{\rho_i^{(c)}}\big[Q_i^{(c)}(\tau)-\xi^{(c^+)}Q_i^{(c^+)}(\tau)-Ve_i\big]^+\right\}\Bigg|\leq\gamma_{\max}\end{aligned}\tag{5.21}$$

Lemma 5.3. *Given Assumption 5.1, and any fixed $V<\infty$, define $F(x)\triangleq\max_{k\leq K_{ij}}\big\{C_{ij}(k)[x-Ve_{ij}]^+-Vw_{ij}(k)\big\}$. Then,*

(a) $F(x)$ is Lipschitz continuous with Lipschitz constant $C_{ij,\max}=\max_{k\leq K_{ij}}C_{ij}(k)$.

(b) If $x>V\big(\min_{k>0}\frac{w_{ij}(k)}{C_{ij}(k)}+e_{ij}\big)$, then $F(x)>0$ and the k^* maximizing $F(x)$ satisfies $k^*>0$; otherwise, $F(x)=0$ and the maximizer is $k^*=0$.

In the following, we are going to show that under the assumption of Lemma 5.1, $\max_c\left\{Q_i^{(c)}(t)-Q_j^{(c)}(t)\right\}>M_{ij}$, the weight difference at time t_2 can not exceed the threshold $g\Big(C_{ij}(k_{ij}(t_2-1))\max_{c\in\mathcal{C}}\left\{Q_i^{(c)}(t_2)-Q_j^{(c)}(t_2)\right\}\Big)$. This, hence, contradicts the assumption that Adptive DCNC reconfigures at time t_2.

To do this, starting from (5.6) we rewrite the transmission weight differential as:

$$\begin{aligned}\Delta W_{ij}(t_2)=&W_{ij}^*(t_2)-W_{ij}(t_2)\\=&(W_{ij}^*(t_2)-W_{ij}^*(t_1))+(W_{ij}^*(t_1)-W_{ij}(t_2))\end{aligned}\tag{5.22}$$

with

$$\begin{aligned} W_{ij}^*(t_2) =& F\Big(\max_{c\in\mathcal{C}} \Big\{ Q_i^{(c)}(t_2) - Q_j^{(c)}(t_2) \Big\} \Big) \\ =& C_{ij}(k_{ij}^*(t_2)) \Big[Q_i^{(c_{ij}^*(t_2))}(t_2) - Q_j^{(c_{ij}^*(t_2))}(t_2) - Ve_{ij} \Big]^+ - Vw_{ij}(k_{ij}^*(t_2)) \end{aligned} \tag{5.23}$$

$$\begin{aligned} W_{ij}^*(t_1) =& F\Big(\max_{c\in\mathcal{C}} \Big\{ Q_i^{(c)}(t_1) - Q_j^{(c)}(t_1) \Big\} \Big) \\ =& C_{ij}(k_{ij}^*(t_1)) \Big[Q_i^{(c_{ij}^*(t_1))}(t_1) - Q_j^{(c_{ij}^*(t_1))}(t_1) - Ve_{ij} \Big]^+ - Vw_{ij}(k_{ij}^*(t_1)) \end{aligned} \tag{5.24}$$

and

$$\begin{aligned} W_{ij}(t_2) =& C_{ij}(\bar{k}_{ij}) \Big[Q_i^{(\bar{c}_{ij})}(t_2) - Q_j^{(\bar{c}_{ij})}(t_2) - Ve_{ij} \Big]^+ - Vw_{ij}(\bar{k}_{ij}) \\ =& C_{ij}(k_{ij}^*(t_1)) \Big[Q_i^{(c_{ij}^*(t_1))}(t_2) - Q_j^{(c_{ij}^*(t_1))}(t_2) - Ve_{ij} \Big]^+ - Vw_{ij}(k_{ij}^*(t_1)) \end{aligned} \tag{5.25}$$

where in (5.25) we have used the fact that, given the assumption that Adaptive DCNC reconfigures at time slots t_1, t_2 , during $[t_1, t_2 - 1]$ the resource allocation and the transmitted commodity remains $k_{ij}^*(t_1)$ and $c_{ij}^*(t_1)$, respectively.

Now, using Lemma 5.3 (a), we have

$$\begin{aligned} W_{ij}^*(t_2) - W_{ij}^*(t_1) =& F\Big(\max_{c\in\mathcal{C}} \Big\{ Q_i^{(c)}(t_2) - Q_j^{(c)}(t_2) \Big\} \Big) - F\Big(\max_{c\in\mathcal{C}} \Big\{ Q_i^{(c)}(t_1) - Q_j^{(c)}(t_1) \Big\} \Big) \\ \le& C_{ij,\max} \Big| \max_{c\in\mathcal{C}} \Big\{ Q_i^{(c)}(t_2) - Q_j^{(c)}(t_2) \Big\} - \max_{c\in\mathcal{C}} \Big\{ Q_i^{(c)}(t_1) - Q_j^{(c)}(t_1) \Big\} \Big| \\ \le& C_{ij,\max}(t_2 - t_1)\gamma_{\max} \\ \le& C_{ij,\max} T \gamma_{\max} \end{aligned} \tag{5.26}$$

On the other hand we have that:

$$\begin{aligned}W^*_{ij}(t_1)-W_{ij}(t_2)=&C_{ij}(k^*_{ij}(t_1))[Q_i^{(c^*_{ij}(t_1))}(t_1)-Q_j^{(c^*_{ij}(t_1))}(t_1)-Ve_{ij}]^+\\&-C_{ij}(k^*_{ij}(t_1))[Q_i^{(c^*_{ij}(t_1))}(t_2)-Q_j^{(c^*_{ij}(t_1))}(t_2)-Ve_{ij}]^+\\\leq&C_{ij}(k^*_{ij}(t_1))\Big|\Big(Q_i^{(c^*_{ij}(t_1))}(t_2)-Q_j^{(c^*_{ij}(t_1))}(t_2)\Big)\\&\qquad-\Big(Q_i^{(c^*_{ij}(t_1))}(t_1)-Q_j^{(c^*_{ij}(t_1))}(t_1)\Big)\Big|\\\leq&C_{ij,\max}(t_2-t_1)\gamma_{\max}\\\leq&C_{ij,\max}T\gamma_{\max}\end{aligned}\tag{5.27}$$

where the last inequality follows from Lemma 5.2.

Combining (5.26) and (5.27), we have

$$W^*_{ij}(t_2)-W_{ij}(t_2)\leq 2C_{ij,\max}T\gamma_{\max}\tag{5.28}$$

From the assumption of Lemma 5.1, and using Lemma 5.2, we have $\max\limits_{c\in\mathcal{C}}\big\{Q_i^{(c)}(t_1)-Q_j^{(c)}(t_1)\big\}>V\big(\min\limits_{k>0}\frac{w_{ij}(k)}{C_{ij}(k)}+e_{ij}\big)$, and hence $k^*_{ij}(t_1)>0$ following Lemma 5.3 (b). Similarly, with the assumption of Lemma 5.1, and using Lemma 5.2, we also have $\max\limits_{c\in\mathcal{C}}\big\{Q_i^{(c)}(t_2)-Q_j^{(c)}(t_2)\big\}>\frac{1}{C_{ij}(1)}g^{-1}\big(2C_{ij,\max}T\gamma_{\max}\big)$, from which it follows that

$$\begin{aligned}g\Big(C_{ij}(k^*_{ij}(t_1))\max_{c\in\mathcal{C}}\Big\{Q_i^{(c)}(t_2)-Q_j^{(c)}(t_2)\Big\}\Big)>&2C_{ij,\max}T\gamma_{\max}\\\geq&W^*_{ij}(t_2)-W_{ij}(t_2)\end{aligned}\tag{5.29}$$

which contradics the assumption that Adaptive DCNC reconfigures at time t_2.

We now consider the condition for the processing decision at node i. From the assumption of Lemma 5.1, $\max\limits_{c}\Big\{Q_i^{(c)}(t)-\xi^{(c^+)}Q_i^{(c^+)}(t)\Big\}>M_i$, the weight difference at time t_2 can not exceed the threshold $g\Big(C_{ij}(k_{ij}(t_2-1))\max\limits_{c\in\mathcal{C}}\Big\{Q_i^{(c)}(t_2)-Q_j^{(c)}(t_2)\Big\}\Big)$.

