The Hong Kong Polytechnic University
Department of Computing

Relative Stability Analysis of Multiple Queueing Systems

Lam Sum

A thesis submitted in partial fulfilment of
the requirements for the degree of
Doctor of Philosophy

January 2008

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

________________________________ (Signed)

Lam Sum

________________________________ (Name of student)

To my parents...

ABSTRACT

Stability is always the most fundamental issue to consider in the development of any system with finite resources, simply because an unstable system is not operable in a real-world environment. In this research, we study the stability problems for single-server systems with multiple queues in which the server services the customers arriving at the queue according to some service policy. Our contribution is to tackle the stability problems from a relative stability point of view and, as a result, to obtain a number of new systems and queue stability results.

We consider two kinds of stability problems in the single-server-multiple-queue systems (SSMQSs)—absolute stability and relative stability. The absolute stability concerns the stability status of the objects under studied. The objects could be individual queues or the entire system, and an absolute stability analysis answers questions, such as whether the objects are stable for some given system inputs. The relative stability, on the other hand, concerns the stability relations among two or more queues and answers questions, such as whether some queues are more (or less) stable than others.

There are three types of absolute stability problems: queue stability, system stability, and degree of stability. The queue/system stability problems aim at obtaining the queue/system stability conditions, and the degree of stability problem measures how stable a queue is. On the other hand, the relative stability problem aims at comparing the degree of stability for two or more queues. Moreover, it involves obtaining the conditions for a given relative stability relation to hold. Knowing the relative stability also helps determine the queue stability conditions. Therefore, our focus of this work is on the relative stability analysis of the SSMQS.

Obtaining the relative stability results of the SSMQSs consists of several steps. First, we provide a criterion to classify the SSMQSs. This classification allows us to identify system models in which the degree of stability of a queue can be defined through indirectly utilizing the Loynes' theorem. The relative stability among the queues can then be defined for the models. The next step is to investigate useful properties of the models, and in particular, we discover properties related to the relative stability. One property is the sufficient and necessary relative stability conditions for any two queues in the models. Another is the existence of the maximum as stable as configuration of the system. Through these properties we can solve the relative stability problems that we have introduced completely. In addition, the properties also allow us to reformulate three problems—system stability region characterization, system stabilization, and achieving maximum stable throughput—into one single optimization problem and provide clue to solving the optimization problem.

The relative stability properties are not only interesting and important in themselves but also essential to solving the queue stability problems. Since queue stability is more general than system stability, the relative stability is also useful to solving the system stability problems. To show the importance of the relative stability properties, we select four practical systems from a class of SSMQS models and investigate both absolute stability and relative stability conditions of these four systems. With the relative stability properties, we can see the approach to derive the stability conditions in the single-server-multiple-queue systems is unified and straightforward, though because of the complexity of some systems, the exact absolute stability conditions for those systems may not be found. Nevertheless, through the relative stability results we can always provide necessary stability conditions for the class of systems.

ACKNOWLEDGMENTS

First of all, I would like to express my genuine gratitude to my supervisor, Professor Rocky Kow-Chuen CHANG for his guidance, patience, encouragement, and contributions in the development of my research. Without his insight, advice, and support, this work would not have been possible. His extensive knowledge, strong analytical skill, and commitment to the excellence of research and teaching are truly treasures to his students. He is willing to share his knowledge and career experience and give emotional and moral encouragement. He is more than an adviser and a teacher but a role model and a friend. Under his supervision is really one of the most rewarding experiences of me.

I would like to thank Prof. CAO Xiren at the Hong Kong University of Science and Technology, Prof. Lang TONG at Cornell University, and Prof. Henry C.B. CHAN for being my examiners and providing valuable comments to the improvement of this dissertation, and for their kindness during the whole process.

I would also like to thank my group-mates of the networking research group, Yi Xie, Xiapu Luo, Samantha Lo, Edmond Chan, Steve Poon, and Kathy Tang, for their considerable help and constructive suggestions.

I am forever indebted to my parents for their endless love and support, and to my brother and sister for their encouragements.

Finally, I thank Maggie for her help and constant support in the past years, most importantly, for her love.

TABLE OF CONTENTS

Page

ABSTRACT iv

LIST OF TABLES ix

LIST OF FIGURES x

SYMBOLS xi

1 Introduction 1

1.1 Absolute Stability and Relative Stability 1

1.2 Problem Statement and Motivation 5

1.3 Contributions 8

1.4 Outline of the Dissertation 9

2 Background 11

2.1 Stability Definitions 11

2.1.1 Lyapunov Stability 11

2.1.2 Stability of Stochastic Systems 13

2.1.3 Stability of Markov Chains 15

2.1.4 Stability in Queueing Systems 17

2.2 Methods of Studying Stability 19

2.2.1 Lyapunov Function Methods 19

2.2.2 Fluid Model Approach 22

2.2.3 Non-Markovian Analysis Methods 23

2.3 Stability Results for Multiple Queue Systems 26

2.4 Summary 30

3 Relative Stability Relation in Single-Server-Multiple-Queue Systems 31

3.1 Introduction 31

3.2 A Polling System with Gated Limited Service 31

Page

3.3 A Slotted Buffered ALOHA Network 37

3.4 A Processor Sharing System 41

3.5 Summary 45

4 Relative Stability 48

4.1 Introduction 48

4.2 Two Types of Single-Server-Multiple-Queue Systems 49

4.3 Degree of Stability 53

4.4 Properties and Relative Stability in Type-1 SSMQSs 58

4.5 Summary 78

5 Applications 79

5.1 The Models 79

5.1.1 NDMA and BNDMA 80

5.1.2 A Slotted Buffered ALOHA Network with Multipacket Reception 81

5.2 Relative Stability of Type-1 Systems 82

5.2.1 The Polling System with Gated Limited Service 83

5.2.2 The Slotted Buffered ALOHA Network 84

5.2.3 NDMA and BNDMA 85

5.2.4 The MPR Model 86

5.3 Absolute Stability Conditions of Type-1 SSMQSs 87

5.3.1 The Polling System with Gated Limited Service 90

5.3.2 The Slotted Buffered ALOHA Network 94

5.3.3 NDMA and BNDMA 97

5.3.4 The MPR Model 100

5.3.5 Stability Region Characterization 102

5.4 Summary 103

6 Conclusion and Future Research 106

6.1 Conclusion 106

6.2 Future Research 108

LIST OF TABLES

Table Page

2.1 Common stability definitions. 19

5.1 Comparison of upper bounds of λ_1 for $k = 3$ and $p_1 = p_2 = p_3 = 0.5$ . . . 96

5.2 Comparison of upper bounds of λ_1 for $k = 3$ and $p_1 = 0.8, p_2 = 0.7, p_3 = 0.6$ 96

5.3 Comparison of upper bounds of λ_1 for $k = 5$ and $p_i = 0.5$ for all $i = 1..5$ 97

5.4 Comparison of upper bounds of λ_1 for $k = 10$ and $p_i = 0.1$ for all $i = 1..10$. 97

LIST OF FIGURES

Figure Page

1.1 A classification of stability problems. 5

1.2 A single-server-multiple-queue system. 6

5.1 The first method to achieve system stability. 88

5.2 The second method to achieve system stability. 88

SYMBOLS

Common notations

k	number of queues
q_i	the ith queue
Q_i^n	queue length process at discrete time n
$\mathbf{Q}^n$	joint queue length process
λ_i	average arrival rate at q_i
ρ_i	server utilization at q_i
ρ	total server utilization
Λ	a traffic point $(\lambda_1, \lambda_2, ..., \lambda_k)$
$\mathcal{U}$	subset of unstable queues
$\mathcal{S}$	subset of stable queues
τ_i	last discrete time when Q_i^n is less than or equal to a constant
τ	$\sup\{\tau_i, \forall q_i \in \mathcal{U}\}$
K^n	the $n{+}1$ discrete time after τ the queues in $\mathcal{S}$ are all empty
R^{n+1}	$K^{n+1} - K^n$
H^n	stationay and ergodic random sequence in Type-1 SSMQSs
μ	average service rate at a queue
$\hat{\mu}$	maximum service rate at a queue
L	a monotonically increasing path of system traffic
K	direction vector of a linear increasing path $(k_1, k_2, ..., k_n)$
L_K	a linear increasing path
$D_L^\lambda(q)$	the degree of stability of a queue at Λ on L
$C_{i,j}$	constant ration between two queues' direction component when the two queues are as stable as each other

$\mathcal{G}$	configuration of a SSMQS
$\mathfrak{S}(L_K, \mathbf{G})$	system stability region on L_K with configuration $\mathcal{G}$
$\mathfrak{D}(L_K)$	closure of $\mathfrak{S}(L_K, \mathbf{G})$ on L_K
$\mathcal{M}_t$	set of queues that are more stable than a target queue q_t
$\mathcal{A}_t$	set of queues that are as stable as a target queue q_t
$\mathcal{L}_t$	set of queues that are less stable than a target queue q_t
Γ	a certain partition of the queues in terms of $\mathcal{M}_t$, $\mathcal{A}_t$, and $\mathcal{L}_t$
L_Γ	paths that can have the same partition of queues Γ

Polling systems with limited service policy

$A_i(t)$	total arrivals at q_i up to time t
B_i	service time process at q_i
b_i	first moment of B_i
M_i	maximum customers can be served at q_i in a cycle
U_i	switch-over time process at q_i
u_i	first moment of U_i
u_0	average total switch-over time in a cycle
$A_i^{1,n}$	arrivals at q_i between server's arrivals at q_1 and q_i in cycle n
$A_i^{2,n}$	arrivals at q_i between server's arrivals at q_i and q_1 in cycle n
X^n	customers served at q_i in cycle n
$\mathbf{E}X_i$	expectation of X^n
C	cycle time process
$\mathbf{E}C$	expectation of C

Slotted buffered ALOHA network

A_i^n	total arrivals at q_i during slot n
p_i	transmission probability of q_i
B_i^n	successful transmission at q_i during slot n
$\mathbf{E}B_i$	expectation of B_i^n

$\mathbf{z}$	k-dimensional random binary vector
P_i^s	$\lim_{n\to\infty} P(B_i^n = 1)$

Processor sharing system

$\boldsymbol{\Psi}$	marked point process $\{[T_l, (S_l, I_l)]\}_{l=-\infty}^{\infty}$
T_l	arrival epoch of the lth customer
S_l	service time requirement of the lth customer
I_l	the queue at which the lth customer joins
b_i	expecation of S_l at q_i
$g(t)$	number of non-empty queues at time t
$R_{i,l}(t)$	residual service time of the l customer at q_i at time t
$C^*(t)$	service received by any served customer at time t

1. INTRODUCTION

1.1 Absolute Stability and Relative Stability

In the last two decades, communication and computer networks have experienced a fascinating advance. New technologies and applications have been invented and developed at all the layers in the network architecture. More importantly, this marching to the new developments has never been slow down but become faster and faster. In order to provide more efficient and effective networking environments and to accommodate more new applications, huge efforts have been attracted from the research community to analyze and improve the performances of network systems.

Typical performance issues of network systems, to list a few, include throughput of channels, delay of packets, efficiency of paths, schedulability of flows, and stability of nodes. Very often, these issues can be formulated and analyzed as queueing problems. Among them, the stability problem is probably the most fundamental one: a stable network is good, and an unstable network is bad. Generally speaking, a queue is stable if the length of the queued up customers does not grow to infinity as time goes by. And a system is stable if all the queues in the system are stable. In this work, we are interested in the stability issues of the *single-server-multiple-queue systems* in which multiple queues contend for services provided by a single server.

For a queueing system, one of the basic questions is to determine whether the system is stable or not under some given inputs. Moreover, because sometimes it is possible that some queues are stable while the whole system is not, in these situations, instead of the whole system, the major concern may be the stability of some special queues. To differentiate these two kinds of concerns, we call the former as the *system stability* while the latter the *queue stability.* It is easy to see that system stability implies queue stability but the reverse case is not necessarily true.

To answer the above stability questions, i.e., to determine the stability status of a system or some queues for some given inputs, it naturally brings up the needs of finding the ultimate *stability conditions* of the system and the queues, which in general is the goal of stability analysis of queueing systems. Usually, the stability conditions are in the forms of parameter regions with respect to the traffic input parameters and called the stability regions. If the given traffic input are within the stability region, the system or the queues under consideration are stable; otherwise, they are unstable. Intuitively, we consider that the stability conditions reflect the qualitative aspect of the stability, i.e., it can determine whether the given system or queues are stable or not for some given traffic inputs. The stability conditions are also static and intrinsic to the system, that is, once the system is given, the stability conditions will be constant and independent to the traffic inputs. The system and queue stability (status), however, are dynamic as they will be affected by the traffic inputs.

Another kind of questions regarding to stability is to ask, say, if a queue is stable under some given traffic inputs, how stable it is. This kind of questions in general cannot be answered directly from the stability conditions. To address the question, we need to go one step further to find out the *degree of stability* of the queue for the given traffic inputs. In particular, we are interested in how the degree of stability be affected by the traffic inputs. Contrast to the stability conditions, the degree of stability reflects the quantitative aspect of the stability, i.e., a measurement of stability, and is dynamic as it will also be affected by the traffic inputs. As both the stability conditions and degree of stability can tell us the stability status of a system or a queue, either qualitatively or quantitatively, we refer them collectively as the *absolute stability.*

Once we know the degree of stability of the queues, it is natural to compare the queues' degree of stability to see which queue is more stable. This comparison leads to another kind of issues related to stability, and we call them the *relative stability.* The reasons of this naming are, firstly, through the relative stability we cannot directly tell whether a system or some queues are stable or not, and secondly, the relative stability concerns the stability relations among the queues. For the relative stability, on one hand, we are interested in the *relative stability relations* among the queues, i.e., which queue is more (or less) stable in terms of degree of stability. These relations of the queues allow us to derive an order of the queues in terms of stability. On the other hand, we like to find out the *relative stability conditions* under which the queues have certain relative stability relations. The relative stability conditions in general can be represented by the traffic input parameters of the queues. Therefore, through the relative stability conditions we can tell how the relative stability relations among the queues be affected by the traffic inputs. Similar to the absolute stability conditions, the relative stability condition is also qualitative and static, i.e., it can determine which queue is more (or less) stable, and is constant and independent to the traffic inputs. The relative stability relation, on the other hand, is dynamic because whether a queue is more (or less) stable than another depends on the traffic inputs and how the inputs vary. Next to the relative stability relation, another step will then be to find out the differences, or the distances, among the queues in terms of degree of stability.

Now we must point out that the above discussion is valid only when the concept of stability as well as the concept of degree of stability are well-defined, and in addition that the definition of degree of stability is consistent with the definition of stability. At this point we assume that these definitions can be well-defined and leave the detailed definitions to Chapters 2 and 4.

Figure 1.1 summarizes the aforementioned stability problems and the intuitive relations and properties among them. The *queue stability* reflects the stability status of a given queue. This stability status can be determined by the *queue stability condition* (①). Similarly, we have the *system stability* and the *system stability condition*, and their relation (②). The relation between the queue stability and the system stability is that the queue stability is necessary to the system stability, as the system stability is just a special case of queue stability, i.e., all the queues are stable. Therefore, the queue stability problem is more general than the system stability and we can achieve system stability through queue stability (③), i.e., the conditions for all the queues to be stable. From the queue stability we may able to introduce the concept of *degree of stability* to measure how stable a queue is. The dot-line arrow ④ indicates that the definition of the degree of stability depends on the definition of the queue stability. Once we have a way to compute the degree of stability of a queue, consequently, we can determine the queue's stability status as well as the queue stability condition (⑤). The intuitive reason of the latter conclusion is that the queue stability condition is equivalent to the condition of maintaining the degree of stability at certain level. We call the queue stability, system stability, and the degree of stability collectively as the *absolute stability*. Through comparing the degree of stability of the queues, we can achieve the *relative stability relations* as well as the *relative stability conditions*. The latter conclusion is because the relative stability condition is equivalent to the condition of maintaining a certain relation among the queues' degree of stability (⑥). The relative stability relation can be determined by the relative stability condition (⑦). Finally, the dash-line arrow ⑧ indicates that the relative stability can assist in achieving the queue stability. This is because, for a given queue, the relative stability can tell us which queues are more (and less) stable than the given queue. This information is indeed essential to the method developed in this study to study queue stability.

From Figure 1.1 we can see that the degree of stability is probably the most dominant problem in the sense that solving the degree of stability can lead to solutions to the other mentioned stability problems. This dominant position, however, makes the degree of stability the most difficult problem to tackle. Fortunately, as the relative stability can be determined merely by comparing the queues' degree of stability, it is thus possible to have these comparison results indirectly without explicitly computing the degree of stability of the queues. In this study we achieve this through investigating a set of relative stability related properties of a class of single-server-multiple-queue systems. The set of properties allows us to completely solve the relative stability problems in the systems. Once this is done, as indicated

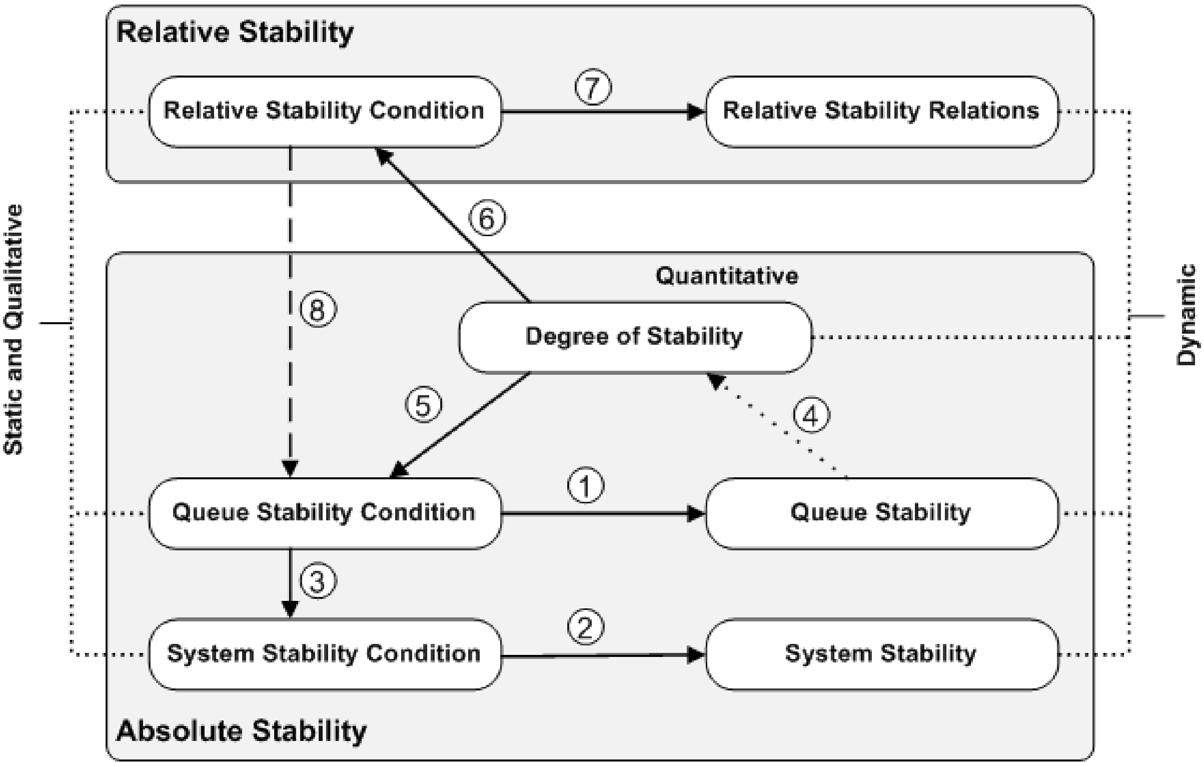

Fig. 1.1. A classification of stability problems.

in Figure 1.1 relation ⑧, we utilize the relative stability to analyze queue stability problems, and subsequently, through the queue stability to achieve system stability. This thread will be the main clue in this work to study the stability problems of the class of single-server-multiple-queue systems. It makes the relative stability the main object of this study.

1.2 Problem Statement and Motivation

Consider a single-server-multiple-queue system (SSMQS) in which k distributed queues is multiplexed by a single server. The topological structure of a typical SSMQS is illustrated in Figure 1.2.

In a SSMQS, at the ith queue, the requests arrive according to some arrival processes A_i. We assume that there is an unlimited buffer at each queue to store

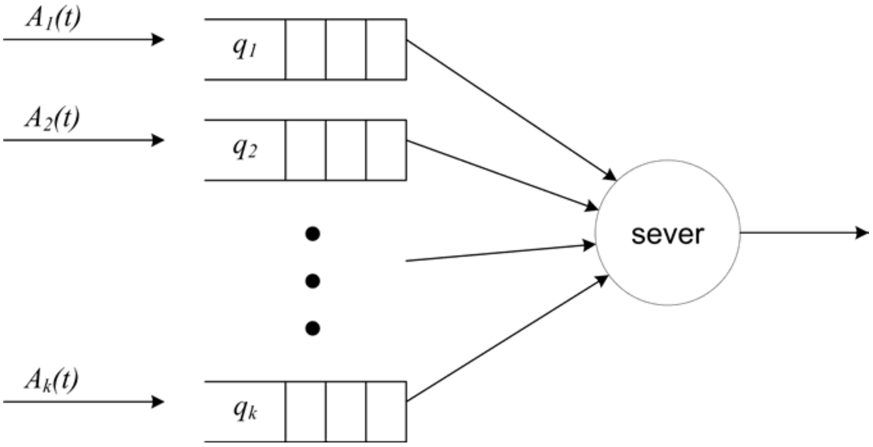

Fig. 1.2. A single-server-multiple-queue system.

the unprocessed requests. At certain moment, the single server employs a scheduling algorithm to determine which queue will be served next. Upon serving a queue, the server determines how many requests can be entertained according to some service policies. Furthermore, there are service time processes at each queue, the setup time processes the server incurred at each queue between its arrival and the actual start of service, and the switch-over time processes the server incurred between its departure at one queue and its arrival at the next queue.

In this study, we identify a special class of SSMQSs in which a single definition of queue stability can be shared based on some common assumptions of the involved processes, e.g., arrival and service time processes. In addition, the scheduling algorithms and the service policies used in the systems ensure that a stable queue will always has a stationary regime of the state process even when some other queues are unstable. For this set of SSMQSs, we address the following issues:

- Degree of stability: Define the concept of degree of stability.
- Relative stability: Analyze the relative stability conditions in those systems; derive the relative stability conditions for some particular systems.

- Absolute stability: Utilize relative stability to analyze queue and system stability; derive queue and system stability conditions for some particular systems.

- Other issues: Utilize the relative stability to study the characterization problem of the system stability regions; prove the equivalence of three problems: the characterization problem of the system stability regions, the stabilization problem of the system, and achieving the maximum stable throughput of the system.

This study is motivated by the need of a general study of the relative stability for SSMQSs. On the theoretical level, we believe that the relative stability is a natural step to go beyond the absolute stability. The relative stability requires us to compare queues in terms of stability. This leads to the definition of the concept of degree of stability, which can be used to measure how stable an individual queue is. The introduction of the degree of stability enriches the absolute stability, which originally considers the stability conditions mainly. For the SSMQSs, through the relative stability conditions, we can tell how the relative stability relations are affected by the system traffic inputs. This kind of questions simply cannot be answered by the absolute stability conditions.

On the application level, the relative stability can facilitate the analysis of the absolute stability conditions of the SSMQSs. This has been evidenced in the stability analysis for the slotted ALOHA network [17, 47] and polling systems [16, 29]. In [47], a necessary stability condition and a better sufficient stability condition for the slotted ALOHA network were derived. In [29] the local stability condition of a version of polling system was derived. While in [16, 17], the queue stability conditions of the slotted ALOHA network and a version of polling systems were established. These results all require the information of the ordering of the queues becoming unstable when the system traffic increases in certain ways, and this stability ordering information can be directly derived from the relative stability conditions of the systems. Therefore, a general study of relative stability not only facilitates the queue stability analysis, but also improves the system stability to a certain extent. This is why we consider the relative stability can be served as a thread to link the stability problems together.

1.3 Contributions

We believe this work is the first attempt to study stability problems of the SSMQSs from the relative stability perspective. The main contribution of this work is the establishment of relative stability results for a class of SSMQSs. The results consist of a set of relative stability related properties of the SSMQSs and some approaches for deriving both absolute and relative stability conditions in the SSMQSs. The set of properties allow us to better understand the SSMQSs. More importantly, based on this set of properties, we can develop unified and effective approaches to achieve both absolute and relative stability conditions in the SSMQSs. The approaches are unified in the sense that they are applicable to a class of SSMQSs. The approaches are also effective in the sense that, in terms of relative stability, the approaches can be easily applied to obtain the relative stability conditions of the SSMQSs completely; while in terms of absolute stability, the approaches can derive both queue and system stability conditions, though for some systems, only separate necessary or sufficient stability conditions can be obtained. In contrast, the existing approaches in general can only be used to study the system stability conditions, though there are some independent results regarding to the queue stability conditions as well as the relative stability have been reported. To investigate the properties, we provide a classification to the SSMQSs. Such a classification allows us to identify the kind of SSMQSs which is both *specific* and *general* enough in the sense that, on one hand, the relative stability problems we have mentioned for the systems can be completely solved, and on the other hand, the systems can cover the typical queueing models that are commonly used in performance evaluation of computer and communication networks. To study relative stability, we also propose a formal definition of degree of stability of a queue.

The introduction of the concept of degree of stability enriches the content of the absolute stability problems of the SSMQSs.

The second contribution of this work consists of those actual relative and absolute stability conditions for some practical systems. These results are the direct applications of our approaches. Besides reproducing the previously reported results, we are able to obtain new results such as the relative stability conditions for a slotted ALOHA networks with multipacket receptions, and a necessary system stability condition for the slotted buffered ALOHA network that is better than the existing one. Through the relative stability related properties, we can also solve the stability region characterization problems of the SSMQSs, for instances, in this study we obtain the closure of the system stability region of the ALOHA network with or without multipacket reception. The properties further allow us to reformulate the following three problems, namely, the characterization problem of the system stability region, the stabilization problem of the system for a given traffic point, and finding the maximum stable throughput, into a single optimization problem. And the solution of the optimization problem is to find a specific configuration of the system under which all the queues have a specific relative stability relation and the stable system throughput achieves its maximum at the system stability boundary when the system traffic increases in a certain way.

1.4 Outline of the Dissertation

The rest of the dissertation is organized as follows: In Chapter 2 we provide some background related to this study. We review some different definitions of the concept of stability, some different approaches to study the stability in queueing systems, and some previous stability results of the SSMQSs. In Chapter 3 we study relative stability relations in three typical SSMQSs. Through the study we show that there are common factors available to all the three systems. These factors provide us a clue to classify the SSMQSs in a way such that we can define degree of stability for a class of SSMQSs, thus making the relative stability analysis of the class of SSMQSs possible. Then in Chapter 4 we first identify two classes of SSMQSs and define the concept of degree of stability. For the special class of SSMQSs, we define the relative stability relations among the queues. Next we investigate the relative stability related properties in the special kind of SSMQSs. In Chapter 5, we perform both relative and absolute stability analysis to some practical systems. Through these examples, we demonstrate the unified and effective approaches to stability analysis of the SSMQSs. In addition, some other applications of the relative stability results will also be discussed in this chapter. These include the characterization problem of the system stability regions, the stabilization problem of the systems, and finding the maximum stable throughput of the systems. We conclude the dissertation in Chapter 6 and discuss some further directions of this research there.

2. BACKGROUND

Although stability is one of the fundamental issues in queueing systems, it is hard to find a universal definition to cover all its aspects. In this chapter, we first look at some common definitions of stability for both dynamic systems and stochastic systems. In particular, we concern about the stability definitions for queueing systems as a proper definition of queue stability is crucial to the definition of the concept of degree of stability. Then we briefly survey some methods that can be used to study stability problems in queueing systems. Finally, we discuss some existing results of stability analysis in SSMQSs.

2.1 Stability Definitions

2.1.1 Lyapunov Stability

A (deterministic) dynamic system is said stable usually means that the system has the good properties that it can be operated normally for a long time under certain conditions, and that the system's long run behaviour can be predicated. Qualitatively, the term *stability* describes the property that if starting the system at some desired operating point, it will stay close to the point in the equilibrium, and such behaviour is insensitive to the initial point as well as to the possible perturbations of inputs to the system. A stable system is the one which has such property.

The formalization of the stability concepts is probably originated by A. M. Lyapunov in studying nonlinear control systems [48]. Suppose a nonlinear dynamic system can be represented by a differential equation

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \tag{2.1}$$

where $\mathbf{x}(t) \in \mathcal{D} \subset \mathbb{R}^n$ is the system state vector with domain $\mathcal{D}$, and $\mathbf{u}(t) \in \mathbb{R}^m$ represents the input vector, $m \leq n$, and $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a vector function. If f is continuous, for each measurable and locally bounded $\mathbf{u}(t)$ and each initial condition $x_0 = \mathbf{x}(0)$, a (not necessarily unique) solution to Eq. (2.1) exists. Denote the set of solutions $\varphi(\cdot)$ to Eq. (2.1) under the assumptions as $\mathbf{S}_{x_0}$. Then different levels of stability of the system at the origin can be defined as follows [61].

Definition 2.1.1 *A dynamic system represented by Eq. (2.1) is (Lyapunov) stable at the origin if for each $\varepsilon > 0$ there exists $\delta > 0$ such that for each x_0 with $\| x_0 \| < \delta$ and all the solutions $\varphi(\cdot) \in \mathbf{S}_{x_0}$ the following holds: $\varphi(\cdot)$ is right continuable for $t \geq 0$ and*

$$\| \varphi(t) \| < \varepsilon, \forall t \geq 0. \tag{2.2}$$

Definition 2.1.2 *A dynamic system represented by Eq. (2.1) is locally asymptotically stable at the origin if it is stable at the origin and if there exists $\delta_0 > 0$ such that for each x_0 with $\| x_0 \| < \delta_0$ and all the solutions $\varphi(\cdot) \in \mathbf{S}_{x_0}$ the following holds:*

$$\lim_{t \rightarrow +\infty} \| \varphi(t) \| = 0. \tag{2.3}$$

The origin is said to be globally asymptotically stable if δ_0 can be arbitrary large.