To do this, we rewrite the processing weight differential as:

$$\begin{aligned}\Delta W_i(t_2) =& W_i^*(t_2) - W_i(t_2) \\ =&(W_i^*(t_2) - W_i^*(t_1)) + (W_i^*(t_1) - W_i(t_2))\end{aligned} \tag{5.30}$$

with

$$W_i^*(t_2) = \frac{C_i(k_i^*(t_2))}{\rho_i^{(c_i^*(t_2))}} \Big[Q_i^{(c_i^*(t_2))}(t_2) - \xi^{({c_i^*}^+(t_2))} Q_i^{({c_i^*}^+(t_2))}(t_2) - Ve_i \Big]^+ - Vw_i(k_i^*(t_2)) \tag{5.31}$$

$$W_i^*(t_1) = \frac{C_i(k_i^*(t_1))}{\rho_i^{(c_i^*(t_1))}} \Big[Q_i^{(c_i^*(t_1))}(t_1) - \xi^{({c_i^*}^+(t_1))} Q_i^{({c_i^*}^+(t_1))}(t_1) - Ve_i \Big]^+ - Vw_i(k_i^*(t_1)) \tag{5.32}$$

and

$$W_i(t_2) = \frac{C_i(k_i^*(t_1))}{\rho_i^{({c_i^*}^+(t_1))}} \Big[Q_i^{(c_i^*(t_1))}(t_2) - \xi^{({c_i^*}^+(t_1))} Q_i^{({c_i^*}^+(t_1))}(t_2) - Ve_i \Big]^+ - Vw_i(k_i^*(t_1)) \tag{5.33}$$

where in (5.33) we have used the fact that, given the assumption that Adaptive DCNC reconfigures at time slots t_1, t_2, during $[t_1, t_2 - 1]$ the resource allocation and the commodity being processed remains $k_i^*(t_1), c_i^*(t_1)$, respectively.

Now, using Lemma 5.3 (a), we have

$$\begin{aligned}
W_i^*(t_2) - W_i^*(t_1) =& F\Big(\max_{c\in\mathcal{C}} \Big\{ \frac{1}{\rho_i^{(c)}} [Q_i^{(c)}(t_2) - Q_j^{(c)}(t_2) - Ve_i]^+ \Big\}\Big) \\
& - F\Big(\max_{c\in\mathcal{C}} \Big\{ \frac{1}{\rho_i^{(c)}} [Q_i^{(c)}(t_1) - Q_j^{(c)}(t_1) - Ve_i]^+ \Big\}\Big) \\
\leq & C_{ij,\max} \Big| \max_{c\in\mathcal{C}} \Big\{ \frac{1}{\rho_i^{(c)}} [Q_i^{(c)}(t_2) - Q_j^{(c)}(t_2) - Ve_i]^+ \Big\} \\
& - \max_{c\in\mathcal{C}} \Big\{ \frac{1}{\rho_i^{(c)}} [Q_i^{(c)}(t_1) - Q_j^{(c)}(t_1) - Ve_i]^+ \Big\} \Big| \\
\leq & C_{ij,\max}(t_2 - t_1)\gamma_{\max} \\
\leq & C_{ij,\max} T \gamma_{\max} \qquad (5.34)
\end{aligned}$$

□

5.8.2 Proof of Theorem 5.2

Proof. Consider the quadratic Lyapunov function $L : \mathbb{R}^{N\times N} \to \mathbb{R}$ where $L(\mathbf{Q}) = \frac{1}{2}\sum_{i,c}(Q_i^{(c)})^2$. Let $\mathbf{X}(t) = \Big(\mathbf{Q}(t), \mathbf{k}(t), \boldsymbol{\mu}(t), \mathbf{r}(t)\Big)$ denote the queue length, resource allocation decision, flow rate decision, and the reconfiguration status of the cloud network at time t.

We first consider zero reconfiguration costs $\eta_i = 0, \eta_{ij} = 0$.

We now leverage Lemma 5.1 to bound the T-step Lyapunov drift-plus-penalty.

$$\begin{aligned}
&\mathbb{E}\Big[L(\mathbf{Q}(t+T)) - L(\mathbf{Q}(t)) + V\sum_{\tau=t}^{t+T} h(\tau) \Big| \mathbf{X}(t) \Big] \\
=&\mathbb{E}\Big[\sum_{\tau=t}^{t+T-1} \mathbb{E}\left[L(\mathbf{Q}(\tau+1)) - L(\mathbf{Q}(\tau)) + Vh(\tau)|\mathbf{X}(\tau)\right] \Big| \mathbf{X}(t) \Big] \qquad (5.35)
\end{aligned}$$

For each time slot $\tau \in [t, t+T]$ we have:

$$\begin{aligned}&\mathbb{E}\left[L(\mathbf{Q}(\tau+1)) - L(\mathbf{Q}(\tau)) + Vh(\tau)\middle|\mathbf{X}(\tau)\right]\\ \le&\Phi + \sum_{i,c} Q_i^{(c)}(\tau)\lambda_i^{(c)} - \mathbb{E}\Big[Z(\tau)\Big|\mathbf{X}(\tau)\Big] + V\mathbb{E}\Big[h(\tau)\Big|\mathbf{X}(\tau)\Big] \qquad (5.36)\end{aligned}$$

where

$$\begin{aligned}\Phi &= \frac{1}{2}N\Big[((v_{\max}+1)C_{\max})^2 + ((v_{\max}+1)C_{\max}+a_{\max})^2\Big]\\ Z(\tau) &= \sum_{i,c} Q_i^{(c)}(\tau)\Big[\sum_{j\in\delta(i)} \mu_{ij}^{(c)}(\tau)\mathbb{1}_{\{r_{ij}(\tau)=0\}} + \mu_i^{(c)}(\tau)\mathbb{1}_{\{r_i(\tau)=0\}}\\ &\qquad - \sum_{j:i\in\delta(j)} \mu_{ji}^{(c)}(\tau)\mathbb{1}_{\{r_{ji}(\tau)=0\}} - \mu_i^{(c^-)}(\tau)\mathbb{1}_{\{r_i(\tau)=0\}}\Big] \qquad (5.37)\end{aligned}$$

and $v_{\max} = \max_{i\in\mathcal{V}}\{\max\{|\mathcal{V}^+(i)|, |\mathcal{V}^-(i)|\}\}$.

Let $W^*(\tau)$ be the sum of transmission max-utility-weights and the processing max-utility-weights:

$$W^*(\tau) = \sum_{(i,j)\in\mathcal{E}} W_{ij}^*(\tau) + \sum_{i\in\mathcal{V}} W_i^*(\tau), \qquad (5.38)$$

where the transmission max-utility-weights and the processing max-utility-weights are given by

$$\begin{aligned}W_{ij}^*(\tau) =&\mu_{ij}^{*^{(c_{ij}^*(\tau)}})(\tau)\big(Q_i^{(c_{ij}^*(\tau)}(\tau) - Q_j^{(c_{ij}^*(\tau))}(\tau) - Ve_{ij}\big) - Vw_{ij}(k_{ij}^*(\tau))\\ W_i^*(\tau) =&\mu_i^{*^{(c_i^*(\tau))}}(\tau)\big(Q_i^{(c_i^*(\tau))}(\tau) - Q_i^{(c_i^{*^+}(\tau))}(\tau) - Ve_i\big) - Vw_i(k_i^*(\tau)) \qquad (5.39)\end{aligned}$$