Definition 2.1.3 *A dynamic system represented by Eq. (2.1) is exponentially stable at the origin if it is locally asymptotically stable at the origin and if there exists α, β, and $\delta_1 > 0$ such that for each x_0 with $\| x_0 \| < \delta_1$ and all the solutions $\varphi(\cdot) \in \mathbf{S}_{x_0}$ the following holds:*

$$\| \varphi(t) \| \leq \alpha \| x_0 \| e^{-\beta t}, \forall t \geq 0. \tag{2.4}$$

In all the definitions above, the stability is in strong sense that the conditions are required to be held for all the solutions in $\mathbf{S}_{x_0}$. Follows are the interpretations of the above definitions:

- A system is Lyapunov stable if it starts *close enough* to the equilibrium (at the origin and $\parallel x_0 \parallel < \delta$), it will stay there *close enough* forever ($\parallel \varphi(t) \parallel < \varepsilon$).
- A system is asymptotically stable means that it not only stays close to the equilibrium but also *converges* to it eventually ($\lim_{t\to+\infty} \parallel \varphi(t) \parallel = 0$).
- A system is exponentially stable means that it not only converges but also converges *fast enough* (with bounded convergence rate $\parallel \varphi(t) \parallel \leq \alpha \parallel x_0 \parallel e^{-\beta t}$).

We can see that the definitions indeed reflect the meaning of the term "stability", i.e., a stable system in the long run will stay close in some equilibrium states, and such behaviour is insensitive to the initial conditions and possible perturbations of the inputs.

2.1.2 Stability of Stochastic Systems

When random factors are added, we have stochastic systems, and the meaning of stability changes. For general stochastic systems, instead of finding exact solutions to the system states, it is more realistic to focus on the probabilistic aspects of the systems, that is, the distribution of the system states in the equilibrium. For these systems, the stability can refer to the convergence of the state distributions to some proper limiting probability distributions, and the convergence is independent to the initial conditions of the system and possible input perturbations. Sometimes, such convergence is also called *ergodicity* because we may obtain the limiting probability distributions through a set of ergodic theorems.

In studies of statistical mechanics of dynamic systems, the term "ergodicity" means the existence of a time average of the system states, and the time average equals to the space (phase) average of the system states. Let $(\Omega, \mathcal{F}, \mu)$ be some measure space with points $\omega \in \Omega$, and $T : \Omega \to \Omega$ be a measure preserving transformation of the space, i.e., for each measurable set $B \in \mathcal{F}$, $\mu(T^{-1}(B)) = \mu(B)$. If the system starts at $\omega_0 \overset{\text{def}}{=} T^0\omega$ and let $T^{k+1} = T(T^k)$, then the trajectory of the system with respect to time will be $T^0\omega, T^1\omega, \ldots, T^n\omega, \ldots$. For any measurable function $f:(\Omega, \mathcal{F}) \to (\mathbb{R}, \mathcal{B}_R)$ we have the time average of the system states with respect to f along the trajectory as:

$$\frac{1}{n}\sum_{k=0}^{n-1} f(T^k\omega). \tag{2.5}$$

If f is integrable, then the well-known Birkhoff Ergodic Theorem states that Eq. (2.5) converges almost everywhere to an integrable limit function f^*,

$$\lim_{n\to\infty} \frac{1}{n}\sum_{k=0}^{n-1} f(T^k\omega) \overset{\text{a.e.}}{=} f^*(\omega), \tag{2.6}$$

and the function f^* is constant on the trajectory, i.e., $f^*(T^k\omega) = f^*(\omega)$ for any k. Furthermore, if T itself is ergodic (or called metrically transitive), i.e., $T^{-1}B = B$ implies either $\mu(\bar{B}) = 0$ or $\mu(B) = 0$, then the limit in Eq. (2.6) (time average) is constant and equals to $\mathbf{E}f(\omega) = \int_\Omega f(\omega)\mu(d\omega)$ (space average). Intuitively speaking, the ergodic theorem says that, in one run and if it is long enough, the system will visit all the states and the fraction of time the system will stay in one particular state is the same as the chance that the system is found in that state in any run.

Now let $f(T^n\omega)$ be an ergodic and stationary random sequence $X(n)$ defined on some probability space, where $X(n) = f(T^n\omega)$, then Eq. (2.6) can be rewritten as:

$$\lim_{n\to\infty} \frac{1}{n}\sum_{k=0}^{n-1} X(k) \overset{\text{a.s.}}{=} \mathbf{E}f(\omega). \tag{2.7}$$

Denote π as the distribution of $X(0) = f(\omega)$, then for each measurable function g, its mean with respect to X will be

$$\lim_{n\to\infty} \frac{1}{n}\sum_{k=0}^{n-1} g(X(k)) \overset{\text{a.s.}}{=} \int g(x)\pi(dx). \tag{2.8}$$

If $g(x) = I_A(x)$, the indicator function of a set A, then the left-hand side of Eq. (2.8) is the fraction of time that $X(k)$ spent in A, and this fraction, according to Eq. (2.8), is independent to the initial state $X(0)$ and converges to $\pi(A)$. The constant right-hand side $\pi(A)$ is the limiting distribution we are looking for. Therefore, the stability of a stochastic systems, i.e., the existence of a limiting probability distribution of the system states, can be considered as the results of applying ergodic theory to stochastic processes [12].

Specifically, if $X(n)$ is a Markovian process, then it can be proved that π is the solution to the equation,

$$\pi(A) = \int \pi(dx)P(x,A), \tag{2.9}$$

where $P(x,A) = \mathbf{P}\{X(1) \in A | X(0) = x\}$ is the transition probability. The process will also be called ergodic if π, the solution to Eq. (2.9), is a probability distribution, i.e., $\int \pi(dx) = 1$. The meaning of Eq. (2.9) is that if we start the process at $X(0)$ with distribution π, the distribution that the process will be at a state in the set A at $X(1)$ will have the exact distribution as π.

2.1.3 Stability of Markov Chains

Now we consider definitions of stability of discrete time Markov chains, which usually are used in modeling queueing systems. An (embedded) discrete time Markov chain (DTMC) is a collection of random variables $\mathbf{X} = \{X_n : n \in \mathbb{Z}_+\}$ with state space $\mathcal{X}$ and the associated countably generated σ-field $\mathcal{B}$ satisfies the following Markovian property

$$P(X_{n+m} \in A | X_m = x, X_l, l < m) = P(X_{n+m} \in A | X_m = x) = P_m^n(x,A). \tag{2.10}$$

That is, the process $\mathbf{X}$ is *memoryless* for all but its most immediate past. The transition probability $P_m^n(x,A)$ represents the probability that started from state x at the mth step, the chain will be in the set A after n step of transitions, where $A \subset \mathcal{B}$. The DTMC with transition probability defined in Eq. (2.10) is called *homogeneous* if $P^n(x,A)$ is independent to m, i.e., $P_m^n(x,A) = P_0^n(x,A)$ for all m. Define the hitting time of the set A as

$$\tau_A = \inf(n \geq 1 : X_n \in A), \tag{2.11}$$

i.e., the first time the chain reaches (or returns to) the set A. If the state space $\mathcal{X}$ is countable, a chain is called *irreducible* if for any nonempty set $A \subset \mathcal{B}$ and any initial state x, the following holds

$$P(\tau_A < \infty | X_0 = x) > 0. \tag{2.12}$$

The irreducibility of a chain means that every state is reachable from any other states. Furthermore, a chain is called *recurrent* for any nonempty set $A \subset \mathcal{B}$ and any initial state $x \in A$ if

$$P(\tau_A < \infty | X_0 = x) = 1. \tag{2.13}$$

Otherwise, the chain is called *transient*. It can be proved that an irreducible chain with countable state space is either recurrent or transient. A chain is recurrent means that any state can be visited infinitely many times. For an irreducible and recurrent chain, if the mean of the hitting time for any set A exists and is finite, i.e., $E[\tau_A | X_0 = x] < \infty$, the chain is called *positive recurrent*; otherwise, it is called *null recurrent*. Furthermore, let $d(x)$ be the period of state x where $d(x) = \text{gcd}\{n \geq 1 : P^n(x, x) > 0\}$, then a DTMC is called *aperiodic* if for each state x the period $d(x) \equiv 1$.

For a stochastic system which can be modeled as a DTMC, its stability is ensured if the underlying DTMC is homogeneous, aperiodic, irreducible, and positive recurrent, or collectively, *ergodic*. In other words, an ergodic DTMC admits a unique invariant distribution (also called the stationary distribution) of the system states satisfies Eq. (2.9). An important result for ergodic DTMC is that the transition probability of the chain converges in total variation to its stationary distribution:

$$\parallel P^n(x, \cdot) - \pi(\cdot) \parallel \longrightarrow 0. \tag{2.14}$$

This allows an even stronger sense of stability to be defined: geometric ergodicity. A DTMC is called geometrically ergodic if there exists a constant $r > 1$ such that

$$\sum_{n=1}^{\infty} r^n \parallel P^n(x, \cdot) - \pi(\cdot) \parallel < \infty, \tag{2.15}$$

where the constant r^{-1} is the rate of convergence.

If a DTMC with transition probability law $P^n(x, A)$ evolves on a general state space $\mathcal{X}$, we require the set A to have some *reasonable* size. For this case, a more general concept called φ-irreducibility is defined. A chain is called φ-irreducible if for any $A \subset \mathcal{B}$ and initial state x, there exists a measure φ on $\mathcal{B}$ such that

$$\varphi(A) > 0 \Longrightarrow P(\tau_A < \infty | X_0 = x) > 0. \tag{2.16}$$

Comparing Eqs. (2.12) and (2.16), it can see that the irreducibility in the countable state space is a special case of the φ-irreducibility by taking φ as the counting measure of a set. Furthermore, a set A is called *Harris recurrent* if Eq. (2.13) holds for all $x \in A$. The chain itself is Harris recurrent if the state space only contains Harris recurrent subsets. Parallel to the countable state space case, if there exists an invariant probability distribution π on $\mathcal{X}$ satisfies Eq. (2.9), the chain is called positive recurrent; otherwise it is called null. Therefore, we can see that a DTMC with a general state space is stable if it is homogeneous, aperiodic, φ-irreducible, and Harris positive recurrent. A rigorous treatment to the stability of Markov chains can be found in [51].

2.1.4 Stability in Queueing Systems

In queueing systems, the queue length processes of all the queues at some time can often be modeled as stochastic processes, e.g., DTMCs. Based on the above discussion, it is natural to define the stability of a queueing system as the convergence of the queue length processes to some limiting distributions. The convergence implies that the queue lengths will be finite or even empty infinitely many times in the equilibrium with probability one, i.e., the queues will not explode to the infinity. Besides the queue length processes, there are other considerations of the system states, for instances, the waiting time (delay) processes of the customers, or the remaining service time at the servers. It is easy to see that a queueing system is stable in terms of finite queue length implies finite waiting time for any customer and finite remaining service time for the server.

Let Q^n be the queue length process at some discrete time epoch n, the stability of a queue can be defined as the *existence of a limiting distribution function* of the queue length process [45, 64].

Definition 2.1.4 *A queue is stable if the distribution of the queue length process Q^n converges to some limiting distribution function $F(x)$ as n tends to infinity:*

$$\lim_{n\to\infty} P(Q^n < x) = F(x), \ \textit{and} \ \lim_{x\to\infty} F(x) = 1. \tag{2.17}$$

A weaker version of the above is called *substable* if the queue length process is only *bounded in probability*, namely,

$$\lim_{x\to\infty} \liminf_{n\to\infty} P(Q^n < x) = 1. \tag{2.18}$$

A stable system is of course substable, while a substable system becomes stable when the queue length process also tends to a limit. If a system is neither stable nor substable, it is unstable. The above definitions also apply to a multidimensional queue length process if there are multiple queues in the system, i.e., Q^n is a vector and the limiting distribution function becomes the joint distribution of the queue length processes.

A special case is that if a queueing system can be modeled as a (multidimensional) DTMC, the stability and substability of the system imply each other because they both are equivalent to the *ergodicity* (positive recurrence) of the chain [13, 19, 51, 64]. Besides, *geometric ergodicity* can also be considered [5, 73, 74]. Another way to define stability of queues is to use the *existence of finite moments* [31, 63, 75], that is, a queue is stable if the lth-moment of the queue length exists and is finite. It can be seen that the existence of finite moments is stronger than the existence of limiting distribution.

All of the above mentioned definitions of stability of queueing systems assume that the arrival processes to the systems are some stochastic processes. On the other hand, if a queueing system is fed by some deterministic input processes such as the ones proposed in [20], i.e., the total arrivals during a period is bounded above and the bound only depends on the length of the period, then the stability of the queues in the system can be defined as the *existence of deterministic bounds* for the interested quantities such as queue length or customer delay [14]. Another stability definition of deterministic queueing systems appears in the sample path analysis of queueing systems [52]. On a sample path, if the average arrival rate to a queue as well as the average departure rate from the queue converge and equal to each other, the queue is called *rate stable.* In the following table, we summarize the stability definitions mentioned in this section.

Dynamic Systems:	Lyapunov Stability Asymptotic Stability Exponential Stability
Stochastic Systems:	Existence of Limiting Distribution
Markov Chains:	Ergodicity Geometrical Stability Moment Stability
Stochastic Queueing Systems:	Substability (bounded in probability) Stability (existence of limiting distribution)
Deterministic Queueing Systems:	Rate Stability & Performance Bounds

Table 2.1
Common stability definitions.

2.2 Methods of Studying Stability

2.2.1 Lyapunov Function Methods

To establish stability for (deterministic) dynamic systems, Lyapunov proposed two methods and the more common one is the *second* method or the *direct* method.

Consider a dynamic system that can be described by Eq. (2.1), the second method says that if there exists a nonnegative function $V(x)$ of the system state x such that

$$\begin{aligned} V(\bar{x}) &= 0, \\ V(x) &> 0 \text{ for any } x \neq \bar{x}, \\ \dot{V}(x) &< 0, \end{aligned}$$

then $\bar{x}$ is asymptotically stable. The function $V(x)$ is often called the *test function*. Intuitively, if considering the nonnegative function $V(x)$ as the energy or potential level of the system at state x, the existence of such a test function (with negative accelerated rate) implies the energy level of the system has a way to continuously decrease until it reaches the equilibrium $\bar{x}$ and stays there forever. For the direct method, however, there is no systematic way for finding suitable test functions for a given system.

Borrowing the idea from the Lyapunov direct method, test function methods have been established to study the stability of stochastic systems, especially the Markovian processes [28, 51]. Consider an irreducible DTMC on a countable state space $\mathcal{X}$ with one-step transition probability matrix $\mathbf{P}$ and state $X(t)$ at time t. Let V be a nonnegative function on $\mathcal{X}$, served as the test function, define the drift function $d(i)$ at state i as following

$$d(i) = \mathbf{E}[V(X(t+1)) - V(X(t))|X(t) = i] = \mathbf{P}V - V. \tag{2.19}$$

Intuitively, the drift function represents the average difference of the energy between this state and the next state. If the drift function is always negative, i.e., corresponding to the negative accelerated rate, we can then conclude the DTMC is stable. The following theorem, which often referred to as the *drift criteria*, summarizes the above idea and can be served as the criteria for the DTMC to be stable [51, 74].

Theorem 2.2.1 *If there exists a function $V : \mathcal{X} \longrightarrow \mathbb{R}^+$ and a finite subset C of $\mathcal{X}$ such that:*

(a) *If $\{i : V(i) \leq K\}$ is finite for all K, and if $\mathbf{P}V - V \leq 0$ on $\mathcal{X} - C$, then X is recurrent.*

(b) *If for $\varepsilon > 0$ and b is a constant such that $\mathbf{P}V - V \leq -\varepsilon + b\mathbb{I}_C$, then X is positive recurrent.*

The meaning of the above theorem can be interpreted as follows. In (a), for any initial state in $\mathcal{X} - C$, the condition $\mathbf{P}V - V \leq 0$ ensures that the value of the test function of the next state is finite. Then because the set $\{i : V(i) \leq K\}$ is finite for all K, the next step can only transit to a finite number of states. The irreducibility assumption of the chain implies there must be some states that belong to the set C in those finite number of states. Therefore, the returning time of the chain to the set C is finite with probability 1, i.e., the chain is recurrent. In (b), for any initial state in $\mathcal{X} - C$, the conditions that $V(\cdot)$ is nonnegative and $\mathbf{P}V - V \leq -\varepsilon$ implies the returning time of the chain to the set C has a finite moment. Once the chain enters C, the *recharging* of $V(\cdot)$ is bounded by $\mathbf{P}V - V \leq -\varepsilon + b$. This implies that if the chain ever leaves C again, it will return to C again within a finite period. Therefore, the chain is positive recurrent, i.e., ergodic.

Following theorems further give the criteria for an irreducible DTMC to be geometrically ergodic and have finite moments, respectively [51, 74].

Theorem 2.2.2 *An irreducible DTMC chain is geometrically ergodic if for $\varepsilon > 0$ and a constant b, there exists a test function $V(x) \geq 1$ for $x \in C$ such that,*

$$\mathbf{P}V - V \leq -\varepsilon V + b\mathbb{I}_C.$$

Theorem 2.2.3 *An irreducible DTMC chain has finite moments with respect to a nonnegative function $f(\cdot)$ if for $\varepsilon > 0$ and a constant b, there exists a test function such that,*

- $\mathbf{P}V - V \leq -\varepsilon f + b\mathbb{I}_C$,

- *if $V(x) \geq f(x)$ for $x \in \mathcal{X} - C$ and $\sum_{x \in C} \pi_x f(x) < \infty$, then $\sum_{x \in \mathcal{X}} \pi_x f(x) < \infty$,* *where π_x is the stationary distribution of the chain.*

Same as the Lyapunov direct method, one of the major difficulties to use the test function methods to establish stability of DTMCs is that it is hard to find suitable test functions for a given DTMC. Nevertheless, the test function methods are still the most general methods to study stability of queueing systems that can be modeled as DTMCs.

2.2.2 Fluid Model Approach

Fluid model is an important approach mainly developed in the last two decades for establishing stability conditions of a class of queueing models called the open multiclass queueing networks (OMQN) [18, 21, 22, 62]. The idea of the fluid model approach can be outlined as following. In general, fluid models are deterministic and continuous approximations of the underlying discrete stochastic networks. The approximation is done through replacing the stochastically arrived and moved discrete packets in the network with deterministic and continuous fluids. The rates of the fluids are the average rates of the corresponding stochastic quantities. For a given initial state of the stochastic queueing network under consideration, with proper time and space scaling, if one can prove (through the Functional Strong Law of Large Numbers) all the sample paths converge, where the limit is called the fluid limit of the original network, then the stability of the fluid model implies the stability of the original stochastic network.

More precisely, the dynamic of an open multiclass queueing networks in general can be formulated as the follows:

$$\begin{aligned}
Q(t) &= Q(0) + E(t) + \sum_{k=1}^{K} \Phi^k(S_k(T_k(t))) - S(T(t)) \quad \text{for } t \geq 0i, && (2.20)\\
Q(t) &\geq 0 \quad \text{for } t \geq 0, && (2.21)\\
T(0) &= 0 \text{ and } T_k(\cdot) \text{ is nondecreasing for } 1 \leq k \leq K. && (2.22)\\
U_i(t) &= t - \sum_k T_k(t) \text{ is nondecreasing for each station i.} && (2.23)
\end{aligned}$$

In the above, there are K queues and $Q(t)$ is the queue length at time t, $E(t)$ is the accumulative arrival to the system up to time t, $S(T(t))$ is the number of customers that have been served up to time t if the server totally devoted $T(t)$ among of time to serve the customers at time t, $\Phi(t)$ represents the internal routing functions of the customers, and $U(t)$ is the total idle time up to time t. If the arrival processes, the service time processes, and the routing processes satisfy the functional strong law of large numbers, then the above model has a fluid limit, which can also by represented by a set of equations similar to the above. A fluid model is stable if there exists $t_0 > 0$ such that for any fluid model solution, $\hat{Q}(t) = 0$ for all $t \geq t_0|\hat{Q}(0)|$. To prove a fluid model is stable, Lyapunov functions are constructed and then with the following theorem [21], one can obtain the stability for the original OMQN. In this regard, fluid model can be considered as an intermediate step of the test function methods to study stability of complicated queueing models.

Theorem 2.2.4 *Fix an open multiclass head-of-line queueing network and consider "the" associated fluid model. Suppose that the interarrival times have unbounded support and satisfy a "spread-out" assumption. If the fluid model is stable, then a Markov process describing the queueing network is positive Harris recurrent.*

2.2.3 Non-Markovian Analysis Methods

In the literature, there are some non-Markovian analysis (non-test function) methods proposed to study stability in SSMQS, even though some of the models may be able to represented by DTMCs [1, 27, 29, 49, 57, 64]. The advantages of these approaches are able to avoid the difficulties in finding suitable test functions, or in considering more general input traffic models such as stationary and ergodic marked point processes.

In [57] and [64], a dominant system method is used to study the ALOHA type communication networks and polling systems. In both works, the models under consideration can be represented by multidimensional DTMCs. Let $\mathbf{Q}^n$ be such a DTMC with components Q_i^n representing the queue length at each individual queue, and n is the Markovian epoch. In general, the components Q_i^n are not Markovian, and it is not easy to find a test function for the multidimensional $\mathbf{Q}^n$. However, by noticing the facts that for DTMC the substability (Eq. (2.18)) of the chain is equivalent to the stability of the chain, the following two *isolation* theorems allow one to transform the multiple queue stability problem into single queue stability problem [63].

Theorem 2.2.5 *A k-dimensional DTMC $\mathbf{Q}^n = (Q_1^n, Q_2^n, ..., Q_k^n)$ is substable if all the one-dimensional processes Q_i^n are stable.*

Theorem 2.2.6 *If any one-dimensional process Q_i^n is unstable, the k-dimensional DTMC $\mathbf{Q}^n$ is also unstable.*

Then, to obtain the stability of each single queue Q_i^n, the famous Loynes' theorem [45] is used. Loynes' theorem specifies stability conditions for a single G/G/1 queue.

Theorem 2.2.7 *For a single server queue, let $\{A^n\}$ be the interarrival times and $\{S^n\}$ be the services times. If the pair $\{A^n, S^n\}$ is a strictly stationary and ergodic process, the following holds:*

i if $EA < ES$, the G/G/1 queue is stable in the sense of Definition 2.1.4,

ii if $EA > ES$, the G/G/1 queue is unstable,

iii if $EA = ES$ the queue may be stable, substable, or unstable. If $\{A^n\}$ and $\{S^n\}$ are independent to each other, and one of them is formed of non-constant mutually independent random variables, then the queue is unstable.

At this point, a subtle technique used in the method is to construct dominant systems to ensure that the stationary and ergodic requirements of the arrival and service time processes of the individual queues are satisfied in the multidimensional environment. The idea of the construction is to split the queues into two subsets, namely, the stable ones and the persistent ones. For the stable ones, they behave the same as in the original system, and because they are stable, the multidimensional DTMC of this subset of queues are ergodic, which means their contributions to the system state are stationary and ergodic. For the persistent ones, they are assumed never empty, i.e., by sending *dummy* traffic if necessary. The effect of this assumption is to put the persistent queues into their worst situations and such situations are predicable. In other words, under this assumption, together with certain scheduling algorithms and service policies the server employs, the persistent queues also contribute stationary and ergodic components to the system state. Then, for any persistent queue, we can apply Loynes' theorem directly to obtain its stability condition. It can be prove that the stability of a queue in the modified system implies the stability of the queue in the original system, thus the meaning of the dominant systems. Finally, by considering each queue as one of the persistent queues and intersect the results, the stability of the original system is achieved. The dominant system approach has been used to study the stability of a variety of polling and random assess systems [15, 16, 23, 32–34, 47, 53, 59, 65].

In [29] a polling model with general service policies is studied. The approach to obtain stability conditions for the system is based on a monotonic property of the DTMC representation of the system. Specifically, the property states that if the initial state of the system is empty, then the system queue length process (which is represented by a multidimensional DTMC) will monotonically increase in distribution (i.e., stochastically increase) until the steady state is reached. To achieve the stability, dominant systems have also been constructed to eliminate some of the queues' random effects, i.e., let those queues' contributions to the system become constants. Then the mathematical induction is used to prove the stability of the system by changing a persistent queue to a normal queue one at a time. It is worth to mention that the purpose of the dominant systems in this approach is mainly to eliminate some of the queues' random effects, while in [57] and [64] the dominant systems in addition be utilized to achieve the stationarity and ergodicity of some queues' contributions to the system state, though both have the same dominant system meaning, i.e., the stability in the dominant system implies the stability in the original system.

In [1, 27, 49], multiple queue systems with stationary and ergodic marked point processes are studied. The techniques used in [1] and [49] to prove stability are the Loynes' construction of stationary system regime and Palm probability, while in [27], the *saturation rule* and the concept of the *maximal dater* have been introduced. All these approaches may have difficulties in constructing the stationary regime through Loynes' backward method, thus the monotonic and contractive [44] properties of the service policies are essential to these methods.

Besides the aforementioned methods, for stability of the deterministic queueing systems, there are adversarial queueing theory [11], network calculus method [14], and the sample-path method [52]. Discussion of methods to studying stability in stochastic models can also be found in [24, 64].

2.3 Stability Results for Multiple Queue Systems

In this section we briefly review some published works related to this research. The focus will be given to the stability analysis of SSMQSs, though results for some others models will also be mentioned.

The concept of relative stability is common in classical control theory [8, 54, 55]. Associated with the relative stability, the degree of stability of a control system can determine how large a perturbation is required to produce an unstable system, and it can be measured by either the *phase margin* or the *gain margin.* To determine the phase margin or the gain margin, a commonly used technique is the Nyquist stability criterion [56]. Besides in control systems, the concepts of relative stability and degree of stability have also been borrowed and applied in different kinds of networking problems. Khotimsky and Krishnan utilize the relative stability to compare the *degree of congestion* of switching planes in a parallel packet switch architecture [39]. In [37], the authors use a nonlinear dynamic model of TCP to analyze and design active queue management control systems with random early detection scheme. One of the system design goal is to maintain the systems within stability margins. In [36], an ATM traffic management model has been formulated as a feedback control system and the relative stability of the system is measured by the phase margin. In the area of stability analysis of multiple queue systems, there are some results of *stability rank* or *stability ordering*, which specifies the ranks or ordering of the queues becoming unstable, have been reported [16, 17, 29, 47]. Specially, a concept of the *least stable queue* has been introduced to obtain system stability of a version of polling system with applications in the satellite communications [15]. It is not hard to find the least stable queue holds one of the end positions in the stability ordering of the queues while the other end can be called as the *most stable queue.* The stability ordering in [29] is obtained through identifying the individual queue whose stability implies the stability of a subset of queues (local stability). In [47] the stability rank is obtained through comparison of each queue's probability of non-empty and no queues transmit (including itself) during a slot in the ALOHA network. The way to find out the stability ordering in [16, 17] is more intuitive, that is, the works first provide conditions under which any two queues are as stable as each other. Then the conditions of the other two relations (one queue is more or less stable than the other) are identified. All these conditions can be represented by some relations of the queues' arrival rates, and once the condition of the as stable as relation is known, the other two can then be obtained straightforwardly. For the usage, the stability ordering is essential when studying the queue stability issues in multiple queue systems [16, 17] because any single queue in general will be at each position in the ordering some time. Furthermore, the stability ordering can be used in construction of dominant systems such that the stables queues are known to any specific dominant system. This technique has been used in [29, 47, 59] to achieve local stability or tighter bounds of the stability regions of the systems. Nevertheless, to the best of our knowledge, this research is the first attempt to provide a general relative stability study of the SSMQSs.

For the absolute stability of SSMQSs, most of the attention has been given to the system stability analysis of two common types of models, namely, the polling systems [35, 66–69, 76] and random access networks [3, 9]. For polling systems, works worth to mention are [4–6, 15, 16, 25–27, 29, 30, 32–34, 34, 38, 40, 41, 49, 58]. Among them, almost all the works consider stochastic models of polling systems except [4], in which deterministic model is used. For the stability definition, most of the works consider only the existence of a stationary distribution of the queue length processes, while [5, 6] in addition consider the geometric ergodicity and moment stability, respectively. However, in [4], stability means the existence of finite bounds of packet delay. Moreover, system stability is the main concern of all the works, while local stability and queue stability have also been achieved in [16, 29, 38]. The main approaches used in the above works are summarized as follows: test functions in [5, 6, 40], dominant systems approach in [32–34], stationary marked point processes in [26, 27, 49, 58], monotonicity of Markov chain in [29, 30], the least stable queue in [15, 41], stability ordering in [16], queue backlog in [38], network calculus in [4]. The service policies that have been covered in the works including both unlimited type, such as pure exhaustive and gated policy, and limited type, such as gated-limited or time-limited, or even mixed type. Also, periodic polling, polling table, and Markovian polling have been covered. Specially, polling system with multiple server is considered in [30], while state dependent set-up time and routing have been considered in [15, 16] and [26], respectively.

In a random access system the single server serves the queues in a random (or probabilistic) fashion. The most well-known example is the ALOHA network and its variants [3]. Though the scheme is simple enough, for the asymmetric case, the exact and computable system stability boundary of an ALOHA network can be solved only when the system consists of two stations, though there is an exception in [7] in which a special assumption of the arrival processes is used. Under this special assumption, the overall system stability region of the ALOHA network can be characterized. In general, when there are more stations in the system, only inner and outer bounds can be obtained [17, 47, 57, 65, 72]. Besides the collision channel in the works cited above, recently, there are works of stability analysis for random access networks with multipacket reception [46, 53], broadcast random access [59], and wireless networks with retransmission diversity [23]. Specifically, [46] confirms the overall system stability region of the ALOHA network obtained in [7] through a *sensitivity monotonicity* conjecture of the model. In all of these works, the major concern of the stability is the existence of stationary distribution of the queue length processes, while the main approach is the dominant system method. Exception is that in [23] the dominant system approach is only used to derive sufficient stability for the NDMA and BNDMA retransmission scheme, while for the sufficient and necessary conditions, both test function and network calculus methods are used. Furthermore, queue stability has also been considered in [17].

In communication networks, both polling systems and random access systems can be used to represented single-hop networks. In practice, these single-hop networks will be interconnected to form multihop networks. Multihop radio networks with station activation constrains is considered in [70]. In the findings, the stability region of the optimal service policy is the superset of the stability regions of all possible service policies. And the optimal service policy tends to equalize the queue length differences among the queues. In [10], a version of interconnected single-hop random access network had been analyzed and the system stability had been obtained. Some other works that worth to mention are: [42, 43], in which fluid limit approach is used to study stability for open queueing models and scheduling policies; [1, 2], in which stationary marked point process is used to study stability for processor-sharing systems; [14, 77], in which stability of networks with deterministic inputs have been studied; [50], in which network backlog approach is used to obtain sufficient stability conditions for some wireless networks.

2.4 Summary

In this chapter we first reviewed some general definitions of stability of stochastic and queueing systems. Then we described the commonly used approaches in stability analysis of those systems. Finally, we briefly introduced some existing stability results in SSMQSs. In the rest of this dissertation, we adopt the stability definitions as Eq. (2.17) and Eq. (2.18). The approach we are using in this study is mainly the non-Markovian analysis. Specifically, we will analyze stability of SSMQSs from a relative stability perspective. The comparison between our approach and the dominant system approach will be provided in Chapter 5 after some demonstrations of our approach are given.

3. RELATIVE STABILITY RELATION IN SINGLE-SERVER-MULTIPLE-QUEUE SYSTEMS

3.1 Introduction

In this chapter we study the relative stability relations among the queues in some SSMQSs. By relative stability relation we mean whether any two queues' stability status can somehow be compared. The purpose of this chapter is to show that such kind of relations commonly exist in some SSMQSs and the relations have connections to the traffic inputs of the queues. To achieve the goal, we select three SSMQSs to study, and the SSMQSs are: a polling system with gated limited service policy, a slotted buffered ALOHA network, and a processor sharing system. The results in this chapter can be considered as a prelude to the more general relative stability results for the SSMQSs.

In this chapter and the sequels, we consider stability in the sense of Eqs. (2.17) and (2.18). That is, a queueing system or an individual queue are called stable if the distributions of the queue length processes converge to some limiting distributions; and the system or the queue are called substable if the limiting queue lengths are finite with probability 1. Consequently, a queue is called unstable if the following holds

$$\lim_{x\to\infty}\limsup_{n\to\infty} P(Q^n \geq x) > 0. \tag{3.1}$$

3.2 A Polling System with Gated Limited Service

For simplicity, we consider in this section only a basic version of the polling systems which is the same as the one studied in [32]. The polling system consists of a single server and a finite set of k queues, denoted as q_i, $i \in \{1, ..., k\}$. Each q_i has infinite buffer to store incoming customers. When attached at q_i, the server serves a certain number of customers that already appeared at q_i upon the server's arrival, up to M_i customers and thus the gated limited service. Each q_i has a Poisson arrival with rate λ_i, and $A_i(t)$, $t \geq 0$, is the total arrivals at q_i up to time t. The service time process at q_i, B_i, is generally distributed i.i.d. with finite first moment b_i. A switch-over time, U_i, is spent by the server between its visit finishes at q_i and service starts at q_{i+1}. The switch-over time process is also generally distributed i.i.d. with finite first moment u_i. All the arrival processes, service time processes, and switch-over time processes are assumed mutually independent to one another. The server visits the queues in a cyclic order. A cycle is the successive arrivals of the server at a particular queue. Without loss of generality, we let q_1 be that particular queue. Denote u_0 as the average total switch-over time during a cycle, where $u_0 = \sum_{i=1}^{k} u_i$. Further denote ρ_i as the server utilization at q_i and ρ as the total server utilization, where $\rho_i = \lambda_i b_i$, and $\rho = \sum_{i=1}^{k} \rho_i$.

Let $Q_i^n(1)$ be the number of customers at q_i when the server visits q_1 for the nth time. Each $Q_i^n(1)$ evolves according to the following:

$$Q_i^{n+1}(1) = [Q_i^n(1) + A_i^{1,n} - X_i^n]^+ + A_i^{2,n}, \tag{3.2}$$

where $A_i^{1,n}$ is the number of customers arrived at q_i during the period of the server's arrival at q_1 and its arrival at q_i in the nth cycle, $A_i^{2,n}$ is the number of customers arrived at q_i during the period of the server's arrival at q_i in the nth cycle and its arrival at q_1 in the $(n+1)$th cycle, X^n is the number of served customers at q_i during the nth cycle, and $[x]^+ = \max(x, 0)$. Note that if q_i is q_1, then $A_i^{1,n} = 0$. It can be proved that the joint queue length process $\{\mathbf{Q}^n(1) = (Q_1^n(1), Q_2^n(1), ..., Q_k^n(1))\}_{n=1}^{\infty}$ is a homogeneous, irreducible, and aperiodic Markov chain [32], and we state it as a lemma in the following.

Lemma 3.2.1 *The joint queue length process* $\mathbf{Q}^n(1)$ *described above is a homogeneous, irreducible, and aperiodic Markov chain.*

Another useful result regarding to our polling model is that if the system is stable, the long run average of the server's cycle time and the number of customers served at a queue during a cycle exist and unique, and the two quantities satisfy a balance equation [32]. We state this result in the following.

Lemma 3.2.2 *Let the Markov chain $\mathbf{Q}^n(1)$ be positive recurrent (ergodic), then the expectation of the cycle time $\mathbf{E}C$ and the expectation number of customers served at each q_i during any cycle $\mathbf{E}X_i$ exist and are unique. Furthermore, we have*

$$\mathbf{E}C = \frac{u_0}{1 - \sum_{i=1}^{k} \rho_i}, \tag{3.3}$$

and for each q_i,

$$\mathbf{E}X_i = \lambda_i \mathbf{E}C. \tag{3.4}$$

In the following we show that the assumption of Lemma 3.2.2 can be relaxed to an unstable system. That is, even there are some queues in the polling system are unstable, the conclusion of Lemma 3.2.2 still holds, i.e., the mean cycle time of the polling system exists and is unique.

Lemma 3.2.3 *Assume that under a certain arrival traffic pattern $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_k)$, there is a partition of the queues in the polling system, i.e., $\{q_1, q_2, ..., q_k\} = \mathcal{S} \cup \mathcal{U}$, such that all the queues in $\mathcal{S}$ are stable while all the queues in $\mathcal{U}$ are unstable in the steady state. Then the expectation of the cycle time $\mathbf{E}C$ and the expected number of customers served at $q_i \in \mathcal{S}$ during any cycle $\mathbf{E}X_i$ exist and are unique. Furthermore, we have*

$$\mathbf{E}C = \frac{u_0 + \sum_{q_i \in \mathcal{U}} M_i b_i}{1 - \sum_{q_i \in \mathcal{S}}^{k} \rho_i}, \tag{3.5}$$

and for each $q_i \in \mathcal{S}$,

$$\mathbf{E}X_i = \lambda_i \mathbf{E}C. \tag{3.6}$$

Proof: If $\mathcal{U} = \emptyset$, the lemma is the same as Lemma 3.2.2. Now assume $\mathcal{U} \neq \emptyset$ and there is at least $q_j \in \mathcal{U}$. This implies that, based on the definition of substability (Definition 2.18), we have

$$\lim_{x \to \infty} \liminf_{n \to \infty} P(Q_j^n(1) < x) < 1,$$

which is equivalent to

$$\lim_{x\to\infty} \limsup_{n\to\infty} P(Q_j^n(1) \geq x) > 0. \tag{3.7}$$

Consider random variable $\tau_j = \sup\{n : Q_j^n(1) \leq M_j\}$, the last cycle when the server visits q_1 and the queue length at q_j is less than or equal to M_j. We claim that $P(\tau_j < \infty) = 1$. Otherwise, it would contradict with Eq. (3.7). In fact, if that is not the case we should then have $P(\tau_j < \infty) < 1$, which is equivalent to $P(\tau_j = \infty) > 0$. In other words, there exists sample path of the system such that $P(Q_j^n(1) \leq M_j) = 1$. Then on the sample path we must have

$$\lim_{x\to\infty} \limsup_{n\to\infty} P(Q_j^n(1) \geq x) = 0,$$

which contradicts with the assumption that $q_j \in \mathcal{U}$ (Eq. (3.7)).

Similarly, for all other q_js in $\mathcal{U}$ we have the same conclusion regarding to the cycle number τ_js. Let $\tau = \sup(\tau_j, \forall q_j \in \mathcal{U})$, we also have $P(\tau < \infty) = 1$. Then for each cycle $n > \tau$, when the server visits q_1, at any $q_j \in \mathcal{U}$, we have $P(Q_j^n(1) > M_j) = 1$, and the server always serves M_j customers during the cycle. Because the service times are i.i.d. and are independent to other processes in the system, the time that the server spends at q_j and the subsequent switch-over time will be equal to $\sum_{l=1}^{M_j} B_j^{n,l} + U_j^n$ with mean $M_j b_j + u_j$, where $B_j^{n,l}$ is the service time for the $l \leq M_j$ customer during the nth cycle at q_j.

Now consider queues in $\mathcal{S}$ and let $\mathbf{Q}_{\mathcal{S}}^n(1)$ be the queue length process vector for the stable queues. Define random variable $K^0 = \inf\{n > \tau : \mathbf{Q}_{\mathcal{S}}^n(1) = \mathbf{0}\}$, the first cycle after τ such that all the queues in $\mathcal{S}$ are empty when the server visits q_1. Further define random variables $K^{n+1} = \inf\{n > K^n : \mathbf{Q}_{\mathcal{S}}^n(1) = \mathbf{0}\}$, $R^0 = K^0$, and $R^{n+1} = K^{n+1} – K^n$. Because all the queues in $\mathcal{S}$ are assumed stable, the $|\mathcal{S}|$ dimensional Markov chain $\mathbf{Q}_{\mathcal{S}}^n(1)$ is ergodic, therefore, $P(K^0 < \infty) = 1$ and consequently $P(R^0 < \infty) = 1$. Furthermore, $\mathbf{Q}_{\mathcal{S}}^n(1)$ is regenerative with respect to R^n, and R^n defines a delayed renewal process with R^0 as delay and $\mathbf{E}(R) < \infty$. The cycle time $C_{\mathcal{S}}^n$ is also regenerative with respect to R^n. In fact, we can consider the time that the server spends at an unstable queue and the subsequent switch-over time as part of the server's switch-over time from a stable queue to another stable queue. Then from Lemma 3.2.2 we have Eq. (3.5) and Eq. (3.6). Finally, it is easy to see that Eq. (3.5) is also true when $\mathcal{S}=\emptyset$. □

Next we present the main result in this section: the relative stability relation of the queues in the polling system.

Theorem 3.2.1 *For any two queues and a given arrival rate vector of the queues* $\Lambda=(\lambda_1,\lambda_2,...,\lambda_k)$ *in the polling system, assume* $\frac{\lambda_i}{M_i}\leq\frac{\lambda_j}{M_j}$*, then*

(a) q_j*'s stability implies* q_i*'s stability;*

(b) q_i*'s instability implies* q_j*'s instability.*

Proof: For the first part of the theorem, assume that q_j is stable but q_i is unstable. Then all the queues can be partitioned into $\mathcal{S}$ and $\mathcal{U}$, and we have at least $q_j\in\mathcal{S}$ and $q_i\in\mathcal{U}$. As discussed in Lemma 3.2.3, after the τth cycle all the queues in the set $\mathcal{U}$ will contribute a constant number of customers during a cycle, where τ is the last cycle that the queue length at all $q_i\in\mathcal{U}$ is less than or equal to M_i, $P(\tau<\infty)=1$. Furthermore, it can be proved that $\{\mathbf{Q}_{\mathcal{S}}^{n+1}(1),C_{\mathcal{S}}^{n}\}_{n>\tau}^{\infty}$ is an ergodic Markov chain [32]. Now, using the stationary distribution of the ergodic Markov chain $\{\mathbf{Q}_{\mathcal{S}}^{n+1}(1),C_{\mathcal{S}}^{n}\}_{n>\tau}^{\infty}$ at cycle $n=\tau+1$, we have a stationary cycle time sequence which has a finite and unique expectation.

For any q_k in any cycle $n>\tau$, consider Eq. (3.2) and rewrite it into the following:

$$Q_k^{n+1}(1)+A_k^{1,n+1}-X_k^{n+1}=[Q_k^n(1)+A_k^{1,n}-X_k^n]^{+}+(A_k^{2,n}+A_k^{1,n+1}-X_k^{n+1}). \tag{3.8}$$

Let $W_k^n=[Q_k^n(1)+A_k^{1,n}-X_k^n]^{+}$ and $H_k^n=(A_k^{2,n}+A_k^{1,n+1}-X_k^{n+1})$, Eq. (3.8) becomes

$$W_k^{n+1}=[W_k^n+H_k^n]^{+}. \tag{3.9}$$

Clearly, the function $[x]^{+}$ is monotonically increasing respect to x. As the cycle time is stationary and ergodic at each cycle $n>\tau$, and the arrival processes are Poisson, these imply that the total number of customers arrive at q_k, i.e., $(A_k^{2,n}+A_k^{1,n+1})$, will also be stationary and ergodic. Furthermore, if $q_k \in \mathcal{U}$, then $X_k^{n>\tau} = M_k$; if $q_k \in \mathcal{S}$, from Lemma 3.2.3 we know that $X_k^{n>\tau}$ also be stationary and ergodic. Consequently, the variable $H_k^n = (A_k^{2,n} + A_k^{1,n+1} - X_k^{n+1})$ is stationary and ergodic when $n > \tau$. Then, for the random sequence W_k^n, according to Loynes' lemma [45], the limit $\lim_{n\to\infty} W_k^n$ (and equivalently $\lim_{n\to\infty} Q_k^n(1)$) exists, though it may either tend to a constant or infinity. Now for any individual queue we are ready to use Loynes' theorem to examine its stability. Because $P(\tau < \infty) = 1$ and therefore the cycles up to τ are negligible when considering the long term average, for $q_i \in \mathcal{U}$, the service rate μ_i is equal to

$$\mu_i = \frac{M_i}{\mathbf{E}C_{\mathcal{S}}},$$

where $\mathbf{E}C_{\mathcal{S}}$ can be computed from Eq. (3.5). Applying Loynes' theorem to the unstable q_i we have

$$\lambda_i > \frac{M_i}{\mathbf{E}C_{\mathcal{S}}} \Rightarrow \frac{\lambda_i}{M_i} > \frac{1}{\mathbf{E}C_{\mathcal{S}}}.$$

According to Lemma 3.2.3, for $q_j \in \mathcal{S}$, we have

$$M_j \geq \mathbf{E}X_j = \lambda_j \mathbf{E}C_{\mathcal{S}} \Rightarrow \frac{1}{\mathbf{E}C_{\mathcal{S}}} \geq \frac{\lambda_j}{M_j}.$$

Then, we have $\frac{\lambda_i}{M_i} > \frac{\lambda_j}{M_j}$, which contradicts with the assumption that $\frac{\lambda_i}{M_i} \leq \frac{\lambda_j}{M_j}$. This implies that if q_j is assumed stable, q_i cannot be unstable, i.e., q_j's stability implies q_i's stability.

Now for the second part of the theorem, assume q_i is unstable but q_j is stable. We then have $q_i \in \mathcal{U}$ and $q_j \in \mathcal{S}$. After applying Loynes' theorem to q_i, we have

$$\lambda_i > \frac{M_i}{\mathbf{E}C_{\mathcal{S}}} \Rightarrow \frac{\lambda_i}{M_i} > \frac{1}{\mathbf{E}C_{\mathcal{S}}}.$$

On the other hand, for q_j, we have

$$M_j \geq \mathbf{E}X_j = \lambda_j \mathbf{E}C_{\mathcal{S}} \Rightarrow \frac{1}{\mathbf{E}C_{\mathcal{S}}} \geq \frac{\lambda_j}{M_j}.$$

Again, we have $\frac{\lambda_i}{M_i} > \frac{\lambda_j}{M_j}$, which contradicts with the assumption that $\frac{\lambda_i}{M_i} \leq \frac{\lambda_j}{M_j}$. Thus, q_i's instability implies q_j's instability. This finishes the proof. □

From the above theorem, we can see that a relative stability relation indeed exists between any two queues in the polling system in the sense that if $\frac{\lambda_i}{M_i} \leq \frac{\lambda_j}{M_j}$, q_j's stability implies q_i's stability (or q_i's instability implies q_j's instability). Equivalently, we can also say if $\frac{\lambda_i}{M_i} \leq \frac{\lambda_j}{M_j}$, q_i is more stable than q_j (in the sense that q_i's instability implies q_j's instability). Though we only consider a simple version of the polling systems in this section, the analysis can be easily applied to more general polling models such as the one (with limited service policy only) considered in [29]. In the next section, we show that the similar properties to the above can also be found in a version of the ALOHA network.

3.3 A Slotted Buffered ALOHA Network

The ALOHA network considered here consists of a single server (a broadcast channel) and a finite set of k distributed queues (stations), denoted as q_i, $i \in \{1, ..., k\}$ [47, 57]. Each queue has infinite buffers for storing incoming fixed-length packets and transmits packets through the broadcast channel. Transmissions over the channel are divided into intervals, called slots. A slot duration corresponds to the transmission time of a packet. Assume that the queues know exactly the boundaries of the slots, and transmissions can only be started at the beginning of slots. During each slot, q_i attempts to transmit a packet with transmission probability p_i, provided that it is not empty. A successful transmission occurs when only one non-empty queue tries to transmit a packet during a slot. Otherwise, a collision occurs when more than one queues try to transmit simultaneously. When a queue successfully transmits a packet, it removes the transmitted packet from its buffer; otherwise, the queue must try to retransmit the packet in the next slot. The arrival process of packets to each q_i is assumed Bernoulli with average rate λ_i. In addition we assume that the arrival processes to the queues are mutually independent, the operations of each queue are independent to the operations of the others as well as independent to all other random processes in the system.

Let A_i^n be the number of packets arrived at q_i during slot n, and Q_i^n be the queue length of q_i at the beginning of slot n. Define B_i^n as the random variable that indicates whether q_i successfully transmits a packet during slot n, i.e., $B_i^n = 1$ when a successful transmission occurs at q_i during slot n and $B_i^n = 0$ otherwise. Then Q_i^n satisfies the following

$$Q_i^{n+1} = (Q_i^n - B_i^n)^+ + A_i^n. \tag{3.10}$$

It can be proved that the k-dimensional queue process $(\mathbf{Q}^n = (Q_1^n, Q_2^n, ..., Q_k^n))_{n=1}^{\infty}$ is an irreducible and aperiodic Markov chain [65, 72] and we state it as a lemma below.

Lemma 3.3.1 *The joint queue length process* $\mathbf{Q}^n$ *of the ALOHA network described above is an irreducible and aperiodic Markov chain at the beginning of each slot.*

According to the meaning of the random variable B_i^n, the expectation of B_i^n is the successful transmission probability at q_i, i.e., $\mathbf{E}B_i = P(B_i = 1)$. Let $\mathbf{z}^n$ be a k-dimensional random binary vector represents the queue length status at the beginning of the nth slot, i.e., $z_i^n = 1$ implies that q_i is not empty at the beginning of the nth slot while $z_i^n = 0$ otherwise. Let $\mathbf{\Theta_z}$ be the sample space of $\mathbf{z}^n$. Then we have

$$P(B_i^n = 1) = p_i \sum_{\substack{\hat{\mathbf{z}} \in \mathbf{\Theta_z}, \\ z_i^n = 1}} [P(\mathbf{z}^n = \hat{\mathbf{z}}) \prod_{i \neq j} (1 - p_j)^{z_j^n}]. \tag{3.11}$$

In the next lemma, we show that for a given traffic pattern of the arrival rates of the queues, the random variable B_i^n of q_i has a stationary distribution in the steady state.

Lemma 3.3.2 *Assume that under a certain arrival traffic pattern* $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_k)$, *there is a partition of the queues in the ALOHA network, i.e.,* $\{q_1, q_2, ..., q_k\} = \mathcal{S} \cup \mathcal{U}$, *such that all the queues in* $\mathcal{S}$ *are stable while all the queues in* $\mathcal{U}$ *are unstable. Then for each* q_i, $P_i^s \triangleq \lim_{n\to\infty} P(B_i^n = 1)$ *exists and equals to*

$$P_i^s = p_i \sum_{\substack{\hat{\mathbf{z}} \in \mathbf{\Theta_z}, \\ z_i = 1}} [P(\mathbf{z} = \hat{\mathbf{z}}) \prod_{i \neq j} (1 - p_j)^{z_j}] \tag{3.12}$$

Proof: In the first case, if $\mathcal{U}=\emptyset$, then the ALOHA network is stable for the given arrival traffic pattern Λ. This implies the k-dimensional Markov chain of the queue length process $\mathbf{Q}^n = (Q_1^n, Q_2^n, ..., Q_k^n)$ is ergodic and there exists a stationary distribution π of $\mathbf{Q}^n$. Now if we let the process $\mathbf{Q}^n$ starts with the initial distribution π, i.e., let $\mathbf{Q}^1$ distributes as π, the resulting process $\mathbf{Q}^n$ is stationary and ergodic. Consequently, the random binary vector $\mathbf{z}^n$ will also be stationary and ergodic because it represents the status of whether the queues are empty or not at the beginning of the nth slot. Let $\mathbf{z}=\lim_{n\to\infty}\mathbf{z}^n$, then $\mathbf{z}$ has a stationary distribution. Take the limit of Eq. (3.11) with respect to the slot number n, we have the conclusion that P_i^s exists and Eq. (3.12) holds.

Now assume $\mathcal{U} \neq \emptyset$ and there is at least one $q_j \in \mathcal{U}$. Define random variable $\tau_j = \sup\{n : Q_j^n = 0\}$, the last slot that the queue length at q_j equals to 0. Similar to the discussion in the last session, we have $P(\tau_j < \infty) = 1$, otherwise, it would contradict with the assumption that q_j is unstable. For any other q_i in $\mathcal{U}$ we can define τ_i and can have similar conclusion that $P(\tau_i < \infty) = 1$. Let $\tau = \sup(\tau_j, \forall q_j \in \mathcal{U})$. Then at the beginning of each slot $n > \tau$, at any $q_j \in \mathcal{U}$, we have $P(Q_j^n \geq 1) = 1$. Now consider the queues in $\mathcal{S}$ and let $\mathbf{Q}_{\mathcal{S}}^n$ be the queue length process vector for the stable queues. Define random variable $K^0 = \inf\{n > \tau : \mathbf{Q}_{\mathcal{S}}^n = \mathbf{0}\}$, the first slot after τ such that all the queues in $\mathcal{S}$ are empty. Further define $K^{n+1} = \min\{n > K^n : \mathbf{Q}_{\mathcal{S}}^n(1) = \mathbf{0}\}$, $R^0 = K^0$, and $R^{n+1} = K^{n+1} - K^n$. Because we assume that all the queues in $\mathcal{S}$ are stable, the $|\mathcal{S}|$ dimensional Markov chain $\mathbf{Q}_{\mathcal{S}}^n$ is ergodic and therefore we have $P(K^0 < \infty) = 1$ and consequently $P(R^0 < \infty) = 1$. Furthermore, $\mathbf{Q}_{\mathcal{S}}^n$ is regenerative with respect to R^n, and R^n defines a delayed renewal process with R^0 as delay and $\mathbf{E}(R) < \infty$. Consequently, if we let $\mathbf{Q}_{\mathcal{S}}^n$, $n > \tau$, starts with the stationary distribution of $\mathbf{Q}_{\mathcal{S}}^n$, then $\mathbf{z} = \lim_{n\to\infty}\mathbf{z}^n$ also exists and the jth component always be 1, where $q_j \in \mathcal{U}$. When taking the limit of Eq. (3.11) with respect with n, we have Eq. (3.12). Finally, the same argument applies to the case that $\mathcal{S} = \emptyset$ and this finishes the proof.

□

Next theorem shows the relative stability relation in the ALOHA network.

Theorem 3.3.1 *For any two queues and a given arrival rate vector of the queues* $\Lambda=(\lambda_1,\lambda_2,...,\lambda_k)$ *in the ALOHA network, assume* $\frac{\lambda_i(1-p_i)}{p_i}\leq\frac{\lambda_j(1-p_j)}{p_j}$*, then*

(a) q_j*'s stability implies* q_i*'s stability;*

(b) q_i*'s instability implies* q_j*'s instability.*

Proof: For the first part of the theorem, assume that q_j is stable but q_i is unstable. For the arrival traffic pattern Λ, all the queues can be partitioned into $\mathcal{S}$ and $\mathcal{U}$, and we have at least $q_j\in\mathcal{S}$ and $q_i\in\mathcal{U}$. According to Lemma 3.3.2, for both q_i and q_j, the random variables B_i^n and B_j^n have stationary distribution P_i^s and P_j^s in the steady state, respectively. Recall that P_i^s and P_j^s is the successful transmission probabilities of q_i and q_j in the steady state. Furthermore, P_i^s and P_j^s are also the service rates of q_i and q_j in the steady state.

Now for each queue q_k in the system at slot $n>\tau$, consider Eq. (3.10) and rewrite it into the following:

$$Q_k^{n+1}-B_k^{n+1}=[Q_k^n-B_k^n]^+ + (A_k^n-B_k^{n+1}). \tag{3.13}$$

Let $W_k^n=[Q_k^n-B_k^n]^+$ and $H_k^n=(A_k^n-B_k^{n+1})$, Eq. (3.13) becomes

$$W_k^{n+1}=[W_k^n+H_k^n]^+. \tag{3.14}$$

Because the arrivals during a slot, i.e., A_k^n, and the variable B_k^{n+1} are stationary and ergodic, the random sequence H_k^n is also stationary and ergodic. According to Loynes' lemma [45], the sequence W_j^n is monotonically increasing with respect to n and the limit $\lim_{n\to\infty}W_j^n$ ($\lim_{n\to\infty}Q_k^n$) exists, though the limit may be infinite. For a stable q_j, according to Loynes' theorem, we have

$$\lambda_j<P_j^s=p_j\sum_{\substack{\hat{\mathbf{z}}\in\mathbf{\Theta_z},\\ z_j=1}}[P(\mathbf{z}=\hat{\mathbf{z}})\prod_{j\neq k}(1-p_k)^{z_k}]. \tag{3.15}$$

Multiply $\frac{(1-p_j)}{p_j}$ to both sides of the above inequality and note that the z_k components of $q_k\in\mathcal{U}$ will always be 1, Eq. (3.15) can be rewritten into

$$\frac{\lambda_j(1-p_j)}{p_j}<\prod_{q_k\in\mathcal{U}}(1-p_k)\sum_{\substack{\hat{\mathbf{z}}\in\mathbf{\Theta_z},\\ z_j=1}}[P(\mathbf{z}=\hat{\mathbf{z}})\prod_{q_k\in\mathcal{S}}(1-p_k)^{z_k}]. \tag{3.16}$$

For the unstable q_i, according to Loynes' theorem, we have

$$\lambda_i > P_i^s = p_i \sum_{\hat{\mathbf{z}} \in \mathbf{\Theta_z}} [P(\mathbf{z} = \hat{\mathbf{z}}) \prod_{i \neq k} (1 - p_k)^{z_k}]. \tag{3.17}$$

After rewritten and rearranged we have

$$\frac{\lambda_i (1 - p_i)}{p_i} > \prod_{q_k \in \mathcal{U}} (1 - p_k) \sum_{\hat{\mathbf{z}} \in \mathbf{\Theta_z}} [P(\mathbf{z} = \hat{\mathbf{z}}) \prod_{q_k \in \mathcal{S}} (1 - p_k)^{z_k}]. \tag{3.18}$$

Clearly, the following inequality holds:

$$\sum_{\hat{\mathbf{z}} \in \mathbf{\Theta_z}} [P(\mathbf{z} = \hat{\mathbf{z}}) \prod_{q_k \in \mathcal{S}} (1 - p_k)^{z_k}] \geq \sum_{\substack{\hat{\mathbf{z}} \in \mathbf{\Theta_z}, \\ z_j = 1}} [P(\mathbf{z} = \hat{\mathbf{z}}) \prod_{q_k \in \mathcal{S}} (1 - p_k)^{z_k}].$$

Hence from Eq. (3.16) and (3.18) we have $\frac{\lambda_i(1-p_i)}{p_i} > \frac{\lambda_j(1-p_j)}{p_j}$, which contradicts with the assumption that $\frac{\lambda_i(1-p_i)}{p_i} \leq \frac{\lambda_j(1-p_j)}{p_j}$. This implies q_i must be stable when q_j is stable, i.e., q_j's stability implies q_i's stability.

Now consider q_i is unstable but q_j is stable, based on similar discussion to the above, we can arrive at the same contradiction that $\frac{\lambda_i(1-p_i)}{p_i} > \frac{\lambda_j(1-p_j)}{p_j}$. Thus, if q_i is unstable, q_j is also unstable, i.e., q_i's instability implies q_j's instability. This finishes the proof. □

From the above theorem, we can see that the relative stability relations also exist among the queues in the ALOHA network. Specifically, if $\frac{\lambda_i(1-p_i)}{p_i} \leq \frac{\lambda_j(1-p_j)}{p_j}$, then q_j's stability implies q_i's stability, and q_i's instability implies q_j's instability. In the next section, we show the relative stability relations also exist in a processor sharing system.