In (5.36), adding and subtracting $\mathbb{E}\Big[W^*(\tau)\Big|\mathbf{X}(\tau)\Big]$ given in (5.38), and recalling that for any $\epsilon > 0$ such that $\boldsymbol{\lambda} + \epsilon \mathbf{1} \in \boldsymbol{\Lambda}$, (see [16, 17] for derivation):

$$\sum_{i,c} Q_i^{(c)}(\tau)\lambda_i^{(c)} - \mathbb{E}\Big[W^*(\tau)\Big|\mathbf{X}(\tau)\Big] \leq -\epsilon \sum_{i,c} Q_i^{(c)}(\tau) + Vh^*(\boldsymbol{\lambda} + \epsilon\mathbf{1}), \tag{5.40}$$

we can further bound the drift-plus-penalty in (5.36) as:

$$\begin{aligned} &\mathbb{E}\left[L(\mathbf{Q}(\tau+1)) - L(\mathbf{Q}(\tau)) + Vh(\tau)\middle|\mathbf{X}(\tau)\right] \\ \leq &\Phi - \epsilon \sum_{i,c} Q_i^{(c)}(\tau) + Vh^*(\boldsymbol{\lambda} + \epsilon\mathbf{1}) \\ &+ \mathbb{E}\Big[W^*(\tau)\Big|\mathbf{X}(\tau)\Big] - \mathbb{E}\Big[Z(\tau)\Big|\mathbf{X}(\tau)\Big] + V\mathbb{E}\Big[h(\tau)\Big|\mathbf{X}(\tau)\Big] \end{aligned} \tag{5.41}$$

From above we have a negative term in the drift-plus-penalty which decreases as the total queue length increases. It then remains to ensure that $W^*(\tau) - Z(\tau) + Vh(\tau)$ could be bounded so that the we can still bound the drift-plus-penalty with a term that decrease when the total queue length increases. To this end notice that:

$$\begin{aligned} W^*(\tau) - Z(\tau) + Vh(\tau) = &\sum_{(i,j)\in\mathcal{E}} \left[W_{ij}^*(\tau) - W_{ij}(\tau)\mathbb{1}_{\{r_{ij}(\tau)=0\}}\right] \\ &+ \sum_{i\in\mathcal{V}} \left[W_i^*(\tau) - W_i(\tau)\mathbb{1}_{\{r_i(\tau)=0\}}\right] \end{aligned} \tag{5.42}$$

where

$$\begin{aligned} W_{ij}(\tau) =&\mu_{ij}^{(c(\tau))}(\tau)\big(Q_i^{(c(\tau))}(\tau) - Q_j^{(c(\tau))}(\tau) - Ve_{ij}\big) - Vw_{ij}(k_{ij}(\tau)) \\ W_i(\tau) =&\mu_i^{(c(\tau))}(\tau)\big(Q_i^{(c(\tau))}(\tau) - Q_i^{(c^+(\tau))}(\tau) - Ve_i\big) - Vw_i(k_i(\tau)) \end{aligned}$$

For the above expression, we now bound the term for each node i and each link (i, j) separately. We do this with the help of Lemma 5.1.

We start with the term for link (i,j). Note that for any given time $\tau \in [t, t+T]$ such that Adaptive DCNC does not reconfigure (i.e $r_{ij}(\tau) = 0$), by construction we have that the transmission weight differential is bounded by:

$$\begin{aligned} W^*_{ij}(\tau) - W_{ij}(\tau) \leq & g\bigg(C_{ij,\max} \max_c \Big\{Q_i^{(c)}(\tau) - Q_j^{(c)}(\tau)\Big\}\bigg) \\ \leq & g\bigg(C_{\max} \max_{i,c} \Big\{Q_i^{(c)}(\tau)\Big\}\bigg). \end{aligned} \tag{5.43}$$

Alternative for any given time $\tau \in [t, t+T]$ such that $r_{ij}(\tau) > 0$ i.e. Adaptive DCNC is under reconfiguration, since the transmission weight differential is always bounded by the transmission max-utility-weight, than it suffices to bound $W^*_{ij}(\tau)$. In particular, if link (i,j) at time t satisfies $\max_c\{Q_i^{(c)}(t) - Q_j^{(c)}(t)\} > M_{ij}$, then at time τ the transmission weight differential can be bound as follow:

$$\begin{aligned} W^*_{ij}(\tau) - W_{ij}(\tau) &\leq W^*_{ij}(\tau) \\ &\leq C_{ij,\max} \max_c \Big\{Q_i^{(c)}(\tau) - Q_j^{(c)}(\tau)\Big\} \\ &\leq C_{\max} \max_{i,c} \Big\{Q_i^{(c)}(\tau)\Big\} \end{aligned} \tag{5.44}$$

while for the case that at time t, $\max_c\{Q_i^{(c)}(t) - Q_j^{(c)}(t)\} \leq M_{ij}$, starting from the expression of the transmission max-utility-weight $W^*_{ij}(\tau)$, we observe that since M_{ij} is the max of two terms one of the following could hold:

1. $\max_c\{Q_i^{(c)}(t) - Q_j^{(c)}(t)\} \leq V\Big(\min_k \frac{w_{ij}(k)}{C_{ij}(k)} + e_{ij}\Big) + T\gamma_{\max}$.

   In this case using Lemma 5.2, we have that at time τ

$$\max_c\{Q_i^{(c)}(\tau) - Q_j^{(c)}(\tau)\} \leq V\Big(\min_k \frac{w_{ij}(k)}{C_{ij}(k)} + e_{ij}\Big) + 2T\gamma_{\max} \tag{5.45}$$

from which it follows that

$$\begin{aligned} W_{ij}^*(\tau) \leq & \max_k \Big\{ C_{ij}(k) \big[V \min_k \frac{w_{ij}(k)}{C_{ij}(k)} + 2T\gamma_{\max} \big] - V w_{ij}(k) \Big\} \\ \leq & \max_k \Big\{ C_{ij}(k) V \min_k \frac{w_{ij}(k)}{C_{ij}(k)} - V w_{ij}(k) \Big\} + 2C_{ij,\max} T\gamma_{\max} \\ \leq & 2C_{ij,\max} T\gamma_{\max} \end{aligned} \tag{5.46}$$

where the last inequality follows from Lemma 5.3 (b).

2. $\max_c \{Q_i^{(c)}(t) - Q_j^{(c)}(t)\} \leq \frac{1}{C_{ij}(1)} g^{-1}\Big(2C_{ij,\max} T\gamma_{\max}\Big) + T\gamma_{\max}$:
   In this case using Lemma 5.2, we have that at time τ

$$\max_c \{Q_i^{(c)}(\tau) - Q_j^{(c)}(\tau)\} \leq \frac{1}{C_{ij}(1)} g^{-1}\Big(2C_{ij,\max} T\gamma_{\max}\Big) + 2T\gamma_{\max} \tag{5.47}$$

   and thus

$$\begin{aligned} W_{ij}^*(\tau) \leq & C_{ij,\max} \max_c \Big\{ Q_i^{(c)}(\tau) - Q_j^{(c)}(\tau) \Big\} \\ \leq & \frac{C_{ij,\max}}{C_{ij}(1)} g^{-1}\Big(2C_{ij,\max} T\gamma_{\max}\Big) + 2C_{ij,\max} T\gamma_{\max} \end{aligned}$$

Combining the two cases, we have for $\max_c \{Q_i^{(c)}(t) - Q_j^{(c)}(t)\} \leq M_{ij}$ that

$$\begin{aligned} W_{ij}^*(\tau) \leq & \frac{C_{ij,\max}}{C_{ij}(1)} g^{-1}\Big(2C_{ij,\max} T\gamma_{\max}\Big) + 2C_{ij,\max} T\gamma_{\max} \\ \triangleq & \ \Phi_{ij} \end{aligned} \tag{5.48}$$

We then use the same approach to give a bound on the term for each node i. Applying