3.4 A Processor Sharing System

In this section we study the relative stability relations of a processor sharing system. There are one server and k queues in the system, denoted as q_i, $i \in \{1, ..., k\}$. The arrival process of the customers to the system is assumed as a marked point process $\mathbf{\Psi} = \{[T_l, (S_l, I_l)]\}_{l=-\infty}^{\infty}$ on the real line with mark space $\mathbb{K} = \mathbb{R}^+ \times \{1, \cdots, k\}$, where T_l is the arrival epoch of the lth customer, S_l is the service time requirement of the lth customer, and I_l is the queue that the lth customer joins. We assume $\mathbf{\Psi}$ is stationary, ergodic, and simple, i.e., $\cdots < T_0 \leq 0 < T_1 < \cdots$. The arrivals to q_i is given by the marked point process $\mathbf{\Psi_i} = \{[T_{i,l}, S_{i,l}]\}_{l=-\infty}^{\infty}$, which is also stationary and ergodic. Let $\lambda_i = \mathbf{E}\mathbf{\Psi_i}((0,1]\times\mathbb{R}_+)$ be the intensity of the points occurred during time interval $(0,1)$, i.e., the average arrival rate at q_i, and $b_i = \mathbf{E}S_i^0$ be the expectation of the service time S_i^0 of a typical q_i customer according to the event stationary distribution of the marked point process. The customers are served by the single server according to the processor sharing policy: the first customers of the non-empty queues are served by the server simultaneously. More precisely, if $Q_i(t)$ is the number of customers in q_i at time t, then totally $g(t) = \sum_{i=1}^{k} \mathbb{I}\{Q_i(t) > 0\}$ customers will be served by the server at time t, and each customer receives $\frac{1}{g(t)}$ of the capacity of the server, provided that $g(t)$ is positive.

At time t, let $R_{i,l}(t) \in \mathbb{R}_+$ be the residual service time of the lth customer who arrived at q_i before t, $l = 1, 2, \cdots$, ordered reversely of the arrival instants. That is, $R_{i,l}(t)$ is the residual service time of the lth last customer who arrived before t when all the arrivals before t are considered. Further let $R_i(t) = (R_{i,1}(t), R_{i,2}(t), \cdots)$ be the infinite vector of the residual service time in q_i, and $R(t) = (R_i(t), \cdots, R_k(t))$ be the residual service time vector of the system at time t, respectively. In the following, we construct a stationary and ergodic version of the $R(t)$ based on the sample paths of $\mathbf{\Psi}$. Let

$$M_K = \{\psi = \{[t_l, (s_l, i_l)]\}_{l=-\infty}^{\infty} : ... < t_0 \leq 0 < t_1 < ..., \lim_{l\to\pm\infty} t_l = \pm\infty, s_l \in \mathbb{R}_+, i_l \in \{1, ..., k\}\}$$

be the set of all possible sample paths of Ψ, i.e., $P(M_K) = 1$. For any sample path $\psi \in M_K$ and $\tau > 0$, define the system state as following

$$r_i^{(\tau)}(t,\psi) = (r_{(i,1)}^{(\tau)}(t,\psi), r_{(i,2)}^{(\tau)}(t,\psi), ...), \qquad i = 1, ..., k, \tag{3.19}$$

$$r^{(\tau)}(t,\psi) = (r_i^{(\tau)}(t,\psi), ..., r_k^{(\tau)}(t,\psi)), \tag{3.20}$$

where $r^{(\tau)}(t,\psi)$ is the residual service time for the customers at time t as if the system was started empty at $t-\tau$ with input ψ. Note that $r_{(i,l)}^{(\tau)}(t,\psi) = 0$ if $t_l < t - \tau$

by definition, i.e., customers arrived before $t-\tau$. The workload and queue length in q_i at time t can be given by the follows respectively,

$$v_i^{(\tau)}(t,\psi) = \sum_{l=1}^{\infty} r_{i,l}^{(\tau)}(t,\psi), \qquad i = 1, ..., k, \tag{3.21}$$

and

$$Q_i^{(\tau)}(t,\psi) = \sum_{l=1}^{\infty} \mathbb{I}\{r_{i,l}^{(\tau)}(t,\psi) > 0\}, \qquad i = 1, ..., k. \tag{3.22}$$

For any $\tau_2 > \tau_1 > 0$, and any lth customer in q_i, we have $r_{i,l}^{(\tau_2)}(t,\psi) \geq r_{i,l}^{(\tau_1)}(t,\psi)$. In fact, if the lth customer arrived at $t_l < t-\tau_2 < t-\tau_1$, then $r_{i,l}^{(\tau_2)}(t,\psi) = r_{i,l}^{(\tau_1)}(t,\psi) = 0$. If it arrived at $t-\tau_2 < t_l < t-\tau_1$, then $r_{i,l}^{(\tau_2)}(t,\psi) \geq r_{i,l}^{(\tau_1)}(t,\psi) = 0$. If it arrived at $t-\tau_2 < t-\tau_1 < t_l$, by noting that for any q_i and any $x \in [t_l, t]$, $Q_i^{(\tau_2)}(x,\psi) \geq Q_i^{(\tau_1)}(x,\psi)$, and each non-empty queue shares an equal portion of the server capacity, we have $1/g^{\tau_2}(x) \leq 1/g^{\tau_1}(x)$, for $x \in [t_l, t]$. Therefore, we also have $r_{i,l}^{(\tau_2)}(t,\psi) \geq r_{i,l}^{(\tau_1)}(t,\psi)$. In other words, $r_{i,l}^{(\tau)}(t,\psi)$ is non-decreasing with respect to τ. Because this monotonicity, the limits as $\tau \to \infty$ exist:

$$\lim_{\tau\to\infty} r_{i,l}^{(\tau)}(t,\psi) = r_{i,l}(t,\psi), \qquad i=1,...,k, \quad l=1,2,..., \tag{3.23}$$

and

$$\lim_{\tau\to\infty} r^{(\tau)}(t,\psi) = r(t,\psi), \tag{3.24}$$

where $r(t,\psi)$ is the system state at t as if it was started empty at time $-\infty$. When consider all $\psi \in M_K$, we have the stationary and ergodic process $r(t, M_K)$. Based on the backward construction we know that the process $r(t, M_K)$ is the minimal stationary and ergodic state process that satisfies the system dynamic [45]. From now on, let $R(t) \triangleq r(t, M_K)$ and $Q_i(t) \triangleq \sum_{l=1}^{\infty} \mathbb{I}\{r_{i,l}(t, M_K) > 0\}$, where $Q_i(t)$ is the stationary queue length at t, with possibility that the queue length for some q_j are infinite.

Based on the above discussion, we have the following lemma regarding to the service received by any served customer and by each queue.

Lemma 3.4.1 *In the processor sharing system, the expectations of the service share received by any served customer and by any q_i exist and are unique.*

Proof: Let $C^*(t)$ be the service received by any served customer at t, we have

$$C^*(t) = \frac{1}{g(t)} = \frac{1}{\sum_{i=1}^{k} \mathbb{I}\{Q_i(t) > 0\}}.$$

Assume $C^*(t) = 1$ if the system is empty. For any q_i, the service share it receives at t is then

$$C_i(t) = \min(Q_i(t), 1)C^*(t).$$

Because $Q_i(t)$ is stationary and ergodic, the following two limits exist:

$$\mathbf{E}C^*(0) = \lim_{t\to\infty} \frac{1}{t} \int_0^t C^*(t)dt, \qquad \mathbf{E}C_i(0) = \lim_{t\to\infty} \frac{1}{t} \int_0^t C_i(t)dt,$$

and satisfy

$$\mathbf{E}C_i(0) \leq \mathbf{E}C^*(0). \tag{3.25}$$

This finishes the proof. □

The next theorem shows the relative stability relation in the process sharing system.

Theorem 3.4.1 *For any two queues and a given arrival rate vector of the queues* $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_k)$ *in the processor sharing system, assume* $\lambda_i b_i \leq \lambda_j b_j$*, then*

(a) q_j*'s stability implies* q_i*'s stability;*

(b) q_i*'s instability implies* q_j*'s instability.*

Proof: First assume q_j is stable but q_i is unstable. Because the stationary and ergodicity of the system state process $R(t)$, we can apply Loynes' theorem to the queues. For the stable q_j, based on Lemma 3.4.1, the average service share that it receives is $\mathbf{E}C_j(0)$. Since the average service time for each customer at q_j is b_j, the average service rate at q_j is $\frac{\mathbf{E}C_j(0)}{b_j}$. According to Loynes' theorem, q_j is stable implies

$$\lambda_j < \frac{\mathbf{E}C_j(0)}{b_j} \Rightarrow \lambda_j b_j < \mathbf{E}C_j(0).$$

On the other hand, for the unstable q_i, it will never empty. Therefore, the average service share q_i receives equals to the average service share each served customer receives, i.e., $\mathbf{E}C_i(0)=\mathbf{E}C^*(0)$. Thus the service rate of q_i is $\frac{\mathbf{E}C^*(0)}{b_i}$. Apply Loynes' theorem to q_i we have

$$\lambda_i > \frac{\mathbf{E}C^*(0)}{b_i} \Rightarrow \lambda_i b_i > \mathbf{E}C^*(0).$$

Because $\mathbf{E}C_j(0) \leq \mathbf{E}C^*(0)$, we have $\lambda_i b_i > \lambda_j b_j$, contradicts with our assumption. Therefore, q_j's stability implies q_i's stability. Similar contradiction also raises when assume q_i is unstable but q_j is stable, and consequently, q_i's instability implies q_j's instability. This completes the proof. □

The above theorem indicates that relative stability relations also exist among the queues in the processor sharing system with a more general arrival process assumption. In this case, if $\lambda_i b_i \leq \lambda_j b_j$, q_j's stability implies q_i's stability. In the next section, we summarize the findings in this chapter.

3.5 Summary

In this chapter we studied the relative stability relations for three SSMQSs. From the results we observe the follows:

1. the relative stability relations exist in all the three systems and the relations have connections to the queues' arrival patterns;
2. to obtain the relative stability relations does not require the explicit system or queue stability conditions;
3. in all the three systems studied, there are stationary regimes of some system state processes even when the system is unstable.

The first item suggests that in the SSMQSs each single queue's stability status correlates with other queues' stability status. Generally, this correlation is caused by the interaction (through competing the single server's capacity) among the queues. Nevertheless, from the results we can see that this correlation can be reflected by the relation of the arrival patterns of the queues. Therefore, by observing the relations of the arrival patterns of the queues, it is possible to tell this correlation (relative stability relation) among the queues. This also confirms one of our claims about the relative stability relation, i.e., it is dynamic and will be affected by the systems' traffic input. Moreover, the diversity of the three systems under consideration also suggests that the same relations may commonly exist in other SSMQSs. The second item implies that the relative stability relations among the queues are more accessible than the absolute stability conditions. For instance, even the exact stability conditions for the ALOHA network is unknown, we are still able to tell the relative stability relations between any two queues based only on the queues' arrival rates and the system settings, i.e., the transmission probabilities of the queues. On the other hand, as we will see in Chapter 5, the relative stability relations of the queues can be a useful tool when study the absolute stability conditions of the SSMQSs. The last item suggests that to use Loynes' theorem to study relative stability in general, we may require a SSMQS to preserve some stationary properties even when the system is unstable. This provide us a clue to identify the SSMQSs in which relative stability can be studied, and this will be done in the next chapter.

In the proofs in this chapter, a major tool we used is Loynes' theorem. Consequently, the stationary and ergodicity requirements of Loynes' theorem is crucial in our study. In order to show there is a stationary regime of the system state process, we used two approaches in this chapter: the Markov chain approach and Loynes' backward construction. Though the systems we selected are the basic ones, the analysis and arguments we used here are capable to show stationary and ergodicity for more complicated systems. For instance, in the polling system case, the server can visit the queues according to an ergodic Markovian routing process in which the transition probability $p^n_{i,j}$ is the probability that the server switch from q_i to q_j, and π_i is the stationary probability that the server will visit q_i next. Also, the number of customers can be served at q_i during a visit can be a (random) function of the queue length, i.e., $f_i(Q^n_i)$, as long as f_i is non-decreasing, satisfies $f_i(x) - f_i(y) \leq x - y$ if

$x > y$, and $\lim_{n\to\infty} f_i(Q_i^n) < \infty$. Service policies satisfy the first two properties are called monotonic and contractive policies [44]. Another possible generalization of the polling system is to allow each queue to have a reservation of services, i.e., gated at the server departure, or state dependent set-up time [15, 16]. For the processor sharing case, an variation can be allowing multiple servers in the system and having some permanent customers in the system [1, 2].

In the proofs we showed that if a queue is unstable, with probability 1, the queue length will be finite for only a finite amount of time. To simplify the analysis in the rest of the dissertation, from now on, we assume the following:

Assumption 3.5.1 *The queue length of an unstable queue is infinite as the time tends to infinity.*

This assumption implies that the queue length of an unstable queue will be finite for only a finite amount of time. We consider this assumption is reasonable since the heuristic meaning of stability is that the queue length remains finite if it is stable and otherwise if it is unstable.

4. RELATIVE STABILITY

4.1 Introduction

In this chapter we are going to present the main results of the dissertation: the properties with respect to relative stability for a class of SSMQSs. As we mentioned in the summary of Chapter 3, the stationary requirements of Loynes' theorem are crucial to the investigation of relative stability relations in SSMQSs since we use Loynes' theorem to examine the queues' stability. However, to check whether a stationary regime exists for any given system is rather troublesome and unnecessary, especially when considering different systems may use different scheduling algorithms and service policies. Therefore, we adopt another approach.

In the analysis of the three systems in the last chapter we learned that some system state processes are stationary and ergodic even when the system is unstable, e.g., the cycle time process in the polling system. This suggests us a criterion to classify the SSMQSs, that is, to classify SSMQSs based on whether a stationary regime of some system state processes exists when some of the queues are unstable. From the classification we identify two types of SSMQSs in this chapter: Type-1 systems and Type-2 systems, and our main focus will be the Type-1 systems. In Type-1 SSMQSs, stationary regimes of some system states exist even when the system is unstable, we can then apply Loynes' theorem to examine individual queues' stability. This allows us to define the degree of stability of a queue as well as the relative stability relations among the queues. In the definitions we also need to consider the dynamic of the traffic inputs because the degree of stability and relative stability relations are dynamic, i.e., affected by the queues' traffic inputs. This can be done by restricting the system traffic on certain paths in the traffic space.

The Type-1 SSMQSs have some useful properties through which we can solve the relative stability problems mentioned in Chapter 1. The most important property is probably the sufficient and necessary condition of two queues to be as stable as each other when the system traffic varies on some linear increasing paths. Intuitively, two queues are as stable as on a path if they become unstable simultaneously on the path. Once the paths on which two queues have as stable as relation are found, the rests of the paths will then be either the ones on which one queue is more stable than another or vice versa. After the relative stability relations and conditions of any two queues can be found on any path, by comparing any pair of queues in turn, we can further derive a stability ordering of the queues for a given path of the system traffic. This ordering specifies the ordinal of the queues of becoming unstable when the system traffic increases along the path. Moreover, the as stable as relation among the queues also suggests an simple approach to find the maximum stable throughput of a system. Specifically, for a given increasing path of system traffic, we can show that the maximum stable throughput can be achieved at the system stability boundary for a configuration of the system parameters such that all the queues are as stable as one another. Then, another two problems, namely, the characterization of the overall stability region and the stabilization of the Type-1 systems can be equivalently interpreted and formulated. Worth to mention that these set of properties are interesting and has not been reported in the literature from our knowledge.

For Type-2 systems, because of their complexity, we only discuss their properties through some examples.

4.2 Two Types of Single-Server-Multiple-Queue Systems

Recall that one of the common items we observed in the studies of the three systems in Chapter 3 is that all the three systems can have stationary regimes of some system states even when the systems are unstable. This observation inspires us to define a general system model accordingly. Based on the existence of stationary regimes of some system state processes, we are able to define two different types of SSMQSs. Follows are the definitions.

Definition 4.2.1 ***(Type-1 Systems)***

In a SSMQS with k queues, let the system state process be $\{\mathbf{W}^n = (W_1^n, W_2^n, ..., W_k^n)\}_{n=1}^{\infty}$. If for any q_i, $i \in \{1, ..., k\}$, the queue state process $\{W_i^n\}_{n=1}^{\infty}$ can be represented recursively by the following transformation

$$W_i^{n+1} = f(W_i^n, H^n(\mathbf{W}^n)), \tag{4.1}$$

where $H^n(\mathbf{W}^n)$ is a function of the k-dimensional process $\mathbf{W}^n$, and $f(x, y)$ is non-negative, monotonic increasing and continuous from the left in x. Suppose in addition that for any given arrival traffic pattern $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_k)$, all the queues can be partitioned into subsets $\mathcal{S}$ and $\mathcal{U}$ such that at Λ all the queues in $\mathcal{S}$ are stable and all the queues in $\mathcal{U}$ are unstable in the steady state. Then the SSMQS is called a Type-1 system if the sequence $H^n(\mathbf{W}^n) = H^n((\mathbf{W}_{\mathcal{S}}^n, \mathbf{W}_{\mathcal{U}}^n))$ is stationary and ergodic under any partition of $(\mathcal{S}, \mathcal{U})$ (even one of them is empty).

In the above definitions, the state process W_i^n of q_i can be many kinds. For example, in the polling system and in the ALOHA network studied in the last chapter, the state processes of q_i are the remaining service at q_i after the server's nth visit, i.e., $W_i^n = [Q_i^n(1) + A_i^{1,n} - X_i^n]^+$ and $W_i^n = [Q_i^n - B_i^n]^+$, respectively. While in the processor sharing system, the state process of q_i is the remaining service time of all its customers at time t, i.e., $r_i(t, \psi)$. The function f represents the service policies employed by the server while the sequence of H^n can be considered as the overall effects to q_i from the other queues, the scheduling algorithms employed by the server, and other server related processes such as the service time processes, the switch-over time processes, and the set-up time processes. In the polling system, the sequence H^n of q_i is $H_i^n = (A_i^{2,n} + A_i^{1,n+1} - X_i^{n+1})$, the function f is $[x]^+$, and the transformation is $W_i^{n+1} = [W_i^n + H_i^n]^+$. Similarly, in the ALOHA network, the sequence H^n of q_i is $H_i^n =$

$(A_i^n - B_i^{n+1})$, the function f is $[x]^+$, and the transformation is $W_i^{n+1} = [W_i^n + H_i^n]^+$. In the process sharing system, as time is continuous, for q_i the equivalence to H^n is the number of queues with positive queue length at time t, i.e., $g(t) = \sum_{i=1}^{k} \mathbb{I}\{Q_i(t) > 0\}$, and the function f and the transformation is represented through the service share q_i received at time t, i.e., $C_i(t) = \min(Q_i(t), 1)C^*(t)$, because the remaining service time of a customer at time t at q_i will be directly affected by how much share q_i received at time t, i.e., $C_i(t)$.

From Lemmas 3.2.3, 3.3.2, and 3.4.1, for the three systems studied in the last chapter, we know that some system state processes are stationary and ergodic when the system is unstable, even with different partition of $(\mathcal{S},\mathcal{U})$, i.e., the H^ns. The definition of the Type-1 SSMQSs is actually based on this observation. Similar definition to the above can be given to continuous time state processes because we have analyzed the processor sharing system accordingly. However, if not specifically mentioned, we assume all the state processes in this study are in discrete time. The following is the definition of the Type-2 systems. The major difference between the two types of SSMQSs is that in Type-2 systems the sequence of $H^n(\mathbf{W}^n)$ may not be stationary and ergodic for any partition of $(\mathcal{S},\mathcal{U})$.

Definition 4.2.2 ***(Type-2 Systems)***

Same assumptions as in Definition 4.2.1, a SSMQS is a Type-2 System if the sequence $H^n(\mathbf{W}^n) = H^n(\mathbf{W}_{\mathcal{S}}^n, \mathbf{W}_{\mathcal{U}}^n)$ is stationary and ergodic only when any member of a certain subset of queues $\mathcal{E}$ does not belong to $\mathcal{U}$ for any partition of $(\mathcal{S},\mathcal{U})$, i.e., $\mathcal{E} \subseteq \mathcal{S}$.

In the above definition, some system state processes can have stationary regimes when all the queues belong to subset $\mathcal{E}$ are stable. Otherwise, because the $H^n(\mathbf{W}^n)$ is no longer stationary and ergodic, we may not able to use Loynes' lemma or other approach to conclude whether the system has stationary system state processes or not. In this sense, the Type-2 systems is more complex than Type-1 systems. Note that the Type-1 and Type-2 systems are not necessarily equivalent to a partition of the SSMQSs, i.e., it is possible to have other type of SSMQSs. Nevertheless, in the rest of this dissertation, our main focus will be the Type-1 systems only.

From the discussion in the Chapter 3, especially from the Eqs. (3.9), (3.14), and (3.24), we know that all the three systems studied there belong to Type-1 systems. In general, however, it is hard to enumerate all the members for the two types of systems. In the following we provide a sufficient condition to identify a Type-1 system. The model in the theorem is similar to the one discussed in [64]. In condition 2 of the theorem, the effective visit means that during such visits at least one of the customers from the queue will be served. This setting allows the theorem to cover systems such as the ALOHA network. The stationary of the service policy means that the service policy either will not change over time, or change according to a stationary distribution. More details about the service policies that satisfy condition 2 can be found in [44, 64].

Theorem 4.2.1 *A SSMQS is Type-1 if the follows hold:*

(a) The multiple dimensional queue length process $\{\mathbf{Q}^n\}_{n=1}^{\infty}$ is a homogeneous, aperiodic, and irreducible Markov chain.

(b) The service policy in each effective visit at each queue is stationary, limited, monotonic, and contractive.

(c) All the input processes such as arrival processes, service time processes, switch time processes, routing processes, set-up time processes, etc. are mutually independent, stationary, and ergodic.

Proof: As the multiple dimensional queue length process $\{\mathbf{Q}^n\}_{n=1}^{\infty}$ is a Markov chain, in general we are able to represent the queue length process of any queue as the following

$$Q^{n+1} = Q^n - X^n + A^n,$$

where X^n is the number of customers served during the server's nth effective visit to the queue and A^n is the total arrival between the server's nth effective visit and its $(n+1)$th effective visit to the queue. Rewrite the above into

$$Q^{n+1} - X^{n+1} = Q^n - X^n + (A^n - X^{n+1}),$$

let $W^n = (Q^n - X^n)$ and $H^n = (A^n - X^{n+1})$, we have a form of the state process as in Definition 4.2.1: $W^{n+1} = f(W^n, H^n)$. The monotonicity of the function f can be checked from the condition that the service policy at each queue is monotonic and contractive. The stationary of H^n is based on the following two reasons. First, because the service policy is stationary, limited, contractive, and monotonic, for any q_i, no matter it is stable or not, $\lim_{n\to\infty} X_i^n$ exists and is finite. In fact, if the queue is stable, then the queue has a stationary and ergodic regime. On the other hand, if the queue is unstable, because the assumption we made in the last chapter that a unstable queue will have an infinite queue length, and because the service policy is limited, the server will always serve the allowed maximum number of customers from the queue. Therefore, in both cases, $\lim_{n\to\infty} X_i^n$ exists and is finite. With the condition that the service time process at any queue is stationary and ergodic, it implies that the period of time that the server spends in any queue is also stationary and ergodic. Consequently, the arrivals to any queue during a period is also stationary and ergodic. The second reason is the condition that all the involved input processes are mutually independent, stationary, and ergodic. The two reasons imply that the function of these involved stationary and ergodic processes, H^n, is also stationary and ergodic. This finishes the proof. □

4.3 Degree of Stability

Before we discuss the degree of stability in a SSMQS, let us first take a closer look of a single-server-queue system. Given that the stationary requirements are satisfied, Loynes' theorem states that a single $G/G/1$ queue is stable if and only if the average arrival rate is less than the average service rate, otherwise the queue is unstable. Care is needed here because in the theorem by "average service rate", it actually means *the maximum rate at which the server can complete service times.* This rate is equivalent to the long term service rate that the server can serve the customers when considering all the customers are already in the queue at time 0 [60]. It is important to distinguish this maximum rate with the actual average service rate, which is equivalent to the long run service rate that the server can serve the customers for a given arrival process, or alternatively, the average departure rate of the customers when the queue operates in normal.

A well known balancing argument about a stable queue says that in average the number of arrivals to the queue should be equal to the average departures from the queue, i.e., the average arrival rate is equal to the average departure rate for a stable queue. As we can see, the average departure rate in the above argument has the meaning of the average service rate, and this average service rate should be less than or equal to the maximum service rate the server can provide to the queue, and the equality may be achieved at or beyond the queue's stability boundary. This is not surprising because for a given arrival rate if the queue is stable means that the server can handle the available customers well and may have potential to handle more customers. When the arrival rate reaches or crosses the queue's stability boundary, the server does not have any potential to serve more customers. In summary, in a single-server-queue system, let λ be the average arrival rate, μ be the average service rate, and $\hat{\mu}$ be the maximum service rate, then $\lambda = \mu \leq \hat{\mu}$ when λ is within or at the queue's stability boundary, i.e., the queue is stable and the equality achieves at the queue's stability boundary; on the other hand, $\lambda > \hat{\mu} = \mu$ when λ goes beyond the queue's stability boundary, i.e., the queue is unstable. In addition, the value of $\hat{\mu}$ is a positive constant for the single-server-queue system.

Based on the above discussion, we consider that Loynes' theorem actually defines a quantity that can measure the level of stability of a queue. More precisely, the theorem says that if $\lambda < \hat{\mu}$ the queue is stable while $\lambda > \hat{\mu}$ the queue is unstable. It is then natural to use the quantity $1 - \frac{\lambda}{\hat{\mu}}$ to measure how stable a queue is, i.e., the higher the value, the more stable the queue in the sense that the server has more potential to serve customers from the queue. Another meaning of the quantity $1 - \frac{\lambda}{\hat{\mu}}$ is the following geometrical interpretation. In a single-server-queue system the stability region is bounded, and the arrival rate can only increase along the real line.

If for a given arrival rate the system is stable, the arrival rate must be within the stability region. If we increase the arrival rate, at some point, it must hit the stability boundary. Hence we can use the difference $1-\frac{\lambda}{\hat{\mu}}$ to represent the distance from the current level of stability to the stability boundary.

Now let us return back to SSMQSs. When there are multiple queues in the system, the above discussion may not directly applicable as the maximal service rate that a queue can have in general is not a constant but a function of the system traffic and service policies at all the queues as well as the server's scheduling algorithms. Because in such a situation, multiple queues will compete for the server's capacity. This competition also implies that, when become unstable, a queue's average service rate in general is not equal to its maximum service rate, i.e., $\hat{\mu} \geq \mu$. However, if we set a constrain to the arrival pattern of all the queues such that it can only increase monotonically along a path (curve), then for each queue, the stability boundary of the queue is still a single point on the path, and the queue still achieves its maximum service rate at that point. In this way, we can still use the quantity $1-\frac{\lambda}{\hat{\mu}}$ to measure the stability level of a queue on the given traffic increasing path. We name the quantity as *the degree of stability of a queue for a given system traffic pattern on a given increasing path*, or degree of stability for short, and give its formal definition in the following.

Definition 4.3.1 ***(Degree of stability)***

In a SSMQS, if the arrival rates of the queues increase monotonically along a curve L in the system traffic space, for a given traffic point $\Lambda \in L$ and for any queue, we call the following quantity, $D_L^{\lambda}(q)$, as the degree of stability of q at Λ on L:

$$D_L^{\lambda}(q) \triangleq 1 - \frac{\lambda}{\hat{\mu}_L}, \tag{4.2}$$

where λ is the arrival rate component of q and $\hat{\mu}_L$ is the maximum service rate q can achieve on L.

As we discussed in Chapter 1, in general, the degree of stability problem is difficult to tackle because the value of $\hat{\mu}_L$ of a queue in a SSMQS is hard to compute.

Nevertheless, the about definition is still useful for solving relative stability problems. Because for relative stability problems, we only need the comparison results of two queues' degree of stability, and in such case, the value of $\hat{\mu}_L$ is not necessarily needed. In the above definition we do not have further restrictions of the curve L as long as it is monotonically increasing. With the definition of degree of stability, we can restate the Loynes' theorem for a single queue in the two types of SSMQSs as the following proposition.

Theorem 4.3.1 ***(Loynes' Theorem through Degree of Stability)***

For a given SSMQS $\mathbf{Q}$ and let the system traffic vary on a given path L, for any $q \in \mathbf{Q}$:

(a) if $\mathbf{Q}$ is Type-1, q is stable on L if and only if $D_L^\lambda(q) > 0$;

(b) if $\mathbf{Q}$ is Type-2 and the queues in $\mathcal{E}$ are stable, further if $q \notin \mathcal{E}$, q is stable on L if and only if $D_L^\lambda(q) > 0$; if $q \in \mathcal{E}$, q is stable on L only if $D_L^\lambda(q) > 0$;

Proof: The theorem is a direct result of the Loynes' theorem when applying it to a queue in the Type-i systems. For the first assertion, recall the definition of a Type-1 system that for any given traffic point Λ of on L, the state process of any queue can be represented by

$$W^{n+1} = f(W^n, H^n(\mathbf{W}^n)),$$

and $H^n(\mathbf{W}^n)$ is stationary and ergodic no matter what stability status of all queues will be. Then based on the Loynes' lemma [45], the state process W^n is stationary and has a limit. We can then apply Loynes' theorem to q. Specifically, if $q \in \mathcal{S}$, i.e., q is stable, it implies $\lambda < \hat{\mu}_L$; if $q \in \mathcal{U}$, i.e., q is unstable, it implies $\lambda > \hat{\mu}_L$. It is easy to see that these conditions are equivalent to the first assertion.

For the second assertion, we first assume $q \notin \mathcal{E}$, then the state process is also stationary and has a limit given that all the queues in $\mathcal{E}$ also belong to the subset $\mathcal{S}$. Using the same argument as in the first assertion, we have the first part of the second assertion. Now if $q \in \mathcal{E}$, then according to the definition of Type-2 systems, q can only be in $\mathcal{S}$ to have a stationary system state process. Therefore, we can only apply the necessary part of the Loynes theorem to $q \in \mathcal{E}$, and this implies $\lambda < \hat{\mu}_L$, which is the second part of the second assertion. This finishes the proof. □

Note that in Theorem 4.3.1 we intentionally not to discuss the case of $D_L^{\lambda}(q) = 1$, because the stability status of a queue at the boundary is notoriously difficult and omitting such case will not affect the analysis [45, 47].

It is also worth to note that in the above proposition we can only have the necessary part for some of the queues in Type-2 systems, i.e., queues in the subset of $\mathcal{E}$ in Type-2 systems. The reason is because the maximum service rates for those queues may not be able to be defined whenever one of those queues is unstable. Consider a scenario, say, in a Type-2 system and assume all the queues employ an exhaustive service policy. Then for a traffic point on a given increasing path, if the system is stable, because the exhaustive policy, each queue should have the maximum service rate equals to the server's capacity. However, whenever one queue becomes unstable, based on our assumption that the queue length of an unstable queue will grow to infinity, the queue will occupy the server forever. This implies the rest of the queues will receive no service at all afterward. Consequently, these queues may never achieve their maximum service rates. Hence for this system, if it is stable, each queue's maximum service rate of course equals to the server's capacity, and this implies the necessary part. On the other hand, without knowing whether the system is stable or not, it is unjustifiable to apply the degree of stability to a queue because it may not able to achieve its maximum service rate. Another issue is that, in Type-1 systems each queue always has a constant maximum service rate on a given path, and this is in general not true for Type-2 systems, even when each queue can achieve its maximum service rate when all the queues become unstable. This phenomenon is caused by the non-ergodicity of the function $H^n(\mathbf{W}^n)$ in Type-2 systems. Later we will construct examples to show this claim. The following corollary is a direct result of Theorem 4.3.1.

Corollary 4.3.1 *In a Type-1 system, for a given monotonic increasing path of the system traffic, and a system traffic point of the queues on the path, i.e.,* $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_k)$*, assume* $\frac{\lambda_i}{\hat{\mu}_i} \leq \frac{\lambda_j}{\hat{\mu}_j}$*, then*

(a) q_j*'s stability implies* q_i*'s stability;*

(b) q_i*'s instability implies* q_j*'s instability.*

Proof: Assume q_j is stable but q_i is unstable. Because it is a Type-1 system, from Theorem 4.3.1 we have $D_L^{\lambda_j}(q_j) > 0 \Rightarrow \frac{\lambda_j}{\hat{\mu}_j} < 1$, and $D_L^{\lambda_i}(q_i) < 0 \Rightarrow \frac{\lambda_i}{\hat{\mu}_i} > 1$, thus contradicting with the condition that $\frac{\lambda_i}{\hat{\mu}_i} \leq \frac{\lambda_j}{\hat{\mu}_j}$. Similarly we can have the same contradiction when letting q_i be unstable and assume q_j be stable. This finishes the proof. □

From the above corollary we can see that the relative stability relations we discovered for the three SSMQSs in Chapter 3 indeed exist commonly in Type-1 SSMQSs.

4.4 Properties and Relative Stability in Type-1 SSMQSs

In this section we are going to derive some useful and interesting properties of the Type-1 systems. Through these properties, we can solve the relative stability problems in Type-1 systems.

In a SSMQS, if the server will spend infinite amount of time to serve a queue solely given that the queue has enough customers, we call the queue a *capture* type; and *non-capture* type otherwise. For example, a queue with the exhaustive service policy will be a capture type while a queue with limited service policy will be a non-capture type. On the other hand, in processor sharing systems, though the server will spend infinite amount of time to serve an unstable queue, it is still able to serve other queues. In this sense, queues in the process sharing system are of non-capture type. In the following theorem, we first find out the types of the queues in Type-1 systems.

Theorem 4.4.1 *In a Type-1 system all the queues are of non-capture type.*

Proof: This property is a consequence of the stationary and ergodicity of the function $H^n(\mathbf{W}^n)$. Suppose one of the queues in the Type-1 system is of capture type. Then once the queue becomes unstable, based on the assumption that its queue length will be infinite, during the server's visit, the queue will occupy the server forever. This will at least cause the arrivals to the other queues become infinite, which violates the stationary and ergodicity of the function $H^n(\mathbf{W}^n)$ in the definition of Type-1 systems. Thus all the queues in a Type-1 system must be of non-capture type. □

Later we will construct an example of Type-2 systems in which all the queues are of non-capture type. This suggests that the condition of non-capture type of queues is only necessary for a SSMQS to be Type-1. In some systems, the concept of non-capture type is equivalent to the concept of *limited* type, which has the meaning that the maximum number of customers can be served at a queue is finite. However, in systems such as the one with processor sharing policy, a server will always serve customers from an unstable queue, i.e., the maximum number of customers that can be served is infinite. For this reason, we consider the concepts of capture and non-capture types of queues are more general and are better for reflecting the properties of the Type-1 systems.

Theorem 4.4.2 *In a Type-1 system, all the queues have positive constant average service rates when all the queues are unstable.*

Proof: Based on Theorem 4.4.1, all the queues in the Type-1 system are of non-capture type implies the server will spend only a finite amount of time (or capacity) at any queue when all the queues are unstable. Moreover, the stationarity and ergodicity of the function $H^n(\mathbf{W}^n)$ further implies that this amount of time (or capacity) will be stationary and ergodic. Together with the assumption that all the other involved processes, e.g., the arrival processes, the service time processes, the switchover time processes, etc. are stationary and ergodic, we can conclude that when all the queues become unstable the average service rate for each queue exists and is unique. This finishes the proof. □

In the following, we use the concept *unique system instability state of average service rate* to reflect the fact proved in Theorem 4.4.2, and *unique instability state* for short. In Type-2 systems, however, some of the queues may not have this guaranteed service share. Before we investigate more properties of Type-1 SSMQSs, in the following, we first introduce the concepts of *relative stability relations* of any two queues in Type-1 systems. The idea is to use the degree of stability of the queues as metrics to compare the queues in terms of stability.

Definition 4.4.1 ***(Relative Stability Relations on a Point)***
In Type-1 SSMQSs, at a given traffic point $\Lambda=(\lambda_1, \lambda_2, ..., \lambda_k)$ on a given monotonic increasing path L of the system traffic in the traffic space, if both q_i and q_j are stable at point Λ, then at Λ on L

- *q_i is said less stable than q_j if $D_L^{\lambda_i}(q_i) < D_L^{\lambda_j}(q_j)$, denoted as $q_i \prec_{(L,\Lambda)} q_j$;*
- *q_j is said more stable than q_i if $D_L^{\lambda_i}(q_i) > D_L^{\lambda_j}(q_j)$, denoted as $q_i \succ_{(L,\Lambda)} q_j$;*
- *q_i is said as stable as q_j if $D_L^{\lambda_i}(q_i) = D_L^{\lambda_j}(q_j)$, denoted as $q_i \asymp_{(L,\Lambda)} q_j$.*

On a give path L, if $q_i \prec_{(L,\Lambda)} q_j$ on any Λ on which both queues are also stable, we can then say q_i is less stable than q_j on the path L. Accordingly, we have the following definition of relative stability relations of two queues on a path L.

Definition 4.4.2 ***(Relative Stability Relations on a Path)***
In Type-1 SSMQSs, on a given monotonic increasing path L of the system traffic in the traffic space and at any Λ on L on which both queues are also stable,

- *q_i is said less stable than q_j if $q_i \prec_{(L,\Lambda)} q_j$ at any such Λ, denoted as $q_i \prec q_j$;*
- *q_j is said more stable than q_i if $q_i \succ_{(L,\Lambda)} q_j$ at any such Λ, denoted as $q_i \succ q_j$;*
- *q_i is said as stable as q_j if $q_i \asymp_{(L,\Lambda)} q_j$ at any such Λ, denoted as $q_i \asymp q_j$;*

Because of this, we also use the notation $D_L(q_i) < D_L(q_j)$ to represent at every traffic point Λ on L on which both queues are stable, $D_L^{\lambda_i}(q_i) < D_L^{\lambda_j}(q_j)$. The other two relations will be notated similarly. For convenience, we also say that q_j is *at least as stable as* q_i on a path if either $q_i \asymp q_j$ or $q_i \prec q_j$, and we denote this relation by $q_i \preceq q_j$. Besides the relations of two queue's degrees of stability, another meaning of the relative stability relations of the two queues on a path is that, say, if $q_i \prec q_j$ and we keep increasing the system traffic along the path, the less stable queue (q_i) becomes unstable first, i.e., the two queues have a relative stability relation on the path such that q_i's stability implies q_j's stability, and q_j's instability implies q_i's instability. Furthermore, the as stable as relation implies both queues are either stable or unstable at the same time. Alternatively, we can also say, when the system traffic increases along the path, q_i will hit its stability boundary before q_j does if q_i is less stable, and they will hit their stability boundaries at the same time if they are as stable as each other.

Even though we do not have any restriction for the path L besides monotonically increasing in the above definitions, it is easier to analyze the relative stability related problems for Type-1 SSMQSs if the paths are simple. Therefore, in the follows, we only consider L as linear increasing paths. To this end, let $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_n)$ be a *traffic point*, where λ_i is q_i's arrival rate. The set of all traffic points forms an Euclidean space, referred to as the *traffic space* and denoted by $\mathbb{R}^n$. Let $\mathcal{O} = (0, 0, ..., 0)$ be the origin of $\mathbb{R}^n$. For any given traffic point $\Lambda \in \mathbb{R}^n$, consider a *linear increasing path* which starts from $\mathcal{O}$ and passes through Λ. We represent the increasing path in its parameterized form, i.e., each point on the path can be represented by $\Lambda = \lambda \cdot K$, where K is the *direction vector* of the line, i.e., $K = (k_1, k_2, ..., k_n)$, $k_i \in \mathbb{R}^+$ is the *direction component* of q_i, and λ is a free variable $\lambda_i = \lambda \cdot k_i, \forall i$ (when some of the k_i are 0, the corresponding queues can be excluded from the model). Hereafter, for a given K, we denote the corresponding linear increasing path as L_K and a traffic point on the path as Λ_K. In the next property, we show that on a given path, the stability boundary of any queue is a fixed point, or equivalently, the maximum service rate of any queue on a given path is a constant. For a given L_K, we denote q_i's maximum service rate as $\hat{\mu}_i^K$.

Theorem 4.4.3 *In Type-1 SSMQSs, on a given linear increasing path L_K, each queue's maximum service rate is a constant.*

Proof: If this is not true, assume that there is at least one q_i does not have constant maximum service rate, and assume the maximum service rate of q_i has a distribution of $G(x) = P(\hat{\mu}_i^K \leq x)$. Then when q_i just becomes unstable, it achieves its maximum service rate and the time that the server spends at q_i will be $b_i/\hat{\mu}_i^K$ with the same distribution as $G(x)$, where b_i is the average service time of a customer at q_i. Therefore, the visit time of the server at the queue becomes a mixture of some stationary processes. This implies the visit time process is not ergodic, which contradicts with the definition of Type-1 SSMQSs. Hence, each queue must have constant maximum service rate on a given path, or equivalently, each queue's stability boundary on a given path is a fixed point. □

The next property is a direct result of Theorem 4.4.3, which states the relative stability relations of any two queues on a given path L_K.

Theorem 4.4.4 *In Type-1 SSMQSs and for a given linear increasing path L_K of the system traffic, $q_i \preceq q_j \Leftrightarrow \frac{k_i}{\hat{\mu}_i^K} \geq \frac{k_j}{\hat{\mu}_j^K}$.*

Proof: Definition 4.4.2 states that if $q_i \preceq q_j$ on L_K, we have $D_L(q_i) \leq D_L(q_j)$ on L_K. Based on the meaning of $D_L(q)$ and Eq. (4.2), and note that both $\hat{\mu}_i^K$ and $\hat{\mu}_j^K$ are constants on L_K from Theorem 4.4.3, the condition $q_i \preceq q_j$ implies the following holds at every traffic point (on which both queues are stable) on L_K:

$$\frac{\lambda_i}{\hat{\mu}_i^K} \geq \frac{\lambda_j}{\hat{\mu}_j^K} \Leftrightarrow \frac{\lambda \cdot k_i}{\hat{\mu}_i^K} \geq \frac{\lambda \cdot k_j}{\hat{\mu}_i^K} \Leftrightarrow \frac{k_i}{\hat{\mu}_i^K} \geq \frac{k_j}{\hat{\mu}_i^K}.$$

As a result, the theorem holds. □

From Theorem 4.4.4 we can obtain the relative stability relations for any two queues on a given path once we know the two queues' maximum service rates on the path. Consequently, we can know the relative stability relations for all the queues on the path. To clearly address these relations, we define the *stability ordering* in Type-1 SSMQSs for a given linear increasing path as following.

Definition 4.4.3 ***(Stability Ordering)***
For a Type-1 SSMQS and a given linear increasing path L_K, the ordering under which the queues becoming unstable is called the stability ordering of the path. Specially, the first queue becoming unstable is called the least stable queue and the last queue becoming unstable is called the most stable queue.

From Theorem 4.4.4 it is easy to see that for any given L_K, the stability ordering is unique. However, for a given L_K, the jth queue becoming unstable may not be unique since two or more queues can be as stable as each other on the path. Another useful point is that if two queues are as stable as each other on a given path implies the two queues have the same stability boundary on the path. In other words, the stability boundaries of the two queues have an intersection on a particular point on that path. Based on this observation, we have the next property states that in Type-1 SSMQSs, for any two queues q_i and q_j, if the linear increasing paths can be selected arbitrary, then there are paths for each of the three relative stability relations to hold i.e., we can find paths for each of the relations $q_i \prec q_j$, $q_i \succ q_j$, and $q_i \asymp q_j$.

Theorem 4.4.5 *In Type-1 SSMQSs and for any given two queues q_i and q_j, there are linear increasing paths for each of the three relative stability relations to hold given that the paths can be selected arbitrarily.*

Proof: First we notice that for any q_i and q_j, there are paths for both $q_i \prec q_j$ and $q_i \succ q_j$. In fact, from Theorem 4.4.2, each queue can receive a guaranteed service from the server even when all the queues are unstable. We can select paths on which q_i's arrival rate increases sufficiently slow when compare to its guaranteed service rate and all other queues have arrival rates increase sufficiently fast when compare to theirs, and keep q_i's arrival rate less than its guaranteed service rate when all other queues exceed theirs. Then when q_i approaches its stability boundary on the paths all the other queues will be unstable already. On these paths, we have $q_i \succ q_j$ relation. Similar, we can have paths on which $q_i \prec q_j$ holds. For the paths on which $q_i \succ q_j$ holds, q_j's stability boundary is within q_i's, while for the paths on which $q_i \prec q_j$ holds, q_i's stability boundary is within q_j's. Next by noting that the stability regions of q_i and q_j are bounded, their stability boundaries must have at least an intersection. Otherwise, there will be a "hole" at the point at where the two queues' stability boundaries cross, and this is obviously not true. Therefore, there is at least one path on which the relation $q_i \asymp q_j$ holds. □

In the above theorem, if the paths cannot be selected arbitrarily, the conclusion may not be true. Consider an example of a symmetric polling system with gated limited service policy at each queue. In this system, this is only one possible linear increasing path of the queues, namely, $\lambda_1 = \lambda_2 = ... = \lambda_n$. On this path only one relative stability relation can exist for all the queues, that is, all the queues are as stable as one another. In the following, if not mentioned specifically, we assume all the linear increasing paths can be selected arbitrarily.

At this point, we have defined relative stability for Type-1 SSMQSs based on the concept of degree of stability. Also we have provided conditions under which the relative stability relations of queues can be determined on a given linear increasing path. Consequently, the stability ordering on the path can be known. However, to apply the above results in any practical systems, we must also provide ways to compute the degree of stability of the queues for any given linear increasing path, which in turn requires the explicit stability conditions of a queue because the maximum service rate of a queue is achieved at its stability boundary. This, however, is not an easy task in general. Nevertheless, as we have seen and discussed in Chapter 3, to determine the relative stability relations of the queues for the three systems on a given traffic point, the explicit stability condition is not necessarily required. In other words, we are able to have the comparison results of the degree of stability of the queues on a given linear increasing path without knowing the degree of stability of the queues at the points on the path. The reason, as we are going to state in Theorem 4.4.7, is that in Type-1 SSMQSs the relative stability relations of any two queues only depend on the relations of their arrival rates (direction components) and are independent to factors of other queues.

To prove the theorem, we need several steps. In Lemma 4.4.1, we first show the necessary condition for any two queues to be as stable as each other in the traffic space. Then we show the sufficient conditions for any one queue to be more stable than or less stable than any other queue in Lemma 4.4.2. Next we prove in Lemma 4.4.3 that for any two queues to have a relative stability relation on a given path, the ratio of their direction components is independent to any other queues' influence. Finally we have the sufficient conditions for any two queues to be as stable as each other, the necessary conditions for any one queue to be more stable than or less stable than any other queue in Lemma 4.4.4. The combination of Lemmas 4.4.1, 4.4.2, 4.4.3, and 4.4.4 leads to the Theorem 4.4.7.

We start the proof with Lemma 4.4.1, which states the necessary conditions for any two queues to be as stable as each other in the whole parameters.

Lemma 4.4.1 *In Type-1 SSMQSs, any two queues are as stable as each other on a given linear increasing path L_K only if the ratio of their direction components of the path is an independent constant, i.e., $q_i \asymp q_j \Longrightarrow k_i/k_j = C_{i,j}$, where $C_{i,j}$ is a universal constant independent to L_K.*

Proof: It is equivalent to prove that all the paths on which $q_i \asymp q_j$ can only be given by $k_i/k_j = C_{i,j}$. Note that in the n-dimensional traffic space $k_i/k_j = C_{i,j}$ is a $(n-1)$-dimensional hyperplane. To prove the proposition, we first show that the instability region of all queues of Type-1 SSMQSs is an open n-dimensional cuboid in the traffic space.

Consider all the linear increasing paths on which q_1 is the most stable queue. On any of these paths, once all the other queues become unstable, the stability boundaries of q_1 on all these paths should be the "same", i.e., have the same value of the maximum service rate, and correspondingly, the same value of arrival rate at the stability boundaries on the paths. In other words, for any two paths L_{K_1} and L_{K_2} on which q_1 is the most stable queue, $\hat{\mu}_1^{K_1}$ and $\hat{\mu}_1^{K_2}$ should be the same. The reasons are: firstly, all the other queues will behave the same after being unstable on the two paths because of the infinite queue length assumption of the unstable queue; and secondly, the unique system instability state, i.e., q_1 is the last one to become unstable on both L_{K_1} and L_{K_2}. Therefore, $\hat{\mu}_1^K$ will be a constant for all L_Ks on which q_i is the most stable queue and we let $\hat{\mu}_1^K = \alpha$. This implies $\lambda_1 = \alpha$ is the queue stability boundary of q_1 for all the paths on which q_1 is the most stable queue. Similar, we can have $\lambda_2 = \beta$ as the queue stability boundary of q_2 for all the paths on which q_2 is the most stable queue, and so on. Then, for each q_i, when the queue is the most stable queue, its queue stability boundary in the traffic space will be a n-dimensional hyperplane with formula $\lambda_i = y_i$, where y_i is a constant. Next, noting the meaning that if two queues are both the most stable queues on a path, they have the same stability boundary on the path, i.e., their queue stability boundaries intersect on a particular point on the path. This implies the intersections of those hyperplanes actually are the stability boundaries of some (as stable as) most stable queues. Because the traffic space is assumed Euclidean and the hyperplanes are linear, we can conclude that the instability region of all queues is an open n-dimensional cuboid such that $U = \{(\lambda_1, \lambda_2, ..., \lambda_n) \in \mathbb{R}^n : \alpha < \lambda_1, \beta < \lambda_2, ..., \gamma < \lambda_n\}$, where α, β, and γ are constants.

Now specifically consider the paths on which q_i and q_j are both the most stable queues. The above discussion implies the direction components of q_i and q_j on these paths satisfy $\frac{k_i}{y_i} = \frac{k_j}{y_j}$, where $\lambda_i = y_i$ and $\lambda_j = y_j$ are the stability boundaries given that q_i and q_j are the most stable queues, respectively. Let $\frac{y_i}{y_j} = C_{i,j}$. Then the hyperplane $H_1 : \frac{k_i}{k_j} = C_{i,j}$ is a hyperplane that satisfies the conclusion of the theorem when q_i and q_j are both the most stable queues.

Assume there are other path L_K on which $q_i \asymp q_j$ and $\frac{k_i}{k_j} = \hat{C} \neq C_{i,j}$. Now consider paths only on the hyperplane $H_2 : \frac{k_i}{k_j} = \hat{C}$. Because $\hat{C} \neq C_{i,j}$, hyperplanes H_1 and

H_2 will never intersect except at the axis. This implies on H_2 we can have any linear increasing paths but the ones on which both q_i and q_j are the most stable queues at the same time because all such paths are on H_1. This is obvious not true because q_i and q_j are already assumed as stable as each other on L_K and the selection of paths on H_2 is assumed arbitrary (Theorem 4.4.5). Consequently, such a L_K cannot exist, hence implying all the paths on which $q_i \asymp q_j$ can only be on H_1, i.e., $q_i \asymp q_j \Longrightarrow k_i/k_j = C_{i,j}$. This finishes the proof. □

A direct result from the above Lemma is that, in Type-1 SSMQSs there is one and only one path on which all the queues are as stable as one another. As we are going to consider situations that the systems are allowed to be reconfigured, the one and only one path claim only applies to a fixed system configuration and we state the fact in the following theorem.

Theorem 4.4.6 *In Type-1 SSMQSs, for a fixed system configuration there is one and only one linear increasing path L_K on which all the queues are as stable as one another, and L_K is given by*

$$\frac{k_1}{\hat{\mu}_1^K} = \frac{k_1}{\hat{\mu}_2^K} = ... = \frac{k_1}{\hat{\mu}_n^K},$$

where $\hat{\mu}_i^K$ is the (maximum) service rate of q_i when all the queues are unstable.

Proof: As stated in Lemma 4.4.1, the paths on which two given queues are as stable as each other can only be on a hyperplane. Then the intersection of these hyperplanes for any two queues is the path on which all the queues are as stable as one another. The existence and uniqueness of such a path is guaranteed by Theorem 4.4.5 and the Euclidean structure of the traffic space and it is easy to see that the intersection is a straight line L_K. Since on L_K all the queues will be unstable at the same time, the maximum service rate of each queue will then equals to the service rate of the queue in the unique instability state. Therefore, we can have the desired result and this finishes the proof. □

The following lemma gives a sufficient condition under which a queue is more (less) stable than another in the traffic space in Type-1 SSMQSs.

Lemma 4.4.2 *In Type-1 SSMQSs, on any given linear increasing path such that* $\frac{k_i}{k_j} \neq C_{i,j}$*, we have* $\frac{k_i}{k_j} > (<)\, C_{i,j} \Longrightarrow q_i \prec (\succ) q_j$*.*

Proof: As the two cases are symmetric, we prove only the case for $q_i \prec q_j$ and the other part can be proved similarly. Consider a partition of the traffic space, in which the paths satisfy $\frac{k_i}{k_j} > C_{i,j}$. First, note that we can always find a path in this partition that gives $q_i \prec q_j$ by setting k_j to a sufficiently small value. Second, we claim that either $q_i \prec q_j$ or $q_j \prec q_i$ holds for all the paths in this partition. If this is not the case, the two queues' stability boundaries should have at least one intersection in the partition. Then it implies that there is an increasing path in the partition on which the two queues are as stable as each other, but we know from Lemma 4.4.1 that this conclusion is not true as all such paths can only be in the hyperplane $\frac{k_i}{k_j} = C_{i,j}$. Hence, $q_i \prec q_j$ holds for all paths in this partition, i.e., $\frac{k_i}{k_j} > C_{i,j} \Longrightarrow q_i \prec q_j$. This finishes the proof. □

The above two lemmas are not "complete" in the sense that in Lemma 4.4.1 we do not know whether there are paths on the hyperplane $\frac{k_i}{k_j} = C_{i,j}$ can cause $q_i \prec (\succ) q_j$ or not. Consequently, Lemma 4.4.1 is only a necessary condition while Lemma 4.4.2 is only a sufficient condition. Before the above properties can be "completed", another property of the Type-1 SSMQSs is needed. Such property states that the relative stability of any two queues will not be affected by factors of any other queues. Until now, the properties of the Type-1 SSMQSs are given under the assumption that the system parameters of the queues are fixed. In other words, we only consider the system under a certain configuration. For example, in the polling system we considered fixed M_is for q_is, while in the ALOHA network we considered fixed p_is for q_is. However, to prove the relative stability of any two queues is independent to factors of other queues we must also consider situations that the system parameters of the queues can be changed on a given linear increasing path, e.g., for polling systems to allow M_is to change and for ALOHA network to allow p_is to change. For a given configuration of the system, we use $\mathcal{G}$ to denote the configuration of the queues, and $C_{i,j}(\mathcal{G})$ to denote the constant of the hyperplane which includes all the paths on which $q_i \asymp q_j$ for the configuration $\mathcal{G}$. By changing the configuration of the system for a given path we mean that first to consider the system on the path under the original configuration $\mathcal{G}_1$, then to consider the system again on the same path with another configuration $\mathcal{G}_2$. Note that a system is not necessarily be reconfigurable, e.g., the processor sharing system is not configurable.

Lemma 4.4.3 *In reconfigurable Type-1 SSMQSs, the ratio of any two queues' direction components on a given path is independent to any other queues' configurations in the sense that if $k_i/k_j \gtreqless C_{i,j}(\mathcal{G}_1)$ for configuration $\mathcal{G}_1$, then for any other configuration $\mathcal{G}_2$ (without changing q_i and q_j's settings), we also have $k_i/k_j \gtreqless C_{i,j}(\mathcal{G}_2)$, and $C_{i,j}(\mathcal{G}_1) = C_{i,j}(\mathcal{G}_2)$.*

Proof: Let q_i, q_j, and q_k be the queues in concern. We first consider the system under configuration $\mathcal{G}_1$. For $\mathcal{G}_1$, from Lemma 4.4.1 we know that there is a constant $C_{i,j}(\mathcal{G}_1)$ for q_i and q_j. Divide all the linear increasing paths into three subsets, namely, $\mathcal{A}(\mathcal{G}_1) = \{L_k | k_i/k_j = C_{i,j}(\mathcal{G}_1)\}$, $\mathcal{B}_1(\mathcal{G}_1) = \{L_k | k_i/k_j > C_{i,j}(\mathcal{G}_1)\}$, and $\mathcal{B}_2(\mathcal{G}_1) = \{L_k | k_i/k_j < C_{i,j}(\mathcal{G}_1)\}$. Now reconfigure the system to $\mathcal{G}_2$ by changing the parameters of q_k. For configuration $\mathcal{G}_2$ we also have a constant $C_{i,j}(\mathcal{G}_2)$ and subsets of paths $\mathcal{A}(\mathcal{G}_2)$, $\mathcal{B}_1(\mathcal{G}_2)$, and $\mathcal{B}_2(\mathcal{G}_2)$. If $C_{i,j}(\mathcal{G}_1) = C_{i,j}(\mathcal{G}_2)$, because of the linear structure of the Hyperplanes in the traffic space, the property is true. Now assume $C_{i,j}(\mathcal{G}_1) \neq C_{i,j}(\mathcal{G}_2)$, and without loss of generality, further assume $C_{i,j}(\mathcal{G}_1) > C_{i,j}(\mathcal{G}_2)$. Then it is easy to see that $\mathcal{A}(\mathcal{G}_1) \subset \mathcal{B}_1(\mathcal{G}_2)$, $\mathcal{B}_1(\mathcal{G}_1) \subset \mathcal{B}_1(\mathcal{G}_2)$, and $\mathcal{A}(\mathcal{G}_2) \cup \mathcal{B}_2(\mathcal{G}_2) \subseteq \mathcal{B}_2(\mathcal{G}_1)$. Specifically, all the paths in the subset of $\mathcal{B}_2(\mathcal{G}_1)$ will be distributed to the subsets $\mathcal{A}(\mathcal{G}_2)$, $\mathcal{B}_1(\mathcal{G}_2)$, and $\mathcal{B}_2(\mathcal{G}_2)$. Also, all the paths in subsets $\mathcal{A}(\mathcal{G}_1)$ and $\mathcal{B}_1(\mathcal{G}_1)$ under configuration $\mathcal{G}_1$ will be in the subset of $\mathcal{B}_1(\mathcal{G}_2)$ under configuration $\mathcal{G}_2$. This implies that for all the paths on which $q_i \asymp q_j$ in the configuration $\mathcal{G}_1$, the $q_i \asymp q_j$ relation cannot be kept in the configuration $\mathcal{G}_2$. Similar conclusion can be drawn if we change the configuration $\mathcal{G}_2$

back to $\mathcal{G}_1$, i.e., for all the paths on which $q_i \asymp q_j$ in the configuration $\mathcal{G}_2$, the $q_i \asymp q_j$ relation cannot be kept in the configuration $\mathcal{G}_1$. Now for both configurations $\mathcal{G}_1$ and $\mathcal{G}_2$, if we project the n-dimension traffic space into a $(n-1)$-dimension traffic space by letting $\lambda_k = 0$, in the $(n-1)$-dimension traffic space we will have two sets of paths on which $q_i \asymp q_j$ and all the queues are having the same settings. The two sets of paths are $\{L_k | k_i/k_j = C_{i,j}(\mathcal{G}_1)\}$ and $\{L_k | k_i/k_j = C_{i,j}(\mathcal{G}_2)\}$, and $C_{i,j}(\mathcal{G}_1) \neq C_{i,j}(\mathcal{G}_2)$. This contradicts to Lemma 4.4.1 that if on paths q_i and q_j have relation $q_i \asymp q_j$ then k_i/k_j must be a constant. Therefore, for any configuration $\mathcal{G}_1$ and q_i, q_j, and q_k, a reconfiguration of the system $\mathcal{G}_2$ by changing q_k's setting and having $C_{i,j}(\mathcal{G}_1) > C_{i,j}(\mathcal{G}_2)$ cannot exist. Similarly, a reconfiguration that causes $C_{i,j}(\mathcal{G}_1) < C_{i,j}(\mathcal{G}_2)$ cannot exist as well. The only possible case will then be $C_{i,j}(\mathcal{G}_1) = C_{i,j}(\mathcal{G}_2)$. Because q_k is selected arbitrarily, the conclusion is true if we select any other queue (rather than q_i and q_j). Now for any initial configuration $\mathcal{G}_1$, and final configuration $\mathcal{G}_2$, besides q_i and q_j, if there are more than one queue has different settings, we can change the setting of one queue at a time and this will not affect the ratio of k_i/k_j. In this way, we can have the final configuration $\mathcal{G}_2$ and still have the desired result, i.e., $C_{i,j}(\mathcal{G}_1) = C_{i,j}(\mathcal{G}_2)$. This finishes the proof. □

With Lemma 4.4.3 we can now have the sufficient part of Lemma 4.4.1 and the necessary part of Lemma 4.4.2 in the next Lemma.