(5.36), (5.40), (5.42), and (5.43)-(5.48) into (5.35), we have

$$
\begin{aligned}
&\mathbb{E}\left[L(\mathbf{Q}(t+T))-L(\mathbf{Q}(t))+V\sum_{\tau=t}^{t+T}h(\tau)|\mathbf{X}(t)\right]\\
\leq&\mathbb{E}\bigg[T\Phi-\epsilon\sum_{\tau=t}^{t+T-1}\sum_{i,c}Q_i^{(c)}(\tau)+TVh^*(\boldsymbol{\lambda}+\epsilon\mathbf{1})\\
&+\sum_{\tau=t}^{t+T}\bigg(\sum_{(i,j)\in\mathcal{E}}g\Big(C_{\max}\max_{i,c}\Big\{Q_i^{(c)}(\tau)\Big\}\Big)\mathbb{1}_{\{r_{ij}(\tau)=0\}}\\
&+\sum_{\substack{(i,j)\in\mathcal{E}:\\ \max\limits_{c}\{Q_i^{(c)}(t)-Q_j^{(c)}(t)\}>M_{ij}}}C_{\max}\max_{i,c}\Big\{Q_i^{(c)}(\tau)\Big\}\mathbb{1}_{\{r_{ij}(\tau)>0\}}\\
&+\sum_{\substack{(i,j)\in\mathcal{E}:\\ \max\limits_{c}\{Q_i^{(c)}(t)-Q_j^{(c)}(t)\}\leq M_{ij}}}\Phi_{ij}\mathbb{1}_{\{r_{ij}(\tau)>0\}}\bigg)\\
&+\sum_{\tau=t}^{t+T}\bigg(\sum_{i\in\mathcal{V}}g\Big(C_{\max}\max_{i,c}\Big\{Q_i^{(c)}(\tau)\Big\}\Big)\mathbb{1}_{\{r_i(\tau)=0\}}\\
&+\sum_{\substack{i\in\mathcal{V}:\\ \max\limits_{c}\{Q_i^{(c)}(t)-Q_i^{(c^+)}(t)\}>M_i}}C_{\max}\max_{i,c}\Big\{Q_i^{(c)}(\tau)\Big\}\mathbb{1}_{\{r_i(\tau)>0\}}\\
&+\sum_{\substack{i\in\mathcal{V}:\\ \max\limits_{c}\{Q_i^{(c)}(t)-Q_i^{(c^+)}(t)\}\leq M_i}}\Phi_i\mathbb{1}_{\{r_i(\tau)>0\}}\bigg)\bigg|\mathbf{X}(t)\bigg]\\
\leq&\mathbb{E}\bigg[T\Phi-\epsilon\sum_{\tau=t}^{t+T-1}\sum_{i,c}Q_i^{(c)}(\tau)+TVh^*(\boldsymbol{\lambda}+\epsilon\mathbf{1})\\
&+\sum_{(i,j)\in\mathcal{E}}\Big[\sum_{\tau=t}^{t+T}g\Big(C_{\max}\max_{i,c}\big\{Q_i^{(c)}(\tau)\big\}\Big)+\delta_{ij}C_{\max}\Big(\max_{i,c}\big\{Q_i^{(c)}(t)\big\}+T\gamma_{\max}\Big)+T\Phi_{ij}\Big]\\
&+\sum_{i\in\mathcal{V}}\Big[\sum_{\tau=t}^{t+T}g\Big(C_{\max}\max_{i,c}\big\{Q_i^{(c)}(\tau)\big\}\Big)+\delta_iC_{\max}\Big(\max_{i,c}\big\{Q_i^{(c)}(t)\big\}+T\gamma_{\max}\Big)+T\Phi_i\Big]\bigg|\mathbf{X}(t)\bigg]
\end{aligned}
\tag{5.49}
$$

Let $\Phi' = \Phi + \sum\limits_{(i,j)\in\mathcal{E}}(\Phi_{ij} + \delta_{ij}C_{\max}T\gamma_{\max}) + \sum\limits_{i\in\mathcal{V}}(\Phi_i + \delta_i C_{\max}T\gamma_{\max})$, we then have

$$
\begin{aligned}
&\mathbb{E}\left[L(\mathbf{Q}(t+T)) - L(\mathbf{Q}(t)) + V\sum_{\tau=t}^{t+T} h(\tau)|\mathbf{X}(t)\right] \\
\leq &\mathbb{E}\Bigg[T\Phi' - \epsilon\sum_{\tau=t}^{t+T-1}\sum_{i,c} Q_i^{(c)}(\tau) + TVh^*(\boldsymbol{\lambda}+\epsilon\mathbf{1}) \\
&+ \sum_{(i,j)\in\mathcal{E}}\Big[\sum_{\tau=t}^{t+T} g\Big(C_{\max}\max_{i,c}\big\{Q_i^{(c)}(\tau)\big\}\Big) + \delta_{ij}C_{\max}\max_{i,c}\big\{Q_i^{(c)}(t)\big\}\Big] \\
&+ \sum_{i\in\mathcal{V}}\Big[\sum_{\tau=t}^{t+T} g\Big(C_{\max}\max_{i,c}\big\{Q_i^{(c)}(\tau)\big\}\Big) + \delta_i C_{\max}\max_{i,c}\big\{Q_i^{(c)}(t)\big\}\Big]\Bigg|\mathbf{X}(t)\Bigg]
\end{aligned}
\tag{5.50}
$$

Now select $T > \max\{\frac{8|\mathcal{E}|\delta_{ij}C_{ij,\max}}{\epsilon}, \frac{8|\mathcal{V}|\delta_i C_{i,\max}}{\epsilon}\}$, and since g is a sublinear function, there exists a constant $K_T < \infty$ such that $x > K_T$ implies $|\mathcal{E}|g(C_{\max}x) < \frac{\epsilon}{8}x$ for any $(i,j) \in \mathcal{E}$ and $|\mathcal{V}|g(C_{\max}x) < \frac{\epsilon}{8}x$ for any node $i \in \mathcal{V}$.

We thus have

$$
\begin{aligned}
&\mathbb{E}\left[L(\mathbf{Q}(t+T)) - L(\mathbf{Q}(t)) + V\sum_{\tau=t}^{t+T} h(\tau)\middle|\mathbf{X}(t)\right] \\
\leq &\mathbb{E}\left[T\Phi'' - \frac{\epsilon}{2}\sum_{\tau=t}^{t+T-1}\sum_{i,c}\mathbb{E}[Q_i^{(c)}(\tau)] + TVh^*(\boldsymbol{\lambda}+\epsilon\mathbf{1})\middle|\mathbf{X}(t)\right]
\end{aligned}
\tag{5.51}
$$

Taking expectation on both sides and summing over time slots $t = 0, T, 2T, \ldots, (K-1)T$, we have

$$
\begin{aligned}
&\mathbb{E}[L(\mathbf{Q}(KT))] - \mathbb{E}[L(\mathbf{Q}(0))] + V\sum_{\tau=0}^{KT}\mathbb{E}[h(\tau)] \\
\leq &KT\Phi'' - \frac{\epsilon}{2}\sum_{\tau=0}^{KT}\sum_{i,c} Q_i^{(c)}(\tau) + KTVh^*(\boldsymbol{\lambda}+\epsilon\mathbf{1})
\end{aligned}
\tag{5.52}
$$

Further divide both sides by KT, and rearranging the terms, we have

$$\begin{aligned}&\frac{\epsilon}{2KT}\sum_{\tau=0}^{KT}\sum_{i,c}\mathbb{E}[Q_i^{(c)}(\tau)]+\frac{V}{KT}\sum_{\tau=0}^{KT}\mathbb{E}[h(\tau)]\\ \leq&\Phi''+Vh^*(\boldsymbol{\lambda}+\epsilon\mathbf{1})+\frac{1}{2KT}\mathbb{E}[\|\mathbf{Q}(0)\|^2]\end{aligned} \tag{5.53}$$

From (5.53), using [35, Prop. 6.1], Eqs. (5.17) and (5.18) in Theorem 5.2 follow. □

5.8.3 Proofs of Lemmas 5.2-5.3

Proof. (of lemma 5.2) First note that with assumption 5.2, for any queue $Q_i^{(c)}$, the queue length change over one time slot may be bounded as follows:

$$Q_i^{(c)}(\tau+1)-Q_i^{(c)}(\tau)\leq a_{\max}+C_{\max}(|\mathcal{V}^-(i)|+1) \tag{5.54}$$

and

$$Q_i^{(c)}(\tau+1)-Q_i^{(c)}(\tau)\geq -C_{\max}(|\mathcal{V}^+(i)|+1) \tag{5.55}$$

Hence we have

$$\left|Q_i^{(c)}(\tau+1)-Q_i^{(c)}(\tau)\right|\leq a_{\max}+C_{\max}(v_{\max}+1) \tag{5.56}$$

where $v_{\max}=\max_i\{\max\{|\mathcal{V}^+(i)|,|\mathcal{V}^-(i)|\}\}$.