Lemma 4.4.4 *In Type-1 SSMQSs, a given linear increasing path L_K satisfies $k_i/k_j = C_{i,j}$ implies $q_i \asymp q_j$ on the path; on the other hand $q_i \prec (\succ) q_j$ on a path implies the path satisfies $k_i/k_j > (<)\, C_{i,j}$.*

Proof: For the first part, we know that there must be a L_K on which $q_i \asymp q_j$ and $k_i/k_j = C_{i,j}$. Now assume there are also paths $\bar{L}_K$ on which $q_i \prec q_j$ and $k_i/k_j = C_{i,j}$ (or $q_i \succ q_j$ and $k_i/k_j = C_{i,j}$). Because change any queues' settings will not affect the k_i/k_j values (Lemma 4.4.3). We can project the n-dimension space to a 2-dimension space by letting $\lambda_k = 0$ for all q_k, where $q_k \neq q_i, q_j$. Then in the 2-dimension space we have the path $k_i/k_j = C_{i,j}$ on which both $q_i \asymp q_j$ and $q_i \prec q_j$ (or $q_i \succ q_j$) hold. This obviously is not true. Hence the first part of the proposition is true. Since now the condition of $k_i/k_j = C_{i,j}$ is sufficient and necessary for $q_i \asymp q_j$ on a path, it implies $q_i \prec (\succ) q_j \implies k_i/k_j > (<) C_{i,j}$, i.e., the second part of the proposition is also true. This finishes the proof. □

Now we are ready to present the relative stability condition of the Type-1 SSMQSs, which is the combination of Lemmas 4.4.1, 4.4.2, 4.4.3, and 4.4.4.

Theorem 4.4.7 *(Relative Stability Condition of Type-1 SSMQSs)*
In Type-1 SSMQSs, the sufficient and necessary condition for any two queues q_i and q_j to have $q_i \asymp q_j$ on a given linear increasing path is that the direction components of the two queues on the path satisfy $k_i/k_j = C_{i,j}$, where $C_{i,j}$ is a universal constant and independent to any other queues. Furthermore, the sufficient and necessary condition for $q_i \prec (\succ) q_j$ on a path is $\frac{k_i}{k_j} > (<) C_{i,j}$.

Proof: The theorem is the direct result of Lemmas 4.4.1, 4.4.2. 4.4.3, and 4.4.4. □

Theorem 4.4.7 says that the relative stability of two queues in Type-1 SSMQSs will not be affected by other queues as long as the two queues' arrival rates satisfy some conditions. The intuition behind the property is, firstly, the interaction between the two queues is constrained as they will have unique instability state when all the queues are unstable. Secondly, no matter what effects the rest of the queues can bring to the two queues, the effect should affect them equally. As $C_{i,j}$ is a constant, it can be determined by just finding one linear increasing path L_K on which $q_i \asymp q_j$. On such a L_K, $C_{i,j} = \hat{\mu}_i^K / \hat{\mu}_j^K$, where $\hat{\mu}_i^K$ and $\hat{\mu}_j^K$ are the maximum service rates of q_i and q_j, respectively. Obviously, the path given in Theorem 4.4.6 is a good candidate.

As this point, we have solved the major problems of relative stability in Type-1 SSMQSs. Namely, on one hand, for any given linear increasing path, we are able to tell the relative stability relations of any two queues, and consequently, the stability ordering on the path; on the other hand, for any given relative stability relation of any two queues, (as well as the stability ordering of all the queues), we are able to tell what kind of condition a linear increasing path should satisfy in order to obtain the given relative stability relation of the two queues, (and the stability ordering of all the queues).

Here, two points are worth to mention. First, from Lemma 4.4.3 we know that if we want to change the relative stability relation of q_i and q_j on a path, (or the ratio of k_i/k_j), the only way is to reconfigure the settings of either q_i or q_j, or both queues. Second, in Type-1 SSMQSs, the paths on which two queues are as stable as each other are unique in the sense that these paths form a $(n-1)$-dimension hyperplane in the traffic space.

In the next theorem we give another important property of the Type-1 SSMQSs, which is a result of the Theorem 4.4.6.

Theorem 4.4.8 *In reconfigurable Type-1 SSMQSs, for a given linear increasing path L_K and a traffic point Λ on the path such that if under configuration $\mathcal{G}_0$ the system is stable at Λ, then there exists a reconfiguration of the system $\mathcal{G}_1$ such that on L_K all the queues are as stable as one another, and the system is also stable at Λ.*

Proof: There is nothing to prove if the queues are already as stable as one another on L_K. Assume that there are at least two queues are not as stable as each other on the path under $\mathcal{G}_0$. Now consider a new configuration $\mathcal{G}_1$. Recall that the condition given in Theorem 4.4.6, i.e., $\frac{k_1}{\hat{\mu}_1^K} = \frac{k_1}{\hat{\mu}_2^K} = ... = \frac{k_1}{\hat{\mu}_n^K}$, which specifies the path on which all the queues are as stable as one another, and $\hat{\mu}_i^K$ is the maximum service rate received by q_i at the unique system instability state. Then the theorem is true if $\Lambda == (\lambda_1, \lambda_2, ..., \lambda_n) = (\hat{\mu}_1^K, \hat{\mu}_2^K, ..., \hat{\mu}_n^K)$, where $\hat{\mu}_i^K$ is the maximum service rate of q_i on L_K under $\mathcal{G}_1$, i.e., Λ is the stability boundary for all queues on L_K under configuration $\mathcal{G}_1$. To find out $\mathcal{G}_1$, let $\hat{\mu}_i^K = f_i(\mathcal{G}_1)$, where f_i is the function to calculate the maximum service rate of q_i on a given path for the system configuration $\mathcal{G}_1$. Then to find out $\mathcal{G}_1$ is equivalent to solve the set of equations:

$$\begin{cases} \lambda_1 = f_1(\mathcal{G}_1), \\ \lambda_2 = f_2(\mathcal{G}_1), \\ \cdots\cdots\cdots\cdots \\ \lambda_n = f_n(\mathcal{G}_1). \end{cases} \tag{4.3}$$

We argue the target configuration $\mathcal{G}_1$ is solvable for the following reasons. First, there are stable reconfigurations exist at Λ. This is because the server's capacity allows (the system is stable at Λ), which implies that for each q_i a better configuration exists in the sense that under such a configuration q_i may still be stable once the traffic point passes beyond Λ. For example, $\mathcal{G}_0$ may be such a better configuration. Second, the traffic point Λ is given, which means in Eq. 4.3 there are totally k unknowns in a set of k equations, assume that there is only one system parameter for each queue . Lastly, as we are going to solve the configuration such that Λ is the stability boundary for all the queues, the function $f_i(\mathcal{G}_1)$ is independent to Λ because we can consider all the queues as if they are already unstable. Therefore, $\mathcal{G}_1$ is solvable. Based on the way of finding the configuration $\mathcal{G}_1$, we conclude that under $\mathcal{G}_1$ all the queues will be as stable as one another on L_K and have Λ as their stability boundaries, i.e., stable at Λ. This finishes the proof. □

Theorem 4.4.8 is very useful as it allows us to characterize the stability region of Type-1 SSMQSs. Furthermore, it can also serve as the criterion to stabilize a Type-1 SSMQS, and to obtain the maximum stable throughput of the system on a given linear increasing path. To see this, we first define the following concept of the *maximum as stable as configuration* on a given path.

Definition 4.4.4 *(Maximum as stable as configuration)*

In reconfigurable Type-1 SSMQSs, for a given linear increasing path L_K, a configuration of the system is called the maximum as stable as configuration, denoted as $\mathcal{G}_m(L_K)$, if under $\mathcal{G}_m(L_K)$ all the queues are as stable as one another on L_K

and the stability boundary of the queues on L_K, denoted as $\Lambda_m(L_K)$, is the maximum in the sense that for any other configuration of the system that can make the queues be as stable as one another and has stability boundary $\Lambda(L_K)$ on L_K, we have $\|\Lambda(L_K)\| \leq \|\Lambda_m(L_K)\|$.

Now for a given path L_K and a system configuration $\mathcal{G}$, denote the system stability boundary as $\Lambda(L_K, \mathcal{G})$, then the stability region of the system on the path, denoted as $\mathfrak{S}(L_K, \mathcal{G})$, will be $\mathfrak{S}(L_K, \mathcal{G}) = \{\Lambda | \Lambda \in L_K, \|\Lambda\| < \|\Lambda(L_K, \mathcal{G})\|\}$. Let $\mathfrak{D}(L_K)$ be the closure of $\mathfrak{S}(L_K, \mathcal{G})$ for L_K and all possible $\mathcal{G}$, i.e., $\mathfrak{D}(L_K) = \cup_{\mathcal{G}} \mathfrak{S}(L_K, \mathcal{G})$. Then the set of $\mathfrak{D}(L_K)$ can be interpreted as the overall system stability region on L_K because for any $\Lambda \in \mathfrak{D}(L_K)$, there exists a configuration of the system such that at Λ the system is stable. Now it can see that Theorem 4.4.8 actually suggests that, on a given path L_K, we have $\mathfrak{D}(L_K) = \{\Lambda | \Lambda \in L_K, \|\Lambda\| < \|\Lambda_m(L_K)\|\}$. If $\mathfrak{D}$ is the overall stability region in traffic space, i.e., $\mathfrak{D} = \cup_{L_K} \mathfrak{D}(L_K)$, then we have $\mathfrak{D} = \cup_{L_K} \{\Lambda | \Lambda \in L_K, \|\Lambda\| < \|\Lambda_m(L_K)\}$. We state this conclusion in the following corollary.

Corollary 4.4.1 *In reconfigurable Type-1 SSMQSs, we have*

$$\mathfrak{D} = \cup_{L_K} \{\Lambda | \Lambda \in L_K, \|\Lambda\| < \|\Lambda_m(L_K)\|\} \tag{4.4}$$

Proof: To prove the corollary is equivalent to prove that for any given L_K the following holds:

$$\mathfrak{D}(L_K) = \{\Lambda | \Lambda \in L_K, \|\Lambda\| < \|\Lambda_m(L_K)\|\}.$$

First note that on L_K for the maximum as stable as configuration $\mathcal{G}_m(L_K)$, we have $\mathfrak{S}(L_K, \mathcal{G}_m) \subseteq \mathfrak{D}(L_K)$ according to the definition of $\mathcal{G}_m(L_K)$ and $\mathfrak{D}(L_K)$. Then for each $\mathfrak{S}(L_K, \mathcal{G})$, Theorem 4.4.8 implies that there exists a $\mathcal{G}_0$ such that on L_K all the queues are as stable as one another and $\mathfrak{S}(L_K, \mathcal{G}) \subseteq \mathfrak{S}(L_K, \mathcal{G}_0)$. Because $\mathfrak{S}(L_K, \mathcal{G}_0) \subseteq \mathfrak{S}(L_K, \mathcal{G}_m)$ as $\mathcal{G}_m$ is the maximum as stable as configuration on L_K, and noting that $\mathcal{G}$ is arbitrarily selected, we have $\cup_{\mathcal{G}} \mathfrak{S}(L_K, \mathcal{G}) \subseteq \mathfrak{S}(L_K, \mathcal{G}_m)$, or equivalently, $\mathfrak{D}(L_K) \subseteq \mathfrak{S}(L_K, \mathcal{G}_m)$. Hence, $\mathfrak{D}(L_K) = \{\Lambda | \Lambda \in L_K, \|\Lambda\| < \|\Lambda_m(L_K)\|\}$, and this finishes the proof. □

Because $\mathcal{G}_m(L_K)$ is an as stable as configuration of the system on L_K, the value of $\Lambda_m(L_K)$ can be easily obtained. Therefore, Corollary 4.4.1 suggests that to characterize the overall system stability region for Type-1 SSMQSs is equivalent to find the maximum as stable as configuration on a linear increasing path of the systems.

Now consider the stabilization problem of a Type-1 system: for a given traffic point Λ, whether there exists a configuration that can stabilize the system at Λ. The existence of such a configuration implies $\Lambda \in \mathfrak{O}$. Therefore, according to Corollary 4.4.1, to find whether such a configuration exists or not is equivalent to consider the maximum as stable as configuration $\mathcal{G}_m(L_K)$ on the linear increasing path L_K that passes the origin and Λ. If $\Lambda \in \mathfrak{S}(L_K, \mathcal{G}_m)$, the system can be stabilized at Λ; otherwise, the system can never be stable at Λ.

Another related problem is to find the maximum stable throughput on a given path L_K. From Corollary 4.4.1, it is easy to see that the maximum stable throughput of the system can be achieve at $\Lambda_m L_K$. Therefore, to find the maximum stable throughput on L_K is also equivalent to find $\mathcal{G}_m(L_K)$.

From the above discussion, with Theorem 4.4.8 and Corollary 4.4.1, we can see that in Type-1 SSMQSs, the characterization problem of the overall system stability region, the stabilization problem of the system, and finding the maximum stable throughput can all be reformulated as an optimization problem, that is, to find the maximum as stable as configuration on a linear increasing path.

Another usage of Theorem 4.4.8 and Corollary 4.4.1 is finding a necessary queue stability condition for a given queue when it is the least stable queue. Consider Eq. (4.3) and fix λ_2, λ_3, ..., λ_k and the system parameter associated with q_1, we have a set of k equations with k unknowns, i.e., the system parameters associated with each queue. Now finding the maximum as stable as configuration in terms of λ_1 is equivalent to finding the maximum stability boundary of q_1 with given arrival rates of the other queues, i.e., $(\lambda_2, \lambda_3, ..., \lambda_k)$. Denote this maximum stability boundary of q_1 as $\hat{\lambda_1}$. According to Corollary 4.4.1, the queue stability boundary of q_1 in the original system configuration with the given λ_2, λ_3, ..., λ_k must be less than or equals to $\hat{\lambda_1}$.

Because, for the given λ_2, λ_3, ..., λ_k, $\hat{\lambda_1}$ is the maximum stability boundary when considering q_1 has a fixed associated system parameter while all other queues can have any possible associated system parameters. Therefore, $\hat{\lambda_1}$ is a necessary queue stability condition of q_1 when given λ_2, λ_3, ..., λ_k in the original system configuration. A special case is that for q_1 with a fixed associated system parameter, if there is only one as stable as configuration on a given path, then the necessary queue stability condition is also sufficient. We will demonstrate this point for the polling system in the next chapter. Also, in the next chapter, we will use this method to obtain a necessary stability condition for the slotted buffered ALOHA network and show that the new condition is better than the existing one obtained in [47].

It is worth to mention that the above results only hold for reconfigurable Type-1 SSMQSs. For non-configurable Type-1 SSMQSs, the conclusions in general are not necessarily true.

To end this section, we provide two examples of Type-2 systems and discuss some properties of them.

The first example is a polling system with all queues having exhaustive service policy. We called this system E_1. In E_1, once any queue becomes unstable, it will occupy the server forever. This makes the cycle time become infinite. Consequently, the random sequence $H^n(\mathbf{W}^n)$ defined in Definition 4.2.2 will be non-stationary and non-ergodic. Only when all the queues are stable, then $H^n(\mathbf{W}^n)$ is stationary and ergodic. The queues in E_1 are of capture type. On any given linear increasing path, the maximum service rate of a queue is a constant, which equals to the server's capacity. However, a queue may not able to achieve this maximum service rate once another queue captures the server and becomes unstable. Therefore, the queues do not have guaranteed service share because a queue may receive 0 service. Furthermore, the queues have variable service rates when all queues are unstable and this depends on which queue can capture the server. In E_1, any queue becomes unstable will cause all other queues become unstable. In this sense, there is only one possible relative stability relation for all queues on any paths, that is, the as stable as relation. In other words, the relative stability relation of the queues is independent to the paths. However, one should note that the instability of all the queues in this case is caused by service starvation.

For the second example, consider a two queues polling system with gated limited service. Let m_1, m_2, and m be three positive integers. In the system, let the maximum number of customers that can be served at each queue determined in the following way: if the queue length of q_i is less than or equal to m_i when the server visits q_i, then the server can serve all the customers during the visit; if the queue length of q_i is more than m_i when the server visits q_i, the server can serve at most $m_i + m$ customers. During any cycle, the server cannot serve more than $m_1{+}m{+}m_2$ customers. In addition, the maximum number of customers that can be served at a queue during is not decreasing, i.e., not less than the maximum number of customers that can be served in previous visits. We called this system E_2. One can see that the difference between E_2 and the polling system we considered in Chapter 3 is that in E_2 the maximum number of customers that can be served at a queue by the server during a cycle varies (variable-limit), while the polling system in Chapter 3 has fixed service limits at the queues (fixed-limit). For E_2, when any queue becomes unstable, the time that the sever spends at that queue is a stationary mixture and therefore not ergodic. Hence, E_2 is also a Type-2 SSMQS. However, in the contrast to E_1, queues in E_2 are of non-capture type and have guaranteed service share, i.e., for q_i at least m_i per cycle. Furthermore, in E_2, the maximum service rate of a queue on a given path varies, depends on the share it has contended previously. For E_2, only for some paths we can tell the relative stability relation of the queues, for example, $k_1/m_1 > k_2/(m_2{+}m)$ on which $q_1 \prec q_2$, and $k_2/m_2 > k_1/(m_1{+}m)$ on which $q_1 \succ q_2$. For other paths, the relative stability relation cannot be told even when the path is given, though paths for each of the three relative stability relation exists. Consequently, the relative stability of the queues in E_2 is also not directly dependent to the linear increasing path. Lastly, in E_1, the paths on which any two queues are as stable as each other are not unique.

From the above examples, we can see that Type-2 SSMQSs do not have the “good” properties as the Type-1 SSMQSs.

4.5 Summary

In this chapter we first identified the SSMQSs into two kinds. This allows us to use Loynes’ theorem to define the concept of degree of stability of a queue. For the Type-1 SSMQSs, we defined three relative stability relations between any two queues on a path. Then we study the properties with respect to relative stability of Type-1 SSMQSs. In particular, these properties include the relative stability conditions for a specific relative stability relation among the queues. As we have seen, the conditions are closely related to how the system traffic changes in the traffic space. In the next chapter we are going to apply these properties in analyzing both the relative and absolute stability for some practical SSMQSs. As we will see, these properties allow us to develop a unified, intuitive, yet simple approach to study the stability issues in those systems and to obtain new results.

5. APPLICATIONS

In this chapter we use the relative stability results of Type-1 SSMQSs developed in Chapter 4 to study stability related issues for some SSMQSs. We start with the relative stability conditions of the SSMQSs. Then we study the absolute stability conditions of the systems. The results include both queue stability conditions and system stability conditions. For systems whose exact stability conditions cannot be obtained, we provide a simple method for the necessary condition of the system stability. Lastly, we consider the characterization of the system stability region. Some of the results in this chapter had been reported previously, while some are new. Nevertheless, through the analysis we can see that the relative stability results indeed provides us a unified, intuitive, and simpler method not only able to reproduce most of the previous results but also able to derive new ones. More importantly, our approach applies to not just one particular one system but all the Type-1 SSMQSs.

5.1 The Models

In this chapter, we select four SSMQSs to study. Two of them are the polling system with gated limited service and the slotted buffered ALOHA network, which have been described in Chapter 3. The other two systems are: wireless network protocols dubbed *network-assisted diversity multiple access* (NDMA) and *blind* NDMA (BNDMA) [23], and the slotted buffered ALOHA with multipacket reception [53]. We have already seen that both the polling system and the ALOHA network belong to Type-1 SSMQSs. In the follows, we describe the other two models.

5.1.1 NDMA and BNDMA

The NDMA and BNDMA protocols were proposed in [71, 78], respectively, and their system stability were studied in [23]. Both of the protocols were proposed to address the collision resolution problem in wireless communication networks. The idea of the protocols is "to generate diversity via immediate simultaneous retransmissions of all collided packets induced by the medium access control layer protocol" [23]. Practically, if M packets collide totally M times, given that the resultant M linear mixtures of the collision of the original packets are linearly independent, then one station which collects these mixtures may able to recover the original packets by solving the associated linear system. To achieve that, it requires the original packets contain additional known prefixes that enable detection and estimation of the mixing matrix. The resultant protocol is the NDMA. A variation is that the set of linear equations can be solved blindly, given that one more collision is provided and a certain type of packet phase modulation is employed at the transmitters. This variation is the BNDMA. More details of the two protocols are referred to [23, 71, 78].

Now assume there are one server and k queues in the system which employs either NDMA or BNDMA protocols. Each queue has unlimited buffer to store incoming packets. The arrival process to q_i is Poisson with arrival rate λ_i, and the arrivals at all queues are mutually independent. The transmissions are slotted, which duration equals to the transmission time of a packet. All the transmissions at the queues are synchronized at the beginning of a slot, given that the queues are not empty. In NDMA, a M-fold collision requires $(M-1)$ retransmissions, while in BNDMA it requires M retransmissions. The slots used for the first transmission and subsequent retransmission comprise a *collision resolution* (CR) cycle. Let $(\mathbf{Q}^n = (Q_1^n, Q_2^n, ..., Q_k^n))_{n=1}^{\infty}$ be the joint queue length process of the queues at the beginning of slot n. Then for each q_i, the following holds:

$$Q_i^{n+1} = \begin{cases} Q_i^n - 1 + A_i^n, & Q_i^n > 0 \\ A_i^n, & Q_i^n = 0 \end{cases} \tag{5.1}$$

where A_i^n is the number of new arrivals to q_i during the n slot and has a mean

$$\text{NDMA:} \quad EA_i^n = \lambda_i \left[\sum_{j=1}^{k} \mathbb{I}\{Q_i^n > 0\} + \delta(\sum_{j=1}^{k} Q_i^n) \right], \tag{5.2}$$

or

$$\text{BNDMA:} \quad EA_i^n = \lambda_i \left[\sum_{j=1}^{k} \mathbb{I}\{Q_i^n > 0\} + 1 \right], \tag{5.3}$$

respectively. Here $\delta(x)$ is the Kronecker delta function, i.e., $\delta(0)=1$, and $\delta(x)=0$ for other values of x. It can be shown that the above joint queue length process $\mathbf{Q}^n$ is a homogeneous, irreducible, and aperiodic Markov chain [23]. In addition, the arrival processes and the service policies in both protocols satisfy the conditions of Theorem 4.2.1, therefore, both NDMA and BNDMA are Type-1 SSMQSs. Also note that both NDMA and BNDMA are non-configurable.

5.1.2 A Slotted Buffered ALOHA Network with Multipacket Reception

The multipacket reception (MPR) model is similar to the slotted buffered ALOHA network considered in Chapter 3. However, this model is more general that it allows the receiver to receive multiple packets simultaneously [53]. Let $\mathcal{K}$ be the set of all queues. During a slot, if there is a subset of queues $\mathcal{S} \subseteq \mathcal{K}$ transmit packets, then define $\mathcal{R} \subseteq \mathcal{S}$ as the subset of queues whose packets can be successfully received. In particular, define the conditional probability $q_{\mathcal{R},\mathcal{S}}$ as

$$q_{\mathcal{R},\mathcal{S}} = P(\text{only packets from } \mathcal{R} \text{ are successfully received } | \mathcal{S} \text{ transmits}). \tag{5.4}$$

The packet receptions are independent from slot to slot. Furthermore, the marginal probability of success $\mathcal{R}$ given the set $\mathcal{S}$ is then

$$q_{\mathcal{R}|\mathcal{S}} = \sum_{\mathcal{V}:\mathcal{R}\subseteq\mathcal{V}\subseteq\mathcal{S}} q_{\mathcal{R},\mathcal{S}}. \tag{5.5}$$

For example, if there are two queues in the system, we have

$$q_{\{1\},\{1\}} = P(q_1 \text{ transmits successfully } | \text{ only } q_1 \text{ transmits}),$$

$$q_{\{1\},\{1,2\}} = P(q_1 \text{ transmits successfully} \mid \text{both } q_1 \text{ and } q_2 \text{ transmit}),$$

$$q_{\{1,2\},\{1,2\}} = P(q_1 \text{ and } q_2 \text{ transmit successfully} \mid \text{both } q_1 \text{ and } q_2 \text{ transmit}).$$

The marginal successful transmission probability for q_i, $i=1,2$, will be

$$q_{\{i\}|\{i\}} = q_{\{i\},\{i\}},$$

and

$$q_{\{i\}|\{1,2\}} = q_{\{i\},\{1,2\}} + q_{\{1,2\},\{1,2\}}.$$

For the MPR model, we use the same assumptions as in the ALOHA network described in Chapter 3. Then the joint queue length process $(\mathbf{Q}^n = (Q_1^n, Q_2^n, ..., Q_k^n))_{n=1}^{\infty}$ at the beginning of slot n is a Markov chain. If $q_{\{i\}|\{i\}} > 0$, then the chain is irreducible and aperiodic [53]. Furthermore, for each q_i, the following holds:

$$Q_i^{n+1} = (Q_i^n - B_i^n)^+ + A_i^n, \tag{5.6}$$

where B_i^n represents whether there is a departure from q_i and A_i^n is the number of new arrivals to q_i during slot n. Based on Theorem 4.2.1, we can see that the MPR model also belongs to the Type-1 SSMQSs.

5.2 Relative Stability of Type-1 Systems

In this section we establish the relative stability for the four Type-1 SSMQSs. More precisely, for all the four systems, we provide the sufficient and necessary conditions of the relative stability relation for any two queues on a linear increasing path. Furthermore, we also give the sufficient and necessary conditions of the stability ordering of all queues on a path. As we have shown in Chapter 3 and Section 1 of this chapter, all the four systems considered in this chapter are of Type-1 SSMQSs. Therefore, the relative stability results are the direct applications of Theorem 4.4.7 and Theorem 4.4.6. The relative stability results for the polling system and the ALOHA network have been reported previously [16, 17, 29, 47], while for the other two models the results presented here are new.

5.2.1 The Polling System with Gated Limited Service

Consider the polling system described in Chapter 3. According to Theorem 4.4.7, the relative stability relations of any two queues q_i and q_j only depend on the ratio of their direction components and the constant $C_{i,j}$. The constant $C_{i,j}$ can be obtained through any paths on which $q_i \asymp q_j$ as long as the configurations of the two queues remain unchanged. To compute the $C_{i,j}$, we select the linear increasing path given by Theorem 4.4.6 on which all the queues are as stable as one another. In the following theorem we provide the relative stability relation of any two queues in the polling system.

Theorem 5.2.1 *In the polling system with gated limited service policy, for a given configuration of the system,* $q_i \succeq q_j \iff \frac{k_i}{k_j} \leq \frac{M_i}{M_j}$ *on the linear increasing path* L_K*.*

Proof: We first consider the linear increasing path L_A on which all queues are as stable as one another. From Theorem 4.4.6, we know on this path

$$\mu_i^A = \frac{M_i}{\sum_{l=1}^k M_l},$$

and

$$\mu_j^A = \frac{M_j}{\sum_{l=1}^k M_l}.$$

Because we have $q_i \asymp q_j$ on L_A, therefore $k_i/\mu_i^A = k_j/\mu_j^A$ and it is easy to see that $C_{i,j} = \mu_i^A/\mu_j^A = M_i/M_j$. Then, according to Theorem 4.4.7, we have $q_i \succeq q_j \iff \frac{k_i}{k_j} \leq \frac{M_i}{M_j}$ on any path L_K. This finishes the proof. □

The stability ordering in the polling system is an immediate result of the above theorem.

Corollary 5.2.1 *In the polling system with gated limited service policy, for a given configuration, the stability ordering of the queues* $q_{(1)} \succeq q_{(2)} \succeq ... \succeq q_{(k)} \iff \frac{k_{(1)}}{M_{(1)}} \leq \frac{k_{(2)}}{M_{(2)}} \leq ... \leq \frac{k_{(k)}}{M_{(k)}}$ *on the linear increasing path* L_K*, where* $(1), (2), ...(k)$ *is a permutation of the sequence* $1, 2, ..., k$*.*

Proof: The conclusion is a direct result of Theorem 5.2.1. □

5.2.2 The Slotted Buffered ALOHA Network

Consider the ALOHA network described in Chapter 3. Applying the same methods in the last subsection, we have relative stability conditions for the ALOHA networks in the following theorem and corollary.

Theorem 5.2.2 *In the slotted buffered ALOHA system, for a given configuration of the system,* $q_i \succeq q_j \iff \frac{k_i(1-p_i)}{p_i} \leq \frac{k_j(1-p_j)}{p_j}$ *on the linear increasing path* L_K*.*

Proof: We consider again the linear increasing path L_A on which all the queues are as stable as one another. From Theorem 4.4.6, we know on this path

$$\mu_i^A = p_i \prod_{l \neq i} (1 - p_l),$$

and

$$\mu_j^A = p_j \prod_{l \neq j} (1 - p_l).$$

Because we have $q_i \asymp q_j$ on L_A, therefore $k_i/\mu_i^A = k_j/\mu_j^A$ and it is easy to see that $C_{i,j} = \mu_i^A/\mu_j^A = (p_i/(1-p_i))/(p_j/(1-p_j))$. Then, according to Theorem 4.4.7, we have $q_i \succeq q_j \iff \frac{k_i(1-p_i)}{p_i} \leq \frac{k_j(1-p_j)}{p_j}$ on any path L_K. This finishes the proof. □

The stability ordering in the ALOHA network is given in the following corollary.

Corollary 5.2.2 *In the slotted buffered ALOHA network, for a given configuration, the stability ordering of the queues* $q_{(1)} \succeq q_{(2)} \succeq ... \succeq q_{(k)} \iff \frac{k_{(1)}(1-p_{(1)})}{p_{(1)}} \leq \frac{k_{(2)}(1-p_{(2)})}{p_{(2)}} \leq ... \leq \frac{k_{(k)}(1-p_{(k)})}{p_{(k)}}$ *on the linear increasing path* L_K*, where* $(1), (2), ...(k)$ *is a permutation of the sequence* $1, 2, ..., k$*.*

Proof: The conclusion is a direct result of Theorem 5.2.2. □

5.2.3 NDMA and BNDMA

Now consider the NDMA and BNDMA protocols. Applying the same approach in the last two subsections, we have relative stability conditions for the protocols in the following theorem and corollary.