By rearranging the terms and using the triangle inequality, we have

$$\begin{aligned}
&\Big|\frac{1}{\rho}\Big(Q_i^{(c_1)}(\tau+1)-\xi Q_j^{(c_2)}(\tau+1)\Big)-\frac{1}{\rho}\Big(Q_i^{(c_1)}(\tau)-\xi Q_j^{(c_2)}(\tau)\Big)\Big| \\
=&\Big|\frac{1}{\rho}\Big(Q_i^{(c_1)}(\tau+1)-Q_j^{(c_2)}(\tau+1)\Big)-\frac{\xi}{\rho}\Big(Q_i^{(c_1)}(\tau)-Q_j^{(c_2)}(\tau)\Big)\Big| \\
\leq&\frac{1}{\rho}\Big|Q_i^{(c_1)}(\tau+1)-Q_i^{(c_1)}(\tau)\Big|+\frac{\xi}{\rho}\Big|Q_j^{(c_2)}(\tau+1)-Q_j^{(c_2)}(\tau)\Big| \\
\leq&\frac{1}{\rho}(1+\xi)\Big(a_{\max}+C_{\max}(v_{\max}+1)\Big) \\
\leq&\frac{1}{\rho_{\min}}(1+\xi_{\max})\big(a_{\max}+C_{\max}(v_{\max}+1)\big)\triangleq\gamma_{\max} \qquad (5.57)
\end{aligned}$$

which establishes the first part.

Using (5.57), we further prove the second part of the lemma. Let $c_{ij}^*(t)$ denote the commodity that maximizes the queue length differential at any time t, i.e. $c_{ij}^*(t)=\arg\max_{c\in\mathcal{C}}\Big\{Q_i^{(c)}(\tau+1)-Q_j^{(c)}(\tau+1)\Big\}$, we then have

$$\begin{aligned}
\max_{c\in\mathcal{C}}\Big\{Q_i^{(c)}(\tau+1)-Q_j^{(c)}(\tau+1)\Big\}=&Q_i^{(c_{ij}^*(\tau+1))}(\tau+1)-Q_j^{(c_{ij}^*(\tau+1))}(\tau+1) \\
\leq&\Big(Q_i^{(c_{ij}^*(\tau+1))}(\tau)-Q_j^{(c_{ij}^*(\tau+1))}(\tau)\Big)+\gamma_{\max} \\
\leq&\max_{c\in\mathcal{C}}\Big\{Q_i^{(c)}(\tau)-Q_j^{(c)}(\tau)\Big\}+\gamma_{\max} \qquad (5.58)
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\max_{c\in\mathcal{C}}\Big\{Q_i^{(c)}(\tau+1)-Q_j^{(c)}(\tau+1)\Big\}\geq&Q_i^{(c_{ij}^*(\tau))}(\tau+1)-Q_j^{(c_{ij}^*(\tau))}(\tau+1) \\
\geq&\Big(Q_i^{(c_{ij}^*(\tau))}(\tau)-Q_j^{(c_{ij}^*(\tau))}(\tau)\Big)-\gamma_{\max} \\
=&\max_{c\in\mathcal{C}}\Big\{Q_i^{(c)}(\tau)-Q_j^{(c)}(\tau)\Big\}-\gamma_{\max} \qquad (5.59)
\end{aligned}$$

Combining the two bounds above, we have (5.20). Similarly, let $c_i^*(t)=$

$\arg\max\limits_{c\in\mathcal{C}}\left\{\frac{1}{\rho_i^{(c)}}\left[Q_i^{(c)}(t)-\xi^{(c^+)}Q_i^{(c^+)}(t)-Ve_i\right]^+\right\}$, we have

$$\begin{aligned}
&\max_{c\in\mathcal{C}}\left\{\frac{1}{\rho_i^{(c)}}\left[Q_i^{(c)}(\tau+1)-\xi^{(c^+)}Q_i^{(c^+)}(\tau+1)-Ve_i\right]^+\right\}\\
=&\frac{1}{\rho_i^{(c_i^*(\tau+1))}}\left[Q_i^{(c_i^*(\tau+1))}(\tau+1)-Q_j^{(c_i^*(\tau+1))}(\tau+1)-Ve_i\right]^+\\
\le&\frac{1}{\rho_i^{(c_i^*(\tau+1))}}\left[Q_i^{(c_i^*(\tau+1))}(\tau)-Q_j^{(c_i^*(\tau+1))}(\tau)-Ve_i\right]^++\gamma_{\max}\\
\le&\max_{c\in\mathcal{C}}\left\{\frac{1}{\rho_i^{(c)}}\left[Q_i^{(c)}(\tau)-\xi^{(c^+)}Q_i^{(c^+)}(\tau)-Ve_i\right]^+\right\}+\gamma_{\max}
\end{aligned}\tag{5.60}$$

and

$$\begin{aligned}
&\max_{c\in\mathcal{C}}\left\{\frac{1}{\rho_i^{(c)}}\left[Q_i^{(c)}(\tau+1)-\xi^{(c^+)}Q_i^{(c^+)}(\tau+1)-Ve_i\right]^+\right\}\\
\ge&\frac{1}{\rho_i^{(c_i^*(\tau))}}\left[Q_i^{(c_i^*(\tau))}(\tau+1)-Q_j^{(c_i^*(\tau))}(\tau+1)-Ve_i\right]^+\\
\ge&\frac{1}{\rho_i^{(c_i^*(\tau))}}\left[Q_i^{(c_i^*(\tau))}(\tau)-Q_j^{(c_i^*(\tau))}(\tau)-Ve_i\right]^+-\gamma_{\max}\\
=&\max_{c\in\mathcal{C}}\left\{\frac{1}{\rho_i^{(c)}}\left[Q_i^{(c)}(\tau)-\xi^{(c^+)}Q_i^{(c^+)}(\tau)-Ve_i\right]^+\right\}-\gamma_{\max}
\end{aligned}\tag{5.61}$$

We then have (5.21) from the two bounds above.

□

Proof. (of lemma 5.3)

(a) Without loss of generality, assume that $y\ge x$. Let k_y be the maximizer in the definition for $F(y)$, i.e. $F(y)=C_{ij}(k_y)[y-Ve_{ij}]^+-Vw_{ij}(k_y)$.

Since $F(x) \geq C_{ij}(k_y)[x - Ve_{ij}]^+ - Vw_{ij}(k_y)$ by definition, we then have

$$\begin{aligned} F(y) - F(x) \leq & F(y) - C_{ij}(k_y)[x - Ve_{ij}]^+ - Vw_{ij}(k_y) \\ \leq & C_{ij}(k_y)(y - x) \\ \leq & C_{ij,\max}(y - x) \end{aligned} \tag{5.62}$$

The result then follows by the fact that $F(x)$ is strictly increasing.

(b) If $\max\limits_{c\in\mathcal{C}} \{Q_i^{(c)}(\tau) - Q_j^{(c)}(\tau)\} > V\big(\min\limits_{k>0} \frac{w_{ij}(k)}{C_{ij}(k)} + e_{ij}\big)$, let $k' > 0$ be the minimizer of $\min\limits_{k>0} \frac{w_{ij}(k)}{C_{ij}(k)}$. We then have

$$\begin{aligned} & C_{ij}(k')\big[\max_{c\in\mathcal{C}} \{Q_i^{(c)}(\tau) - Q_j^{(c)}(\tau)\} - Ve_{ij}\big]^+ - Vw_{ij}(k') \\ > & C_{ij}(k')\Big(V \min_{k>0} \frac{w_{ij}(k)}{C_{ij}(k)} - V\frac{W_{ij}(k')}{C_{ij}(k')}\Big) = 0 \end{aligned} \tag{5.63}$$

hence the weight maximizing resource allocation is nonzero.

On the other hand, if $\max\limits_{c\in\mathcal{C}} \{Q_i^{(c)}(\tau) - Q_j^{(c)}(\tau)\} \leq V\big(\min\limits_{k} \frac{w_{ij}(k)}{C_{ij}(k)} + e_{ij}\big)$, then for any $k' > 0$, we have

$$\begin{aligned} & C_{ij}(k')\big[\max_{c\in\mathcal{C}} \{Q_i^{(c)}(\tau) - Q_j^{(c)}(\tau)\} - Ve_{ij}\big]^+ - Vw_{ij}(k') \\ \leq & C_{ij}(k')\Big(V \min_{k>0} \frac{w_{ij}(k)}{C_{ij}(k)} - V\frac{W_{ij}(k')}{C_{ij}(k')}\Big) \leq 0 \end{aligned} \tag{5.64}$$

Hence the resource allocation that maximizes the weight, $\max\limits_{k} \{C_{ij}(k)$ $\big[\max\limits_{c\in\mathcal{C}} \{Q_i^{(c)}(\tau) - Q_j^{(c)}(\tau)\} - Ve_{ij}\big]^+ - Vw_{ij}(k)\}$, is $k = 0$.

□

5.8.4 Proof of Theorem 5.1

Proof. The necessity of the Theorem follows the proof for [16, Theorem 1].

According to [16, Theorem 1], the capacity region with zero reconfiguration delay and cost, $\mathbf{\Lambda}$, consists of arrival rates $\lambda_i^{(c)}$ for which there exist multi-commodity flow variables $f_{ij}^{(c)}$, $f_i^{(c)}$ and probability values $\alpha_{i,k}, \alpha_{ij,k}, \beta_{i,k}^{(c)}, \beta_{ij,k}^{(c)}$ that satisfy:

$$\sum_{j\in\mathcal{V}^-(i)} f_{ji}^{(c)} + \xi^{(c)} f_i^{(c^-)} + \lambda_i^{(c)} \le \sum_{j\in\mathcal{V}^+(i)} f_{ij}^{(c)} + f_i^{(c)}, \ \forall i, c \tag{5.65}$$

$$f_i^{(c)} \le \frac{1}{\rho_i^{(c)}} \sum_{k\in\mathcal{K}_i} \alpha_{i,k}\beta_{i,k}^{(c)} C_i(k), \ \forall i, c \tag{5.66}$$

$$f_{ij}^{(c)} \le \frac{1}{\rho_{ij}^{(c)}} \sum_{k\in\mathcal{K}_{ij}} \alpha_{ij,k}\beta_{ij,k}^{(c)} C_{ij}(k), \ \forall (i,j), c \tag{5.67}$$

$$f_i^{(c)} \ge 0, \ \forall i, c, \ \ f_{ij}^{(c)} \ge 0, \ \forall (i,j), c \tag{5.68}$$

$$\sum_{k\in K_i} \alpha_{i,k} \le 1, \ \forall i, \ \ \sum_{k\in K_{ij}} \alpha_{ij,k} \le 1, \ \forall (i,j) \tag{5.69}$$

$$\sum_{c\in\mathcal{C}} \beta_i^{(c)} \le 1, \ \forall i, \ \ \sum_{c\in\mathcal{C}} \beta_{ij}^{(c)} \le 1, \ \forall (i,j) \tag{5.70}$$

Furthermore, the minimum average cloud network cost required for network stability under zero reconfiguration delay and cost is given by

$$\begin{aligned} h^*(\boldsymbol{\lambda}) = \min & \sum_i \sum_{k\in\mathcal{K}_i} \alpha_{i,k}\Big(w_i(k) + e_i C_i(k) \sum_c \frac{\beta_{i,k}^{(c)}}{\rho_i^{(c)}}\Big) \\ & + \sum_{(i,j)} \sum_{k\in\mathcal{K}_{ij}} \Big(w_{ij}(k) + e_{ij} C_{ij}(k) \sum_c \beta_{ij,k}^{(c)}\Big) \end{aligned} \tag{5.71}$$

where the minimum is over all $f_{ij}^{(c)}$, $f_i^{(c)}$, $\alpha_{i,k}$, $\alpha_{ij,k}$, $\beta_{i,k}^{(c)}$, $\beta_{ij,k}^{(c)}$ that satisfy (5.65)-(5.70).

The necessity of the Theorem (i.e. any rate $\boldsymbol{\lambda} \notin \mathbf{\Lambda}$ could not be stabilized by any policy) follows the same approach in the proof of [16, Theorem 1]. Therefore, we have $\mathbf{\Lambda}_\Delta \subset \mathbf{\Lambda}$ and $h_\Delta^*(\boldsymbol{\lambda}) \ge h^*(\boldsymbol{\lambda})$ for any reconfiguration delay/cost structure Δ. In the following, for each $\boldsymbol{\lambda} \in \mathbf{\Lambda}$

under any reconfiguration delay/cost structure Δ, we construct policies that stabilize $\boldsymbol{\lambda}$ with average cloud network cost arbitrarily close to $h^*(\boldsymbol{\lambda})$.

We first establish that for each $\boldsymbol{\lambda} = \{\lambda_i^{(c)}\}$ that is in the interior of $\boldsymbol{\Lambda}$, i.e. $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^o$, there exists a policy that stabilizes the cloud network under the exogenous arrival rate $\boldsymbol{\lambda}$ and under any finite reconfiguration delay and cost, $\Delta = \Big\{\{\delta_i\}_{i\in\mathcal{V}}, \{\delta_{ij}\}_{(i,j)\in\mathcal{E}}, \{\eta_i\}_{i\in\mathcal{V}}, \{\eta_{ij}\}_{(i,j)\in\mathcal{E}}\Big\}$. Notice that since reconfiguration cost does not affect the queue dynamics as in (5.1), hence we may ignore the reconfiguration cost when considering cloud network stability. We also denote the maximum reconfiguration delay as $\delta_{\max} = \max\{\max_{i\in\mathcal{V}} \delta_i, \max_{(i,j)\in\mathcal{E}} \delta_{ij}\}$.

Since $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^o$, there exists $\epsilon, \epsilon' > 0$ such that $(1+\epsilon')\boldsymbol{\lambda} + \epsilon\mathbf{1} \in \boldsymbol{\Lambda}^o$. Then substituting $\boldsymbol{\lambda}$ as $(1+\epsilon')\boldsymbol{\lambda} + \epsilon\mathbf{1}$ in (5.65), there exist variables $f_{ij}^{(c)}$, $f_i^{(c)}$, $\alpha_{i,k}$, $\alpha_{ij,k}$, $\beta_{i,k}^{(c)}$, $\beta_{ij,k}^{(c)}$ that satisfy (5.65)-(5.70).

At each time instance, we consider the flow allocation variables $\{\mu_i^{(c)}(t)\}_{i,c}$, $\{\mu_{ij}^{(c)}(t)\}_{(i,j),c}$ as a $L = (|\mathcal{V}| + |\mathcal{E}|)|\mathcal{C}|$-dimensional vector denoted by $\boldsymbol{\mu}(t) = [\mu_i^{(c)}(t); \mu_{ij}^{(c)}(t)] \in \mathbb{R}^L$. Let $\mathcal{M}$ be the set of all feasible L-dimensional flow allocations, i.e. $\boldsymbol{\mu}(t) \in \mathcal{M} \subset \mathbb{R}^L$.