Theorem 5.2.3 *In the NDMA and BNDMA protocols, $q_i \succeq q_j \iff k_i \leq k_j$ on the linear increasing path L_K.*

Proof: We consider again the linear increasing path L_A on which all the queues are as stable as one another. For the NDMA protocol, from Theorem 4.4.6, we know on this path

$$\mu_i^A = \frac{1}{k},$$

and

$$\mu_j^A = \frac{1}{k}.$$

For the BNDMA on path L_A we have

$$\mu_i^A = \frac{1}{k+1},$$

and

$$\mu_j^A = \frac{1}{k+1}.$$

Because we have $q_i \asymp q_j$ on L_A, therefore $k_i/\mu_i^A = k_j/\mu_j^A$ and it is easy to see that for both NDMA and BNDMA $C_{i,j} = \mu_i^A/\mu_j^A = 1$. Then, according to Theorem 4.4.7, we have $q_i \succeq q_j \iff k_i \leq k_j$ on any path L_K. This finishes the proof. □

The stability ordering in the NDMA and BNDMA protocols is given in the following corollary.

Corollary 5.2.3 *In both NDMA and BNDMA protocols, the stability ordering of the queues $q_{(1)} \succeq q_{(2)} \succeq ... \succeq q_{(k)} \iff k_{(1)} \leq k_{(2)} \leq ... \leq k_{(k)}$ on the linear increasing path L_K, where $(1), (2), ...(k)$ is a permutation of the sequence $1, 2, ..., k$.*

Proof: The conclusion is a direct result of Theorem 5.2.3. □

5.2.4 The MPR Model

Now consider the MPR model. Before we proceed, we introduce the notation of $\vec{\mathfrak{p}_i}$ and $\vec{\mathfrak{q}_i}$ for q_i in the MPR model. Both $\vec{\mathfrak{p}_i}$ and $\vec{\mathfrak{q}_i}$ are vectors with $2^{(k-1)}$ components, given that there are k queues in the model. For q_i, let $\mathfrak{B}_i = (\mathfrak{b}_1, \mathfrak{b}_2, ..., \mathfrak{b}_{i-1}, \mathfrak{b}_{i+1}, ..., \mathfrak{b}_k)$ be a binary vector with $(k-1)$ components. There are totally $2^{(k-1)}$ possible values for $\mathfrak{B}_i$. For a particular value of $\mathfrak{B}_i$, let

$$p_i(\mathfrak{B}_i) = p_i \prod_{l \neq i}^{k} [p_l^{(\mathfrak{b}_l)} \bar{p}_l^{(1-\mathfrak{b}_l)}],$$

where $\bar{p}_l = (1 - p_l)$. Now let $\vec{\mathfrak{p}_i}$ be the vector formed by $p_i(\mathfrak{B}_i)$ for all $2^{(k-1)}$ different values of $\mathfrak{B}_i$. Similarly, let $q_{i|\mathfrak{B}_i}$ represent the marginal successful transmission probability of q_i such that $q_l \in \mathcal{S}$ if and only if $\mathfrak{b}_l = 1$, where $l \neq i$. Again, let $\vec{\mathfrak{q}_i}$ be the vector formed by $q_{i|\mathfrak{B}_i}$ for all $2^{(k-1)}$ different values of $\mathfrak{B}_i$. The ordering of the vectors $\vec{\mathfrak{p}_i}$ and $\vec{\mathfrak{q}_i}$ match to each other in the sense that the lth components of both vectors have the same $\mathfrak{B}_i$ value.

Now applying the same approach in the last three subsections, we have relative stability for the model in the following theorem and corollary.

Theorem 5.2.4 *In the MPR model, for a given configuration of the system,* $q_i \succeq q_j \iff \frac{k_i}{\vec{\mathfrak{p}_i} \cdot \vec{\mathfrak{q}_i}} \leq \frac{k_j}{\vec{\mathfrak{p}_j} \cdot \vec{\mathfrak{q}_j}}$ *on the linear increasing path* L_K*, where* $\vec{\mathfrak{p}_i} \cdot \vec{\mathfrak{q}_i}$ *is the dot product of the two vectors.*

Proof: We consider again the linear increasing path K_A on which all the queues are as stable as one another. For the MPR protocol, from Theorem 4.4.6, we know on this path

$$\mu_i^A = \vec{\mathfrak{p}_i} \cdot \vec{\mathfrak{q}_i},$$

and

$$\mu_j^A = \vec{\mathfrak{p}_j} \cdot \vec{\mathfrak{q}_j}.$$

Because we have $q_i \asymp q_j$ on L_A, therefore $k_i/\mu_i^A = k_j/\mu_j^A$ and it is easy to see that $C_{i,j} = \mu_i^A/\mu_j^A = \vec{\mathfrak{p}_i} \cdot \vec{\mathfrak{q}_i} / \vec{\mathfrak{p}_j} \cdot \vec{\mathfrak{q}_j}$. Then, according to Theorem 4.4.7, we have $q_i \succeq q_j \iff \frac{k_i}{\vec{\mathfrak{p}_i} \cdot \vec{\mathfrak{q}_i}} \leq \frac{k_j}{\vec{\mathfrak{p}_j} \cdot \vec{\mathfrak{q}_j}}$ on any path L_K. This finishes the proof. □

The stability ordering in the MPR model is given by the following corollary.

Corollary 5.2.4 *In the MPR model, for a given configuration, the stability ordering of the queues* $q_{(1)} \succeq q_{(2)} \succeq ... \succeq q_{(k)} \iff \frac{k_{(1)}}{\vec{\mathfrak{p}_{(1)}} \cdot \vec{\mathfrak{q}_{(1)}}} \leq \frac{k_{(2)}}{\vec{\mathfrak{p}_{(2)}} \cdot \vec{\mathfrak{q}_{(2)}}} \leq ... \leq \frac{k_{(k)}}{\vec{\mathfrak{p}_{(k)}} \cdot \vec{\mathfrak{q}_{(k)}}}$ *on the linear increasing path* L_K*, where* $(1), (2), ...(k)$ *is a permutation of the sequence* $1, 2, ..., k$.

Proof: The conclusion is a direct result of Theorem 5.2.4. □

In this section, we have established the relative stability conditions for the four SSMQSs. As we have shown, with the relative stability related properties of Type-1 SSMQSs, the procedure to find the relative stability condition for a particular system is straightforward and simple. The results for the polling system and the ALOHA network we have obtained in this section are consistent with Theorems 3.2.1 and 3.3.1. However, in Theorems 3.2.1 and 3.3.1 we are only able to consider the relative stability relation of two queues in the sense that one queue's stability implies another queue's stability on a single traffic point. If two queues have the same stability status on a point, e.g., both stable or unstable, then it is not very meaningful to discuss which one is more stable or unstable. On the other hand, if a linear increasing path of the traffic pattern is given, we can then compare their stability in the sense that to tell which queue is more stable or less stable, and to have the stability ordering on the path. Hence, in a more general sense, the relative stability reflects the trend of the stability of the queues. The results of the polling system and the ALOHA networks we have obtained in this section are also consistent with those that have previously reported.

5.3 Absolute Stability Conditions of Type-1 SSMQSs

In this section we demonstrate how to use the relative stability results to establish absolute stability conditions for Type-1 SSMQSs. As a system or a queue's stability conditions are often equivalent to the region of arrival patterns within which the system or the queue are stable, we use the term stability condition and stability region interchangeably in this section. Now we describe our approach first. Simply speaking, the idea to obtain stability condition of a system is to establish queue stability condition in the system first. Then there are two ways to obtain system stability condition through the queue stability condition. The first way is to consider the intersection of all the queues' queue stability regions, while the second way is to consider the union of every individual queue's queue stability regions within which the queue is the least stable queue. We illustrate the two methods through a Type-1 SSMQSs with two queues in Figs. 5.1 and 5.2. In Fig. 5.1 (a) and (b) the light shadow areas represent the queue stability regions of q_1 and q_2, respectively. The system stability region (the dark shadow area) in Fig. 5.1 (c) is then the intersection of the two queues' queue stability regions. In Fig. 5.2 (a) and (b) the shadow areas represent the queue stability regions of q_1 and q_2 when they are the least stable queues, respectively. Then the system stability shown in Fig. 5.2 (c) is the union of these two shadow areas.

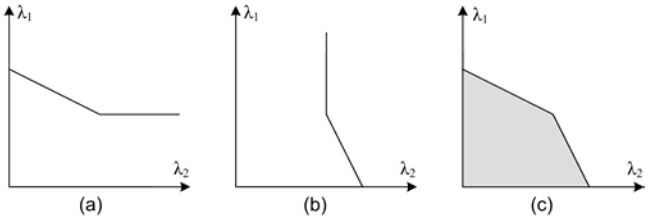

Fig. 5.1. The first method to achieve system stability.

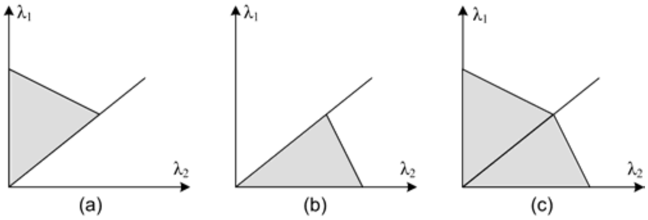

Fig. 5.2. The second method to achieve system stability.

To obtain queue stability condition, we consider the queues on a given linear increasing path. The stability ordering of the queues on the path can be know through the relative stability results. For the target queue, we can divide all the queues into three groups, namely, the queues that are less stable than, as stable as, and more stable than the target queue. Then consider the target queue in the state as if it just passes its queue stability boundary. Consequently, the queues that are as stable as and less stable than the target queue will be unstable. Now all the queues are partitioned into two sets, i.e., the stable ones and the unstable ones. Based on our definition of Type-1 SSMQSs, we still can construct a stationary and ergodic regime of the system state. This implies we can apply the second part (the instability part) of Loynes' theorem's to the target queue and obtain its queue instability boundary. Because Loynes' theorem is sufficient and necessary, that boundary is also the target queue's stability boundary. At this point, theoretically, we need to calculate the maximum service rate of the target queue in order to apply Loynes' theorem. Note that in practice, however, for some systems, the maximum service rate may not be analytically computable due to the nonlinear feature of the systems. The ALOHA network and the MPR model belong to this kind of systems. Nevertheless, once the maximum service rate can be computed, the queue stability condition of the target queue on the given path can be achieved. Then repeat the procedure for every possible linear increasing path, we achieve the stability condition of the target queue for the traffic space.

A necessary stability condition for the target queue through the reconfiguration method is discussed in the last chapter. That is, assume the target queue with fixed associated system parameter on a given linear increasing path is the least stable queue, then a necessary condition of the queue is the solution of the maximum as stable as configuration of the system in terms of the target queue's arrival rate.

5.3.1 The Polling System with Gated Limited Service

We consider first the queue stability condition of a target queue q_t on a linear increasing path L_K. As described in the outline of the approach, on L_K, all the queues can be categorized into 3 groups with respect to q_t. Let $\mathcal{M}_t$ be the set of queues that are more stable than q_t, $\mathcal{L}_t$ be the set of queues that are less stable than q_t, and $\mathcal{A}_t$ be the set of queues that are as stable as q_t. Now assume we can push the traffic pattern along the path and let it just pass q_t's queue stability boundary. Then q_t, the queues in $\mathcal{L}_t$ and $\mathcal{A}_t$ will be unstable. Because there exists a stationary and ergodic regime of the system, the mean cycle time of the server can be given as following,

$$\mathbf{E}C = \sum_{q_l \in \mathcal{M}_t} (X_l b_l) + \sum_{q_l \in (\{q_t\} \cup \mathcal{A}_t \cup \mathcal{L}_t)} (M_l b_l) + u_0, \tag{5.7}$$

where $X_l \leq M_l$ is the average number of customers served at $q_l \in \mathcal{M}_t$. In Eq. (5.7), the first term corresponds to the time incurred by the set of stable queues whereas the second term correspond to the time incurred by the set of unstable queues, and the last term is the total switch-over time of the server. According to Lemma 3.2.3, we have $X_l = \lambda_l \mathbf{E}C$ for $q_l \in \mathcal{M}_t$, which implies

$$\mathbf{E}C = \frac{\sum_{q_l \in (\{q_t\} \cup \mathcal{A}_t) \cup \mathcal{L}_t)} M_l b_l + u_0}{1 - \sum_{q_l \in \mathcal{M}_t} \rho_l}. \tag{5.8}$$

On L_K, the value of $\hat{\mu}_t^K$ thus is equal to $\frac{M_t}{\mathbf{E}C}$. By applying Loynes' theorem, q_t is unstable on L_K if and only if the $\lambda_t > \frac{M_t}{\mathbf{E}C}$. Therefore, the stability boundary of q_t is then

$$\lambda_t < \frac{M_t}{\mathbf{E}C}.$$

From the above discussion for a given L_K, we can now consider the set of paths on which the queues are partitioned into the same sets as $\mathcal{M}_t, \mathcal{L}_t$, and $\mathcal{A}_t$. This can be done because of Theorem 5.2.1 and Corollary 5.2.1. Let $\Gamma_o \equiv (\mathcal{M}_{t,o}, \mathcal{L}_{t,o}, \mathcal{A}_{t,o})$ be a particular partition of all the queues given that q_t is the target queue, we denote the set of paths that can partition the queues into Γ_o by $L(\Gamma_o)$. Therefore, we have the following queue stability condition of q_t with respect to the set $L(\Gamma_o)$.

Lemma 5.3.1 *The target queue q_t is stable on $L_K \in L(\Gamma_o)$ if*

$$\lambda_t < \frac{M_t}{\mathbf{E}C}, \tag{5.9}$$

where $\Gamma_o \equiv (\mathcal{M}_{t,o}, \mathcal{L}_{t,o}, \mathcal{A}_{t,o})$, and

$$\mathbf{E}C = \frac{\sum_{q_l \in (\{q_t\} \cup \mathcal{A}_{t,o}) \cup \mathcal{L}_{t,o})} M_l b_l + u_0}{1 - \sum_{q_l \in \mathcal{M}_{t,o}} \rho_l}. \tag{5.10}$$

Moreover, q_t is unstable if $\lambda_t > \frac{M_t}{EC}$.

Proof: From the discussion of q_t's stability on a particular linear increasing path, and noting the set of queues in the partition Γ_o do not change, the term $\mathbf{E}C$ is well defined and is given by Eq. (5.10). The target queue's maximum service rate is therefore given by $\hat{\mu}_t^K = \frac{M_t}{\mathbf{E}C}$ on any $L_K \in P(\Gamma_o)$. By Loynes' theorem, the queue is unstable if $\lambda_t > \frac{M_t}{\mathbf{E}C}$, and stable if $\lambda_t < \frac{M_t}{\mathbf{E}C}$. This finishes the proof. □

When consider all possible partitions of the queues with respect to q_t, we can then have the queue stability condition for q_t in the whole traffic space.

Theorem 5.3.1 *The stability region of q_t in the whole traffic space is given by $\cup_{\Gamma_o} R(\Gamma_o)$, where $R(\Gamma_o)$ is q_t's stability region for the set of paths $L(\Gamma_o)$.*

Proof: The proof is straightforward because the stability region of q_t's is simply the union of $R(\Gamma_o)$ for all possible Γ_o with respect to q_t, and $R(\Gamma_o)$ can be obtained by Lemma 5.3.1. □

As we can see, the essential part in the above results is that we are able to identity paths on which the queues are partitioned identically as a given Γ_o with respect to q_t. And this can be achieved only when we have the relative stability conditions of the system.

Next we show that the q_t's stability condition when it is the least stable queue on a path L_K can also be obtained through the reconfiguration method. Let $\mathcal{G}$ be the original configuration and Λ be the stability boundary of q_t on L_k. Now reconfigure the system by changing the parameters of $q_i \neq q_t$ from M_i to M'_i. Let the new configuration be $\mathcal{G}_m$. To make all the queues be as stable as one another, on path L_k, $\mathcal{G}_1$ should satisfy the following set of equations

$$\begin{cases} \frac{k_t}{M_t} = \frac{k_1}{M'_1}, \\ \frac{k_t}{M_t} = \frac{k_2}{M'_2}, \\ \cdots\cdots\cdots \\ \frac{k_t}{M_t} = \frac{k_k}{M'_k}. \end{cases}$$

Then for each $q_i \neq q_t$, we have $M'_i = k_i M_t / k_t$. For convenience, also denote $M'_t = M_t$. For the configuration $\mathcal{G}_m$, since all the queues are as stable as one another, when the traffic point just passes q_t's boundary, the mean cycle time will be

$$\mathbf{E}C = \sum_{j=1}^{k} M'_j b_j + u_0.$$

The stability condition of q_t under $\mathcal{G}_m$ through Loynes' theorem will be

$$\lambda_t < M'_t / \mathbf{E}C.$$

Substitute all the $M'_i = k_i M_t / k_t$ to the above and note that $\lambda_i = k\lambda$, where λ is the free parameter, we can have the necessary queue stability condition for q_t as the same as in Eq. (5.9). It is easy to see that once we have the solution of the boundary of λ_t, the solution of M'_is can be obtained and they are unique for the given M_t and L_k. This implies the necessary queue stability condition of q_t obtained is also sufficient, as on L_K and the given M_t, there is only one as stable as configuration of all the queues.

Now we show how to derive the system stability conditions from the queue stability conditions. The first approach is to consider the set of paths on which q_i is the least stable queue. Denoted such set of paths as L^l_i. If we can obtain the corresponding queue stability region for q_i on the paths in L^l_i, denoted by R^l_i, then the system will also be stable within the region of R^l_i, since the system stability boundary point on a path is the same as the least stable queue's boundary point. Therefore, the system stability region can be expressed as a union of $R_i^l, i = 1, \ldots, k$, and we have the following system stability result.

Theorem 5.3.2 *The polling system is stable in the region* $\cup_{q_i \in \mathcal{K}} R_i^l$*, where* R_i^l *is given by*

$$\cup_{L_K \in L_i^l} \{\lambda_i < \frac{M_i}{\mathbf{E}C}\} \text{ and } \mathbf{E}C = \frac{u_0}{1 - \sum_{i=1,\ldots,k} \rho_i}. \tag{5.11}$$

Proof: According to Lemma 5.3.1, q_i is stable on a path in L_i^l if $\lambda_i < \frac{M_i}{\mathbf{E}C}$, where $\mathbf{E}C = \frac{u_0}{1 - \sum_{i=1,\ldots,k} \rho_i}$. Thus, we can obtain R_i^l as in Eq. (5.11). Since each queue is the least stable queue in some nonempty partition, the entire system stability region is the union of R_i^l for all q_i. □

Instead of performing a set union operation as in the last method, another method is based on a set intersection method. That is, taking an intersection of all queues' stability regions will yield the system stability region, because only the regions correspond to the least stable queues will remain after the intersection operation. For the purpose of illustration, we consider a polling system with two queues. According to Theorem 5.3.1, q_1 is stable in the region $R_1^1 \cup R_2^1$, where

$$R_1^1 = \begin{cases} \frac{\lambda_1}{M_1} \geq \frac{\lambda_2}{M_2} \\ \\ \lambda_1 < \frac{M_1 - M_1 \lambda_2 b_2}{M_1 b_1 + u_0} \end{cases}$$

and

$$R_2^1 = \begin{cases} \frac{\lambda_1}{M_1} < \frac{\lambda_2}{M_2} \\ \\ \lambda_1 < \frac{M_1}{M_1 b_1 + M_2 b_2 + u_0}. \end{cases}$$

Similarly, q_2's stability region is given by $R_1^2 \cup R_2^2$, where

$$R_1^2 = \begin{cases} \frac{\lambda_1}{M_1} \geq \frac{\lambda_2}{M_2} \\ \\ \lambda_2 < \frac{M_2}{M_1 b_1 + M_2 b_2 + u_0} \end{cases}$$

and

$$R_2^2 = \begin{cases} \frac{\lambda_1}{M_1} < \frac{\lambda_2}{M_2} \\ \\ \lambda_2 < \frac{M_2 - M_2\lambda_1 b_1}{M_2 b_2 + u_0}. \end{cases}$$

Then the intersection of $R_1^1 \cup R_2^1$ and $R_1^2 \cup R_2^2$ is given by $R_1 \cup R_2$, where

$$R_1 = \begin{cases} \frac{\lambda_1}{M_1} \geq \frac{\lambda_2}{M_2} \\ \\ \lambda_1 < \frac{M_1 - M_1\lambda_2 b_2}{M_1 b_1 + u_0} \end{cases}$$

and

$$R_2 = \begin{cases} \frac{\lambda_1}{M_1} < \frac{\lambda_2}{M_2} \\ \\ \lambda_2 < \frac{M_2 - M_2\lambda_1 b_1}{M_2 b_2 + u_0}. \end{cases}$$

One can easily verify that $R_1 \cup R_2$ is identical to the system stability region obtained through Theorem 5.3.2.

5.3.2 The Slotted Buffered ALOHA Network

It is well known that the exact queue and system stability conditions for the ALOHA network is only available when there are two queues in the system. If there are more than two queues in the ALOHA network, the analytical results of the absolute stability conditions are still open problems. In this subsection, instead of reproducing those published results [17, 47, 57], we derive a new necessary system stability condition for the ALOHA network through the method of finding the maximum as stable as configuration described in Chapter 4. We then compare our necessary stability condition with the ones obtained in [47]. Without loss of generality, we assume q_1 is the target queue. As discussed in Chapter 4, a necessary condition of q_1 can be obtained by finding the stability condition of q_1 for the maximum as stable as configuration in terms of λ_1 on a path. For the given p_1 and $\lambda_2, \lambda_3, ..., \lambda_k$, such a path satisfies the following according to Theorem 5.2.2 and Corollary 5.2.2:

$$\frac{\lambda_1(1-p_1)}{p_1} = \frac{\lambda_2(1-p_2')}{p_2'} = .. = \frac{\lambda_k(1-p_k')}{p_k'},$$

where p_i's are the transmission probability of $q_i \neq q_1$ for the maximum as stable as configuration. Rewrite the above into the following

$$\begin{cases} \frac{\lambda_1(1-p_1)}{p_1} = \frac{\lambda_2(1-p_2')}{p_2'}, \\ \frac{\lambda_1(1-p_1)}{p_1} = \frac{\lambda_3(1-p_3')}{p_3'}, \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \frac{\lambda_1(1-p_1)}{p_1} = \frac{\lambda_k(1-p_k')}{p_k'}. \end{cases}$$

Then we have a set of $k-1$ equations with $k-1$ unknowns. For each $q_i \neq q_1$, we have $p_i' = \frac{\lambda_i P_1}{\lambda_1 + \lambda_i P_1}$, where $P_1 = p_1/(1-p_1)$. On this path, with the new p_i's, all the queues will be as stable as one another, therefore, the stability boundary of q_1 will be

$$\lambda_1 < p_1 \prod_{i=2}^{k} (1 - p_i').$$

Substitute $p_i' = \frac{\lambda_i P_1}{\lambda_1 + \lambda_i P_1}$ into the above, we reach at

$$\prod_{i=2}^{k} (\lambda_1 + P_1 \lambda_i) < p_1 \lambda_1^{(k-2)}.$$

Then the maximum λ_1 that satisfies the above inequality is the outer bound of λ_1 for the given p_1 and $\lambda_2, \lambda_3, ..., \lambda_k$. Moreover, any λ_1' satisfies the above inequality will lead to a path (or, in other words, provide a set of p_i's) such that all the queues are as stable as one another on the path and have the boundary at $(\lambda_1, \lambda_2, ..., \lambda_k)$. Among them, the minimum λ_1' will lead to the path on which q_1 is the most stable queue in the original configurations, i.e., the original p_is, while the maximum λ_1' will lead to the path on which q_1 is the least stable queue in the original configurations. Therefore, the maximum λ_1' satisfies the above inequality is the necessary system stability condition for the ALOHA network. As the minimum λ_1' can also serve as the sufficient system stability condition, it will be too loose when compare with the sufficient system stability conditions obtained in [47, 57].

In Tables 5.1-5.4 we compare our necessary condition with the one obtained in [47]. From the simulation results, one can find that our bound is tighter than [47]'s. The reason is that the necessary bound obtained in [47] can be considered as the solution of the maximum λ_1 for all the paths when considering q_1 is the least stable queue and has a fixed p_1. In our approach, we only consider those paths on which all the queues are as stable as one another. Note that on these paths, q_1 is also the least stable queue. This implies our necessary bound is tighter than [47]'s.

λ_2	λ_3	Simulation	Our bound	[47]'s bound
0.0	0.0	0.500	0.500	0.500
0.0	0.12	0.380	0.380	0.380
0.06	0.06	0.3646	0.3703	0.380
0.12	0.123	0.1508	0.1704	0.257
0.12	0.13	0.130	0.130	0.250

Table 5.1
Comparison of upper bounds of λ_1 for $k = 3$ and $p_1 = p_2 = p_3 = 0.5$

λ_2	λ_3	Simulation	Our bound	[47]'s bound
0.0	0.0	0.800	0.800	0.800
0.0	0.05	0.600	0.600	0.600
0.018	0.028	0.5817	0.6026	0.616
0.03	0.05	0.3282	0.4233	0.480
0.035	0.0561	0.1565	0.3444	0.4356
0.025	0.0563	0.3363	0.4214	0.4748

Table 5.2
Comparison of upper bounds of λ_1 for $k = 3$ and $p_1 = 0.8, p_2 = 0.7, p_3 = 0.6$

λ_2	λ_3	λ_4	λ_5	Simulation	Our bound	[47]'s bound
0.0	0.0	0.0	0.0	0.500	0.500	0.500
0.0	0.0	0.0	0.015	0.485	0.485	0.485
0.0	0.0	0.015	0.015	0.4693	0.4695	0.470
0.0	0.015	0.015	0.015	0.4525	0.4535	0.455
0.015	0.015	0.015	0.015	0.4337	0.4368	0.440
0.03	0.03	0.03	0.03	0.3357	0.3643	0.380
0.03	0.03	0.03	0.033	0.3274	0.3604	0.377
0.033	0.032	0.031	0.03	0.3158	0.3563	0.374
0.0325	0.032	0.0315	0.03	0.3138	0.3563	0.374

Table 5.3
Comparison of upper bounds of λ_1 for $k = 5$ and $p_i = 0.5$ for all $i = 1..5$

$\lambda_2, \lambda_3, ..., \lambda_{10}$	Simulation	Our bound	[47]'s bound
$\lambda_2 = ... = \lambda_{10} = 0.0$	0.10	0.10	0.10
$\lambda_2 = 0.0$, $\lambda_3 = ... = \lambda_{10} = 0.019$	0.0812	0.0815	0.083
$\lambda_2 = ... = \lambda_{10} = 0.019$	0.07881	0.07883	0.081
$\lambda_2 = ... = \lambda_{10} = 0.036$	0.0494	0.0501	0.064
$\lambda_2 = 0.039$, $\lambda_3 = ... = \lambda_{10} = 0.036$	0.0485	0.0491	0.0636
$\lambda_2 = ... = \lambda_5 = 0.039$, $\lambda_6 = ... = \lambda_{10} = 0.036$	0.0453	0.0459	0.0627

Table 5.4
Comparison of upper bounds of λ_1 for $k = 10$ and $p_i = 0.1$ for all $i = 1..10$.

5.3.3 NDMA and BNDMA

For stability conditions of the NDMA and BNDMA, again, we start with queue stability conditions. In the NDMA protocol, for a target queue q_t and a given L_K, all the queues can be grouped into three sets with respect to q_t according to the relative stability relation on the path, namely, $\mathcal{M}_t$, $\mathcal{L}_t$, $\mathcal{A}_t$. Let l^n be the length of the nth collision resolution cycle. Because the protocol is a Type-1 SSMQS, l^n has a unique and finite expectation even when some of the queues are unstable. Denote the mean as $\mathbf{E}l$. Then when the system traffic increases along L_K and just passes q_t's stability boundary, q_t as well as the queues in the sets of $\mathcal{L}_t$ and $\mathcal{A}_t$ will never empty. We have

$$\mathbf{E}l = \sum_{q_i \in \mathcal{M}_t} \lambda_i \mathbf{E}l + |\{q_t\} \cup \mathcal{L}_t \cup \mathcal{A}_t|$$

The first item in the right hand side represents the average number of packets that can be served from queues in the $\mathcal{M}_t$ set during $\mathbf{E}l$. Because all the queues in $\mathcal{M}_t$ are stable, based on the balance argument that the average number of arrivals during a period is equal to the average number of departures, we have the average departures from the queues as $\sum_{q_i \in \mathcal{M}_t} \lambda_i \mathbf{E}l$. The second item in the right hand side represents the number of packets that can be served from queues in the set of $\{q_t\} \cup \mathcal{L}_t \cup \mathcal{A}_t$. Because all these queues will never empty, the value of $|\{q_t\} \cup \mathcal{L}_t \cup \mathcal{A}_t|$ is a constant and also equals to $1 + |\mathcal{L}_t \cup \mathcal{A}_t|$. Solve El we have

$$\mathbf{E}l = \frac{1 + |\mathcal{L}_t \cup \mathcal{A}_t|}{1 - \sum_{q_i \in \mathcal{M}_t} \lambda_i}.$$

Therefore, the queue stability of q_t on L_k will be

$$\lambda_t < \frac{1}{\mathbf{E}l} = \frac{1 - \sum_{q_i \in \mathcal{M}_t} \lambda_i}{1 + |\mathcal{L}_t \cup \mathcal{A}_t|}. \tag{5.12}$$

Let $\Gamma_o \equiv (\mathcal{M}_{t,o}, \mathcal{L}_{t,o}, \mathcal{A}_{t,o})$ be a particular partition of all the queues given that q_t is the target queue, we denote the set of paths that can partition the queues into Γ_o by $L(\Gamma_o)$. Then, we have the following queue stability condition of q_t with respect to the set $L(\Gamma_o)$.

Lemma 5.3.2 *In NDMA protocol, the target queue q_t is stable on $L_K \in L(\Gamma_o)$ if*

$$\lambda_t < \frac{1}{\mathbf{E}l}, \tag{5.13}$$

where $\Gamma_o \equiv (\mathcal{M}_{t,o}, \mathcal{L}_{t,o}, \mathcal{A}_{t,o})$, and

$$\mathbf{E}l = \frac{1 + |\mathcal{L}_{t,o} \cup \mathcal{A}_{t,o}|}{1 - \sum_{q_i \in \mathcal{M}_{t,o}} \lambda_i}. \tag{5.14}$$

Moreover, q_t is unstable if $\lambda_t > \frac{1}{\mathbf{E}l}$. □

When consider all possible partitions of the queues with respect to q_t, we have the queue stability condition for q_t in the whole traffic space.