Now concatenating flow variables $\{f_i^{(c)}\}_{i,c}$ and $\{f_{ij}^{(c)}\}_{(i,j),c}$ as $\mathbf{f} = [f_i^{(c)}; f_{ij}^{(c)}] \in \mathbb{R}^L$, it is straightforward from (5.66) and (5.67) that $\mathbf{f}$ is in the convex hull of $\mathcal{M}$. Hence according to Caratheodory's Theorem, $\mathbf{f}$ could be decomposed as a convex combination of $L+1$ vectors $\{\boldsymbol{\mu}_l\}_{l=1}^{L+1}$ in $\mathcal{M}$.

$$\mathbf{f} = \sum_{l=1}^{L+1} \gamma_l \boldsymbol{\mu}_l \tag{5.72}$$

With the decomposition given in (5.72), we may construct a periodic flow allocation schedule of period T such that each vector $\boldsymbol{\mu}_l$ in the $L+1$ vectors $\{\boldsymbol{\mu}_l\}_{l=1}^{L+1}$ is actively scheduled for $\gamma_l(T-(L+1)\delta_{\max})$ time slots within one period. In particular, each $\boldsymbol{\mu}_l$ is scheduled for consecutive $\gamma_l(T-(L+1)\delta_{\max}) + \delta_{\max}$ time slots, with the first $\delta_{\max}$ being reserved for reconfiguration. We then have $\sum_{t=kT}^{(k+1)T-1} \mu_{ij}^{(c)}(t)\mathbb{1}_{\{r_{ij}(t)=0\}} = (T-(L+1)\delta_{\max})f_{ij}^{(c)}$ and $\sum_{t=kT}^{(k+1)T-1} \mu_{i}^{(c)}(t)\mathbb{1}_{\{r_{i}(t)=0\}} = (T-(L+1)\delta_{\max})f_{i}^{(c)}$. Now consider the Lyapunov function $L(\mathbf{Q}) = \sum_{i,c} Q_i^{(c)}$, and we have that for each node i and each commodity c that:

$$\begin{aligned}
&\mathbb{E}\Big[Q_i^{(c)}(t+T) - Q_i^{(c)}(t)\Big|\mathbf{X}(t)\Big]\\
\le& T\lambda_i^{(c)} + \sum_{\tau=t}^{t+T-1}\mathbb{E}\Big[-\sum_{j\in\mathcal{V}^+(i)}\mu_{ij}^{(c)}(\tau)\mathbb{1}_{\{r_{ij}(t)=0\}} - \mu_i^{(c)}(\tau)\mathbb{1}_{\{r_i(\tau)=0\}}\\
&\qquad\qquad + \sum_{j\in\mathcal{V}^-(i)}\mu_{ji}^{(c)}\mathbb{1}_{\{r_{ji}(\tau)=0\}} + \xi_i^{(c)}\mu_i^{(c^-)}(\tau)\mathbb{1}_{\{r_i(\tau)=0\}}\Big|\mathbf{X}(t)\Big]\\
=& T\lambda_i^{(c)} + \Big(T-(L+1)\delta_{\max}\Big)\Big[-\sum_{j\in\mathcal{V}^+(i)}f_{ij}^{(c)} - f_i^{(c)}(\tau) + \sum_{j\in\mathcal{V}^-(i)}f_{ji}^{(c)} + \xi_i^{(c)}f_i^{(c^-)}(\tau)\Big]\\
\le& T\lambda_i^{(c)} - \Big(T-(L+1)\delta_{\max}\Big)\Big[(1+\epsilon')\lambda_i^{(c)} - \epsilon\Big] \qquad (5.73)
\end{aligned}$$

Then for any T that satisfies $(1-\frac{(L+1)\delta_{\max}}{T})(1+\epsilon') > 1$, or in other words, any $T > (1+\frac{1}{\epsilon'})(L+1)\delta_{\max}$, we have

$$\mathbb{E}\Big[Q_i^{(c)}(t+T) - Q_i^{(c)}(t)\Big|\mathbf{X}(t)\Big] \le -\Big(T-(L+1)\delta_{\max}\Big)\epsilon \qquad (5.74)$$

The strong stability then follows from Foster-Lyapunov Theorem.

For the average cloud network cost, with the solution $f_{ij}^{(c)}$, $f_i^{(c)}$, $\alpha_{i,k}$, $\alpha_{ij,k}$, $\beta_{i,k}^{(c)}$, $\beta_{ij,k}^{(c)}$ of (5.65)-(5.70), we may similarly construct a periodic flow allocation schedule that achieves arbitrarily close to the minimal average cloud network cost $h^*(\boldsymbol{\lambda})$. In particular, following the same construction, a periodic flow schedule of period T achieves average cost of $h^*(\boldsymbol{\lambda}) + \frac{L+1}{T}(\sum_{i\in\mathcal{V}}\eta_i + \sum_{(i,j)\in\mathcal{E}}\eta_{ij})$. Since T could be arbitrarily large without affecting the stability (as long as $T > (1+\frac{1}{\epsilon'})(L+1)\delta_{\max}$), as shown previously in (5.74), we have $h_\Delta^*(\boldsymbol{\lambda}) = h^*(\boldsymbol{\lambda})$. □

Chapter 6

Conclusions

The concept of throughput optimality has been an essential first order performance metric as it pertains to the robustness of the scheduling of queueing systems. In this thesis, we have shown that the presence of the reconfiguration delay not only incurs overhead on the delay performance, but also degrades the fundamental throughput optimality guarantee for many scheduling policies not addressing the reconfiguration delay. Nonetheless, the capacity region remains the same for any fixed reconfiguration overhead. For both the centralized and distributed setup considered in this thesis, we propose scheduling policies that achieve throughput optimality for any reconfiguration delay and do not require any prior knowledge on the arrival traffic.

For the delay performance of the Adaptive MaxWeight policy, the analysis in this thesis is restricted in the heavy traffic regime and that the arrival rate matrix approaches to an all-port saturated arrival rate. Some open questions arise from this analysis may be to address more general arrival rates. For example, switches with heavy traffic limit to an arrival rate that only a portion of the ports are saturated have been considered in the literature, in the context without reconfiguration delay. It would be interesting to consider a similar setup for switches with reconfiguration delay to complete the characterization of the heavy traffic queue length behavior of the AMW policy. An even more relevant but challenging direction would be to consider the delay analysis in non-asymptotic regime, where a tight queue length bound for switch scheduling remains an open question.

In Chapter 5, we considered the dynamic scheduling of a distributed computing network where each computing function could be instantiated or terminated at any time and at any node in the network. This inherently assumes that the computing functions are stateless, or do not evolve over time. With the wider adoption of machine learning modules and data-driven processing discipline, a possible future direction is to also consider functions that require data transfers when instantiated at a different node. For this type of functions, not only the reconfiguration delay needs to further incorporate data transfer delay, we also need to consider the data transfer which occupies the network resources.

Bibliography

[1] AWS direct connect pricing. https://aws.amazon.com/directconnect/pricing/. Accessed: 2018-08-01.

[2] Why Big Data Needs Big Buffer Switches. https://www.arista.com/assets/data/pdf/Whitepapers/BigDataBigBuffers-WP.pdf.

[3] BARCELO, M., LLORCA, J., TULINO, A. M., AND RAMAN, N. The cloud service distribution problem in distributed cloud networks. In *2015 IEEE International Conference on Communications (ICC)* (June 2015), pp. 344–350.

[4] BERTSIMAS, D., GAMARNIK, D., AND TSITSIKLIS, J. N. Performance of multiclass markovian queueing networks via piecewise linear lyapunov functions. *Annals of Applied Probability 11* (2001), 1384–1428.

[5] CELIK, G., BORST, S. C., WHITING, P. A., AND MODIANO, E. Dynamic scheduling with reconfiguration delays. *Queueing Syst. Theory Appl. 83*, 1-2 (June 2016), 87–129.

[6] CELIK, G., AND MODIANO, E. Scheduling in networks with time-varying channels and reconfiguration delay. *Networking, IEEE/ACM Transactions on 23*, 1 (Feb 2015), 99–113.

[7] CHAN, C. W., ARMONY, M., AND BAMBOS, N. Maximum weight matching with hysteresis in overloaded queues with setups. *Queueing Syst. Theory Appl. 82*, 3-4 (Apr. 2016), 315–351.

[8] CHANG, C.-S., CHEN, W.-J., AND HUANG, H.-Y. Birkhoff-von neumann input buffered crossbar switches. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (Mar 2000), vol. 3, pp. 1614–1623 vol.3.

[9] COHEN, R., LEWIN-EYTAN, L., NAOR, J. S., AND RAZ, D. Near optimal placement of virtual network functions. In *2015 IEEE Conference on Computer Communications (INFOCOM)* (April 2015), pp. 1346–1354.

[10] DASYLVA, A., AND SRIKANT, R. Optimal wdm schedules for optical star networks. *IEEE/ACM Transactions on Networking 7*, 3 (Jun 1999), 446–456.

[11] DESTOUNIS, A., PASCHOS, G. S., AND KOUTSOPOULOS, I. Streaming big data meets backpressure in distributed network computation. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications* (April 2016), pp. 1–9.

[12] ERYILMAZ, A., AND SRIKANT, R. Asymptotically tight steady-state queue length bounds implied by drift conditions. *Queueing Syst. Theory Appl. 72*, 3-4 (Dec. 2012), 311–359.

[13] FARRINGTON, N., FORENCICH, A., PORTER, G., SUN, P. C., FORD, J. E., FAINMAN, Y., PAPEN, G. C., AND VAHDAT, A. A multiport microsecond optical circuit switch for data center networking. *IEEE Photonics Technology Letters 25*, 16 (Aug 2013), 1589–1592.

[14] FENG, H., LLORCA, J., M. TULINO, A., AND F. MOLISCH, A. Optimal control of wireless computing networks. *IEEE Transactions on Wireless Communications PP* (10 2017).

[15] FENG, H., LLORCA, J., TULINO, A. M., AND D. RAZ, A. F. M. Approximation algorithms for the nfv service distribution problem. In *2017 IEEE Conference on Computer Communications (INFOCOM)* (April 2017).

[16] FENG, H., LLORCA, J., TULINO, A. M., AND MOLISCH, A. F. Dynamic network service optimization in distributed cloud networks. In *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (April 2016), pp. 300–305.

[17] FENG, H., LLORCA, J., TULINO, A. M., AND MOLISCH, A. F. Optimal dynamic cloud network control. In *2016 IEEE International Conference on Communications (ICC)* (May 2016), pp. 1–7.

[18] FENG, H., LLORCA, J., TULINO, A. M., AND MOLISCH, A. F. Optimal dynamic cloud network control. *IEEE/ACM Trans. Netw. 26*, 5 (Oct. 2018), 2118–2131.

[19] GIACCONE, P., PRABHAKAR, B., AND SHAH, D. Randomized scheduling algorithms for high-aggregate bandwidth switches. *Selected Areas in Communications, IEEE Journal on 21*, 4 (May 2003), 546–559.

[20] HAJEK, B. Notes for ece 534: An exploration of random processes for engineers, 2009.

[21] HSIEH, P.-C., LIU, X., JIAO, J., HOU, I.-H., ZHANG, Y., AND KUMAR, P. R. Throughput-optimal scheduling for multi-hop networked transportation systems with switch-over delay. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (New York, NY, USA, 2017), Mobihoc '17, ACM, pp. 16:1–16:10.

[22] KESLASSY, I., AND MCKEOWN, N. Analysis of scheduling algorithms that provide 100% throughput in input-queued switches, 2001.

[23] KRISHNASAMY, S., T., A. P., ARAPOSTATHIS, A., SHAKKOTTAI, S., AND SUNDARESAN, R. Augmenting max-weight with explicit learning for wireless scheduling with switching costs. In *IEEE INFOCOM 2017 - The 36th Annual IEEE International Conference on Computer Communications* (May 2017), pp. 1–9.

[24] Li, X., and Hamdi, M. On scheduling optical packet switches with reconfiguration delay. *IEEE Journal on Selected Areas in Communications 21*, 7 (Sept 2003), 1156–1164.

[25] Li, Y., Panwar, S., and Chao, H. Frame-based matching algorithms for optical switches. In *High Performance Switching and Routing, 2003, HPSR. Workshop on* (June 2003), pp. 97–102.

[26] Little, J. D. C. A proof for the queuing formula: L= w. *Operations Research 9*, 3 (1961), 383–387.

[27] Liu, H., Lu, F., Forencich, A., Kapoor, R., Tewari, M., Voelker, G. M., Papen, G., Snoeren, A. C., and Porter, G. Circuit switching under the radar with reactor. In *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation* (Berkeley, CA, USA, 2014), NSDI'14, USENIX Association, pp. 1–15.

[28] Maguluri, S. T., Burle, S. K., and Srikant, R. Optimal heavy-traffic queue length scaling in an incompletely saturated switch. *SIGMETRICS Perform. Eval. Rev. 44*, 1 (June 2016), 13–24.

[29] Maguluri, S. T., and Srikant, R. Queue length behavior in a switch under the maxweight algorithm. *arxiv preprint* (2015).

[30] U.S. Energy Information Administration. Electric power monthly: with data for may 2018. `https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=epmt_5_6_a`. Accessed: 2018-07-20.

[31] McKeown, N., Anantharam, V., and Walrand, J. Achieving 100% throughput in an input-queued switch. In *INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE* (Mar 1996), vol. 1, pp. 296–302 vol.1.

[32] Milenkoski, A., Jaeger, B., Raina, K., Harris, M., Chaudhry, S., Chasiri, S., David, V., and Liu, W. Security position paper: Network function virtualization, 04 2016.

[33] Morabito, R. Power consumption of virtualization technologies: An empirical investigation. In *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)* (Dec 2015), pp. 522–527.

[34] Neely, M. J. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool Publishers, 2010.

[35] Neely, M. J. Stability and probability 1 convergence for queueing networks via lyapunov optimization. *Journal of Applied Mathematics* (2012).

[36] Paris, S., Destounis, A., Maggi, L., Paschos, G. S., and Leguay, J. Controlling flow reconfigurations in sdn. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications* (April 2016), pp. 1–9.

[37] PORTER, G., STRONG, R., FARRINGTON, N., FORENCICH, A., CHEN-SUN, P., ROSING, T., FAINMAN, Y., PAPEN, G., AND VAHDAT, A. Integrating microsecond circuit switching into the data center. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (New York, NY, USA, 2013), SIGCOMM '13, ACM, pp. 447–458.

[38] RAZAVI, K., RAZOREA, L. M., AND KIELMANN, T. Reducing vm startup time and storage costs by vm image content consolidation. In *Euro-Par Workshops* (2013).

[39] ROSS, K., AND BAMBOS, N. Adaptive batch scheduling for packet switching with delays. In *High-performance Packet Switching Architectures*, I. Elhanany and M. Hamdi, Eds. Springer London, 2007, pp. 65–79.

[40] ROSS, K., AND BAMBOS, N. Projective cone scheduling (pcs) algorithms for packet switches of maximal throughput. *IEEE/ACM Transactions on Networking 17*, 3 (June 2009), 976–989.

[41] SHAH, D., AND KOPIKARE, M. Delay bounds for approximate maximum weight matching algorithms for input queued switches. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (2002), vol. 2, pp. 1024–1031 vol.2.

[42] SHAH, D., AND WISCHIK, D. Optimal scheduling algorithms for input-queued switches. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings* (April 2006), pp. 1–11.

[43] STOLYAR, A. L. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *Ann. Appl. Probab. 14*, 1 (02 2004), 1–53.

[44] TASSIULAS, L. Adaptive back-pressure congestion control based on local information. *IEEE Transactions on Automatic Control 40*, 2 (Feb 1995), 236–250.

[45] TASSIULAS, L. Linear complexity algorithms for maximum throughput in radio networks and input queued switches. In *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (Mar 1998), vol. 2, pp. 533–539 vol.2.

[46] TASSIULAS, L., AND EPHREMIDES, A. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *Automatic Control, IEEE Transactions on 37*, 12 (Dec 1992), 1936–1948.

[47] THAKULSUKANANT, K. Mems technology for optical switching. *Walailak Journal of Science and Technology 10* (2013), 9–18.

[48] VENKATAKRISHNAN, S. B., ALIZADEH, M., AND VISWANATH, P. Costly circuits, submodular schedules and approximate carathodory theorems. In *Proceedings of the 13th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems* (2016), SIGMETRICS '16, ACM.

[49] WANG, C. H., AND JAVIDI, T. Adaptive policies for scheduling with reconfiguration delay: An end-to-end solution for all-optical data centers. *IEEE/ACM Transactions on Networking 25*, 3 (June 2017), 1555–1568.

[50] WANG, C.-H., JAVIDI, T., AND PORTER, G. End-to-end scheduling for all-optical data centers. In *INFOCOM, 2015 Proceedings IEEE* (April 2015).

[51] XIE, Y., LIU, Z., WANG, S., AND WANG, Y. Service Function Chaining Resource Allocation: A Survey. *ArXiv e-prints* (July 2016).