Theorem 5.3.3 *In NDMA protocol, the stability region of q_t in the whole traffic space is given by $\cup_{\Gamma_o} R(\Gamma_o)$, where $R(\Gamma_o)$ is q_t's stability region for the set of paths $L(\Gamma_o)$.* □

From Eq. 5.14 it is easy to see when q_t is the least stable queue on L_K, $\mathbf{E}l = 1/(1 - \sum_{i \neq t} \lambda_i)$. Hence, the system stability of NDMA on L_K is

$$\lambda_t < (1 - \sum_{i \neq t} \lambda_i) \Longleftrightarrow \sum_i \lambda_i < 1.$$

The above form will not change when consider every L_K and the least stable queue on L_K, therefore, the system stability condition of the NDMA protocol in the whole parameter space is also $\sum_i \lambda_i < 1$, and we state it as the following theorem.

Theorem 5.3.4 *The system stability condition of NDMA protocol is*

$$\sum_i \lambda_i < 1.$$

□

For the BNDMA protocol, as one more slot is needed in the collision resolution cycle, we have

$$\mathbf{E}l = \frac{2 + |\mathcal{L}_t \cup \mathcal{A}_t|}{1 - \sum_{q_i \in \mathcal{M}_t} \lambda_i}.$$

And the queue stability conditions of the BNDMA are as follows.

Lemma 5.3.3 *In BNDMA protocol, the target queue q_t is stable on $L_K \in L(\Gamma_o)$ if*

$$\lambda_t < \frac{1}{\mathbf{E}l}, \tag{5.15}$$

where $\Gamma_o \equiv (\mathcal{M}_{t,o}, \mathcal{L}_{t,o}, \mathcal{A}_{t,o})$, and

$$\mathbf{E}l = \frac{2 + |\mathcal{L}_{t,o} \cup \mathcal{A}_{t,o}|}{1 - \sum_{q_i \in \mathcal{M}_{t,o}} \lambda_i}. \tag{5.16}$$

Moreover, q_t is unstable if $\lambda_t > \frac{1}{\mathbf{E}l}$. □

Theorem 5.3.5 *In BNDMA protocol, the stability region of q_t in the whole traffic space is given by $\cup_{\Gamma_o} R(\Gamma_o)$, where $R(\Gamma_o)$ is q_t's stability region for the set of paths $L(\Gamma_o)$.* □

Finally, when q_t is the least stable queue on L_K, the system stability will be

$$\lambda_t < \frac{(1 - \sum_{i \neq t} \lambda_i)}{2} \Longleftrightarrow \sum_i \lambda_i + \lambda_t < 1.$$

If q_t is the least stable queue on a path, according to Theorem 5.2.3, λ_t will be the largest among all the λ_i. Therefore, the system stability condition of the BNDMA protocol can be given in the following.

Theorem 5.3.6 *The system stability condition of BNDMA protocol is*

$$\sum_i \lambda_i + \max_i \lambda_i < 1.$$

□

The system stability conditions for the NDMA and BNDMA protocols had been obtained in [23] with different approaches. As we have seen, the method we used in this section is more simple, especially for the BNDMA protocol. Moreover, the queue stability conditions of the protocols can also be easily obtained.

5.3.4 The MPR Model

For the MPR model, similar to the ALOHA network, the exact queue and system stability conditions for systems with more than two queues are still unknown. In this subsection, we demonstrate our approach by deriving the queue and system stability conditions for a two queue system. Moreover, we use the maximum as stable as configuration method to obtain a necessary stability condition for a k queue system. We start with queue stability conditions. Consider q_1 and q_2 in the system, according to Theorem 5.2.4, in the partition $(\lambda_1/(\vec{p_1}\cdot\vec{q_1}))\leq(\lambda_2/(\vec{p_2}\cdot\vec{q_2}))$, $q_1\succeq q_2$, i.e., q_2 is the least stable queue. Therefore, the queue stability condition of q_2 in this partition is

$$\begin{cases} \lambda_1/(\vec{p_1}\cdot\vec{q_1})\leq\lambda_2/(\vec{p_2}\cdot\vec{q_2}), \\ \lambda_2 < p_2 q_{\{2\}|\{2\}}(1-\frac{\lambda_1}{\vec{p_1}\cdot\vec{q_1}}) + (\vec{p_2}\cdot\vec{q_2})\frac{\lambda_1}{\vec{p_1}\cdot\vec{q_1}}. \end{cases} \tag{5.17}$$

While in the partition $(\lambda_1/(\vec{p_1}\cdot\vec{q_1}))\geq(\lambda_2/(\vec{p_2}\cdot\vec{q_2}))$, where q_2 is the most stable queue, q_2's queue stability condition is

$$\begin{cases} \lambda_1/(\vec{p_1}\cdot\vec{q_1})\geq\lambda_2/(\vec{p_2}\cdot\vec{q_2}), \\ \lambda_2 < \vec{p_2}\cdot\vec{q_2}. \end{cases} \tag{5.18}$$

Hence, the queue stability condition of q_2 in the traffic space is

$$\begin{cases} \lambda_2 < p_2 q_{\{2\}|\{2\}}(1-\frac{\lambda_1}{\vec{p_1}\cdot\vec{q_1}}) + (\vec{p_2}\cdot\vec{q_2})\frac{\lambda_1}{\vec{p_1}\cdot\vec{q_1}}, & \text{if } \lambda_1/(\vec{p_1}\cdot\vec{q_1})\leq\lambda_2/(\vec{p_2}\cdot\vec{q_2}), \\ \lambda_2 < \vec{p_2}\cdot\vec{q_2}, & \text{if } \lambda_1/(\vec{p_1}\cdot\vec{q_1})\geq\lambda_2/(\vec{p_2}\cdot\vec{q_2}). \end{cases} \tag{5.19}$$

Similar, we have the queue stability condition of q_1 as following.

$$\begin{cases} \lambda_1 < p_1 q_{\{1\}|\{1\}}(1-\frac{\lambda_2}{\vec{p_2}\cdot\vec{q_2}}) + (\vec{p_1}\cdot\vec{q_1})\frac{\lambda_2}{\vec{p_2}\cdot\vec{q_2}}, & \text{if } \lambda_1/(\vec{p_1}\cdot\vec{q_1})\geq\lambda_2/(\vec{p_2}\cdot\vec{q_2}), \\ \lambda_1 < \vec{p_1}\cdot\vec{q_1}, & \text{if } \lambda_1/(\vec{p_1}\cdot\vec{q_1})\leq\lambda_2/(\vec{p_2}\cdot\vec{q_2}). \end{cases} \tag{5.20}$$

Lastly, when intersect the region represented by Eqs. (5.19) and (5.20), we have the system stability for a two queue MPR model.

$$\begin{cases} \lambda_1 < p_1 q_{\{1\}|\{1\}}(1-\frac{\lambda_2}{\vec{p_2}\cdot\vec{q_2}}) + (\vec{p_1}\cdot\vec{q_1})\frac{\lambda_2}{\vec{p_2}\cdot\vec{q_2}}, & \text{if } \lambda_1/(\vec{p_1}\cdot\vec{q_1})\geq\lambda_2/(\vec{p_2}\cdot\vec{q_2}), \\ \lambda_2 < p_2 q_{\{2\}|\{2\}}(1-\frac{\lambda_1}{\vec{p_1}\cdot\vec{q_1}}) + (\vec{p_2}\cdot\vec{q_2})\frac{\lambda_1}{\vec{p_1}\cdot\vec{q_1}}, & \text{if } \lambda_1/(\vec{p_1}\cdot\vec{q_1})\leq\lambda_2/(\vec{p_2}\cdot\vec{q_2}), \end{cases} \tag{5.21}$$

One can easily verify Eq. (5.21) is the same as the one obtained in [53].

For a MPR model with k queues, by using the maximum as stable as configuration approach, we can have the necessary system stability condition (which is also the necessary queue stability condition when consider the queue is the least stable queue) in the following.

Theorem 5.3.7 *For a MPR model with given p_1 and $\vec{q_i}$ for each $i=1..k$, a necessary system stability condition (as well as a necessary queue stability condition of q_1 when* *it is the least stable queue) is the maximum λ_1 which satisfies the following set of inequalities.*

$$\begin{cases} \lambda_1/(\vec{\mathfrak{p}_1}'\cdot\vec{\mathfrak{q}_1})=\lambda_2/(\vec{\mathfrak{p}_2}'\cdot\vec{\mathfrak{q}_2}), \\ \lambda_1/(\vec{\mathfrak{p}_1}'\cdot\vec{\mathfrak{q}_1})=\lambda_3/(\vec{\mathfrak{p}_3}'\cdot\vec{\mathfrak{q}_3}), \\ \dots\dots\dots\dots\dots\dots \\ \lambda_1/(\vec{\mathfrak{p}_1}'\cdot\vec{\mathfrak{q}_1})=\lambda_k/(\vec{\mathfrak{p}_k}'\cdot\vec{\mathfrak{q}_k}), \\ \lambda_1 < \vec{\mathfrak{p}_1}'\cdot\vec{\mathfrak{q}_1}. \end{cases}$$

□

5.3.5 Stability Region Characterization

As discussed in Chapter 4, we can reformulate the overall system region characterization problem as an optimization problem of finding the maximum as stable as configuration of the system on a given path. As examples, we can immediately provide the overall system stability region of the ALOHA network in the following theorem.

Theorem 5.3.8 *In the slotted buffered ALOHA network, we have*

$$\mathfrak{O} = \{(\lambda_1, \lambda_2, ...\lambda_k)|\lambda_i < g_i \textit{ for all } i\},$$

where g_i is the solution of the following set of equations in terms of $p_i \in [0,1]$ and subject to $\max\{\sum_{i=1}^{k} g_i^2\}$

$$\begin{cases} g_1 = p_1 \prod_{i\neq 1}(1-p_i), \\ g_2 = p_2 \prod_{i\neq 2}(1-p_i), \\ \dots\dots\dots\dots\dots \\ g_k = p_k \prod_{i\neq k}(1-p_i). \end{cases}$$

Proof: As discussed in Chapter 4. □

Theorem 5.3.8 confirms the overall stability region of the ALOHA network obtained in [7] (through a special assumption of the arrival process) and [46] (through a sensitivity monotonicity of the ALOHA network).

For ALOHA network with two queues, from Theorem 5.3.8, the equations in the above becomes

$$\begin{cases} g_1 = p_1(1-p_2) \\ g_2 = p_2(1-p_1) \end{cases}$$

Since both p_1, p_2, $1-p_1$, and $1-p_2$ are positive. For g_1, we have $p_1(1-p_2) \leq \sqrt{p_1+(1-p_2)}/2$, and the equal sign holds when $p_1=(1-p_2)$ (or $p_1+p_2=1$). For g_2, we have $p_2(1-p_1) \leq \sqrt{p_2+(1-p_1)}/2$, and the equal sign holds $p_2=(1-p_1)$. This happens to be $p_1+p_2=1$ again, i.e., when $p_1+p_2=1$ both g_1 and g_2 reach their maximums. Therefore, the sum $g_1^2+g_2^2$ also reaches its maximum when $p_1+p_2=1$. Hence, the $\mathfrak{D}$ for the two queue ALOHA network is bounded by the curve $\sqrt{p_1}+\sqrt{p_2}=1$.

For the MPR model, we can have a similar result.

Theorem 5.3.9 *In the MPR model, we have*

$$\mathfrak{D} = \{(\lambda_1, \lambda_2, ...\lambda_k) | \lambda_i < g_i \textit{ for all } i\},$$

where g_i *is the solution of the following set of equations in terms of* $p_i \in [0,1]$ *and subject to* $\max\{\sum_{i=1}^{k} g_i^2\}$

$$\begin{cases} g_1 = \vec{\mathfrak{p}_1} \cdot \vec{\mathfrak{q}_1}, \\ g_2 = \vec{\mathfrak{p}_2} \cdot \vec{\mathfrak{q}_2}, \\ \ldots\ldots\ldots\ldots \\ g_k = \vec{\mathfrak{p}_k} \cdot \vec{\mathfrak{q}_k}. \end{cases}$$

□

5.4 Summary

In this chapter, based on the relative stability results established in last chapter, we derive both the relative and absolute stability conditions for some Type-1 SSMQSs. As we have shown, the approach used to derive those stability conditions is unified and simple. For the relative stability, the conditions can be solved completely. While for the absolute stability, both the queue and system stability conditions can be analyzed. Though for some systems such as the ALOHA and MPR model, the exact absolute stability conditions cannot be given, the approach still allows us to obtain necessary stability conditions directly. This can be considered as an evidence of one of the claims we have made in the Chapter 1: the relative stability indeed can help in the analysis of the absolute stability conditions.

To end this chapter, we briefly compare our approach of deriving stability conditions for Type-1 SSMQSs with the dominant system approach introduced in [57, 64]. Both approaches require Loynes' theorem to derive system stability conditions for the systems studied in this chapter. However, the dominant system approach is limited in the following aspects. First, in the dominant system approach, dominant systems are constructed for two purposes, namely, to satisfy the stationary and ergodic requirements of Loynes' theorem, and to eliminate the interaction among the queues so that a target queue can be isolated. To set up the dominant system, one essential assumption is the Poisson or Bernoulli arrival processes so that the joint queue length process can be represented as a Markov chain. Second, the dominant system alone in general cannot solve the queue stability problem except the stability ordering on a path is known, such as the case for the ALOHA [47]. Lastly, the dominant system approach cannot provide an effective way to find out the relative stability conditions of the queues in general.

At the contrast, in our approach, we constrain ourself only to concentrate on Type-1 SSMQSs. This constrain allows us to remove the difficulty of proving the stationary and ergodic requirements from the stability analysis of the system, though the difficulty of proving a target system is a Type-1 SSMQS still remains. Nevertheless, our approach to the system stability can accommodate more general arrival processes such as the one assumed for the processor sharing model in Chapter 3. Of course, to use our approach, one still needs to show that systems with such arrival processes have stationary regimes. Second, because our approach is based on the relative stability properties of Type-1 SSMQSs, it provides simple ways to solve the relative stability problems. This further allows us to derive not only system but also queue stability conditions of Type-1 SSMQSs in similar steps. In these senses, we consider our approach is more general.

6. CONCLUSION AND FUTURE RESEARCH

6.1 Conclusion

In this work we have studied the stability problems for the SSMQSs from a relative stability point of view. In the following, we first conclude the study, then we outline some possible extensions for the results established here.

Chapter 1: We briefly introduced the stability problems of SSMQSs. Especially, we described two different kinds of stability problems, namely, the absolute stability and the relative stability. For absolute stability we can have queue/system stability and the corresponding stability conditions, and degree of stability. For relative stability we can have relative stability relations and conditions. We then discussed the connections among these stability problems, in particular that the relative stability can help in achieving queue stability. This last point, together with the fact that there is a lack of studying relative stability of SSMQSs motivate us to have this study.

Chapter 2: We gave a condensed literature survey in this chapter. The survey reviewed different kinds of stability definitions in different settings, e.g., in dynamic systems, in stochastic systems, and in queueing systems. Then some commonly used methods for analyzing stability problems were discussed. Lastly, some existing stability results of SSMQSs were mentioned.

Chapter 3: In this chapter we studied the queues' relative stability relations in three SSMQSs, namely, a polling system with gated limit service, a slotted buffered ALOHA network, and a processor sharing system. Through examining these three systems, we observed that there are some properties that commonly shared by some SSMQSs. In particular, for all three systems, we found that some system state processes may have stationary regime even when some queues are unstable, i.e., the system is unstable. In addition, we found that any two queues in those three systems can have relative stability relations in the sense that the stability of one queue implies the stability of another queue, and this kind of relations can be reflected by the relations of the queues' arrival rates. Desirably, the explicit queue or system stability conditions are not required to derive the queues' relative stability relations. These observations provide us clues to study SSMQSs' relative stability more generally later. In the study of the relative stability relations of the three systems, we applied two different approaches, namely, the non-Markovian approach for Poisson or Bernoulli arrivals, and Loynes' backward reconstruction method for stationary marked point arrival processes.

Chapter 4: Based on the observations in Chapter 3, we identified two classes of SSMQSs based on the criterion whether there exists a stationary regime of some system state processes when the system is unstable. For the Type-1 SSMQSs we can define the concept of degree of stability as well as three relative stability relations among any two queues when comparing their degree of stability. That is, when the system traffic increases along a given path, a queue can be more stable than, less stable than, or as stable as another queue. Then we studied the properties related to the relative stability in the Type-1 SSMQSs. These properties allow us to obtain the relative stability conditions of Type-1 SSMQSs easily. In particular, one of the properties allows us to find the maximum as stable as configuration of the system on a given path. Then the characterization problem of the system stability region, the stabilization problem of the system, and finding the maximum stable throughput of the system can all be reformulated equivalent to finding such a configuration.

Chapter 5: In this chapter we provided a unified approach to the stability analysis of Type-1 SSMQSs. The approach is based on the relative stability results of the Type-1 SSMQSs. We used the approach to obtain both relative and absolute stability conditions for four Type-1 SSMQSs. The approach allows us to reproduce most of the previous results regarding to the stability of the systems and also obtain some new results. In particular, we derived a necessary stability condition of the ALOHA network which is better than the existing ones. We have also derived the relative stability condition for the MPR model.

6.2 Future Research

One possible future extension of this research will be finding the criteria for Type-1 SSMQSs. In this study we are only able to provide one sufficient condition to determine whether a given SSMQS is Type-1. For generally assumed models, a case by case analysis is still needed. Therefore, In order to make use of the relative stability results in general, such criteria are very desired.

Another extension is to further explore the problems of degree of stability. In this study, though we proposed a definition of the concept through which the relative stability of SSMQSs can be defined and studied, however, we have not touched on how to compute the degree of stability of a queue. In addition, the meaning of our definition of degree of stability can be considered as the distance between the current traffic point to the stability boundary on a given path. In other words, it highly depends on how the traffic patterns varies. This means that, without the path, the degree of stability will become not well-defined. Therefore, another kind of definition of the degree of stability may be needed such that it can tell us how stable a queue is only based on a given traffic point. We consider the empty probability of a queue is a good candidate for the purpose. Intuitively, the empty probability of a queue only depends on the current traffic input to the systems. We believe this kind of exploration can help us to know more about the concept of the degree of stability.

In this study we have seen the importance of the as stable as relation among the queues. In fact, in the study of multihop radio networks, a result states that the optimal service policy of the models tends to equalize the queue length differences among the queues [70]. As the connection between queue stability and queue length is obvious, we consider that our results about the as stable as relation also have a connection to the results in [70] regarding the optimal service policy. This is evidenced by the maximum as stable as configuration of the system, i.e., to configure the system such that all the queues are as stable as one another can achieve the maximum stable throughput. Therefore, to further investigate the optimality of the as stable as relation will be another possible extension.

Finally, through the relative stability results and approaches established in this study we can try to derive both absolute and relative stability conditions for more Type-1 SSMQSs. This can on one hand let us have the conditions for individual systems and on the other hand let us understand the Type-1 SSMQSs better in general.

Bibliography

[1] B. A and M. Brandt, "A note on the stability of the many-queue head-of-the-line processor-sharing system with permanent customers," *Queueing Systems*, vol. 32, pp. 363–381, 1999.

[2] B. A and M. Brandt, "On the stability of the multi-queue multi-server processor sharing with limited service," *Queueing Systems*, vol. 56, pp. 1–8, 2007.

[3] N. Abramson, "The aloha system—another alternative for computer communications," *AFIPS Conf. Proc.*, vol. 37, pp. 281–285, 1970.

[4] E. Altman and D. Kofman, "Bounds for performance measures of token rings," *IEEE/ACM Trans. Network.*, vol. 4, pp. 292–299, April 1996.

[5] E. Altman, P. Konstantopoulos, and Z. Liu, "Stability, monotonicity and invariant quantities in general polling systems," *Queueing Systems*, vol. 11, pp. 35–57, 1992.

[6] E. Altman and F. M. Spieksma, "Ergodicity, moment stability and central limit theorems of station times in polling systems," *Stochastic Models*, vol. 12, no. 2, pp. 307–328, 1996.

[7] V. Ananthatam, "The stability region of the finite-user slotted aloha protocol," *IEEE Trans. Inform. Theory*, vol. 37, no. 3, pp. 535–540, 1991.

[8] P. Belanger, *Control Engineering: Modern Approach.* HBJ College and School Division, 1997.

[9] D. Bertsekas and R. Gallager, *Data Network.* Prentice-Hall, 2nd ed., 1992.

[10] C. Bisdikian, L. Merakos, and L. Georgiadis, “Stability analysis of interconnected single-hop random-access networks,” *IEEE Trans. Commun.*, vol. 40, pp. 556–567, March 1992.

[11] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. P. Williamson, “Adversarial queueing theory,” *Journal of the ACM*, vol. 48, pp. 13–38, Jan. 2001.

[12] A. A. Borovkov, *Ergodicity and Stability of Stochastic Processes.* John Wiley & Sons, 1998.

[13] A. Brandt, P. Franken, and B. Lisek, *Stationary Stochastic Models.* Willy, 1992.

[14] C. S. Chang, “Stability, queue length, and delay of deterministic and stochastic queueing networks,” *IEEE Trans. Auto. Control*, vol. 39, pp. 913–931, May 1994.

[15] K. C. Chang, “Stability conditions for a pipeline polling scheme in satellite communications,” *Queueing Systems*, vol. 14, pp. 339–348, 1993.

[16] K. C. Chang and S. Lam, “A novel approach to queue stability analysis of polling models,” *Performance Evaluation*, vol. 40, pp. 27–46, March 2000.

[17] K. C. Chang and S. Lam, “Per-queue stability analysis of a random access system,” *IEEE Trans. Automatic Control*, vol. 46, pp. 1466–1470, Sept 2001.

[18] H. Chen, “Fluid approximations and stability of multiclass queueing networks: work-conserving disciplines,” *The Annals of Applied Probability*, vol. 5, pp. 637–665, Aug 1995.

[19] K. L. Chung, *Markov Chains With Stationary Transition Probabilities.* Springer-Verlag, Berlin, 1967.

[20] R. L. Cruz, “A calculus for network delay, part i: network elements in isolation,” *IEEE Trans. Information Theory*, vol. 37, pp. 114–131, Jan 1991.

[21] J. G. Dai, “On positive harris recurrence of multiclass queueing networks: A unified approach via fluid models,” *Ann. Appl. Probab.*, vol. 5, pp. 49–77, 1995.

[22] J. G. Dai and S. P. Meyn, "Stability and convergence of moments for multiclass queueing networks via fluid models," *IEEE Trans. Automat. Control*, vol. 40, pp. 1899–1904, 1995.

[23] G. Dimić, N. D. Sidiropoulos, and L. Tassiulas, "Wireless networks with retransmission diversity access mechanisms: stable throughput and delay properties," *IEEE Trans. Signal Processing*, vol. 51, pp. 2019–2030, Aug. 2003.

[24] S. Foss and T. Konstantopoulos, "An overview of some stochastic stability methods," *Journal of the Operations Research Society of Japan*, vol. 47, no. 4, pp. 275–303, 2004.

[25] S. Foss and A. Kovalevskii, "A stability criterion via fluid limits and its application to a polling system," *Queueing Systems*, vol. 32, pp. 131–168, 1999.

[26] S. Foss and G. Last, "Stability of polling systems with exhaustive service policies and state-dependent routing," *Ann. Appl. Probab.*, vol. 6, no. 1, pp. 116–137, 1996.

[27] S. G. Foss and N. I. Chernova, "Dominance theorems and ergodic properties of polling systems," *Problems of Information Transmission*, vol. 32, no. 4, pp. 46–71, 1996.

[28] F. G. Foster, "On stochastic matrices associated with certain queueing processes," *Ann. Math. Statist*, vol. 24, pp. 355–360, 1953.

[29] C. Fricker and M. R. Jaïbi, "Monotonicity and stability of periodic polling models," *Queueing Systems*, vol. 15, pp. 211–238, 1994.

[30] C. Fricker and M. R. Jaïbi, "Stability of multi-server polling models," Tech. Rep. RR-3347, INRIA, 1998.

[31] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource Allocation and Cross-Layer Control in Wireless Networks.* now Publishers Inc., 2006.

[32] L. Georgiadis and W. Szpankowski, "Stability of token passing rings," *Queueing Systems*, vol. 11, pp. 7–33, 1992.

[33] L. Georgiadis and W. Szpankowski, "Stability analysis for yet another class of multidimensional distributed system," *Proceedings of the 11th International Conference on Analysis and Optimization System, Discrete Event Systems*, pp. 523–530, 1994.

[34] L. Georgiadis, W. Szpankowski, and L. Tassiuals, "Stability analysis of quota allocation access protocols in ring networks with spacial uses," *IEEE Trans. Information Theory*, vol. 43, pp. 923–937, 1997.

[35] D. Grillo, "Polling mechanism models in communication system - some application examples," *Stochastic Analysis of Computer and Communications Systems*, pp. 659–698, 1990. Elsevier Science/North-Holland.

[36] M. Hassan, H. Sirisena, and M. Atiquzzaman, "A congestion control mechanism for enterprise network traffic over asynchronous transfer mode networks," *Computer Communications*, vol. 22, pp. 1296–1306, 1999.

[37] C. V. Hollot, V. Misra, D. F. Towsley, and W. Gong, "A control theoretic analysis of RED," in *INFOCOM*, pp. 1510–1519, 2001.

[38] O. C. Ibe and X. Cheng, "Stability conditions for multiqueue systems with cyclic service," *IEEE Trans. Auto. Control*, vol. 33, pp. 102–103, Jan. 1988.

[39] D. Khotimsky and S. Krishnan, "Stability analysis of a parallel packet switch with bufferless input demultiplexors," in *Proc. IEEE ICC*, pp. 100–111, June 2001.

[40] M. L. Kotler, "Proof of stability conditions for token passing rings by lyapunov functions," *IEEE Trans. Auto. Control*, vol. 41, pp. 908–912, June 1996.

[41] P. J. Kuehn, "Multiqueue systems with non-exhaustive cyclic-service," *Bell System Tech. J.*, vol. 58, pp. 671–698, Mar. 1979.

[42] P. R. Kumar and S. P. Meyn, “Stability of queueing networks and scheduling policies,” *IEEE Trans. Auto. Control*, vol. 40, pp. 251–260, Feb. 1995.

[43] P. R. Kumar and S. P. Meyn, “Duality and linear programs for stability and performance analysis of queueing networks and scheduling policies,” *IEEE Trans. Auto. Control*, vol. 41, pp. 4–17, Jan. 1996.

[44] H. Levy, M. Sidi, and O. J. Boxma, “Dominance relations in polling systems,” *Queueing systems*, vol. 6, pp. 155–172, 1990.

[45] R. Loynes, “The stability of a queue with non-independnet inter-arrival and service times,” *Proc. Camb. Philos.*, vol. 58, pp. 497–520, 1962.

[46] J. Luo and A. Ephremides, “On the throughput, capacity, and stability regions of random multiple access,” *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2593–2607, 2006.

[47] W. Luo and A. Ephremides, “Stability of n interacting queues in random-access systems,” *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1579–1587, 1999.

[48] A. M. Lyapunov, *Stability of Motion.* Academic Press, 1966.

[49] L. Massoulie, “Stability of non-markovian polling systems,” *Queueing Systems*, vol. 21, pp. 67–95, 1995.

[50] G. Mergen and L. Tong, “Stability and capacity of regular wireless networks,” *IEEE Trans. Information Theory*, vol. 51, pp. 1938–1953, June 2005.

[51] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability.* Springer, 1993.

[52] E. Muhammad and S. Shaler, *Sample-Path Analysis of Queueing Systems.* Kluwer Academic Publishers, 1999.

[53] V. Naware, G. Mergen, and L. Tong, “Stability and delay of finite-user slotted aloha with multipacket reception,” *IEEE Trans. Information Theory*, vol. 51, pp. 2636–2656, July 2005.

[54] K. Ogata, *Discrete-Time Control Systems (2nd Edition).* Prentice Hall, 1994.

[55] K. Ogata, *Modern Control Engineering (4th Edition).* Prentice Hall, 2001.

[56] A. B. Pippard, *Response and Stability.* Cambridge University Press, 1985.

[57] R. Rao and A. Ephremides, “On the stability of interacting queues in a multiple-access system,” *IEEE Trans. Inform. Theory*, vol. 34, no. 5, pp. 918–930, 1988.

[58] V. Sharma, “Stability and continuity of polling systems,” *Queueing Systems*, vol. 16, pp. 115–137, 1994.

[59] B. Shrader and A. Ephremides, “Random access broadcast: stability and throughput analysis,” *IEEE Trans. Information Theory*, vol. 53, pp. 2915–2921, Aug. 2007.

[60] K. Sigman, *Stationary Marked Point Processes: An Intuitive Approach.* CHAPMAN & HALL, 1995.

[61] J. J. E. Slotine and W. Li, *Applied Nonlinear Control.* Prentice Hall, 1991.

[62] A. L. Stolyar, “On the stability of multiclass queueing networks: a relaxed sufficient condition via limiting fluid processes,” *Markov Processes and Related Fields*, vol. 1, no. 4, pp. 491–512, 1995.

[63] W. Szpankowski, “Stability conditions for multidimensional queueing systems with computer applications,” *Operations Research*, vol. 36, pp. 944–957, Nov. 1988.

[64] W. Szpankowski, “Towards computable stability criteria for some multidimensional stochastic processes,” *Stochastic Analysis of Computer and Communications Systems*, pp. 131–172, 1990. Elsevier Science/North-Holland.

[65] W. Szpankowski, "Stability conditions for some distributed systems: buffered random access systems," *Adv. Appl. Prob.*, vol. 26, pp. 498–515, 1994.

[66] H. Takagi, "Queueing analysis of polling models," *ACM Computing Surveys*, vol. 20, pp. 5–28, Mar. 1988. Elsevier Science/North-Holland.

[67] H. Takagi, "Queueing analysis of polling models: An update," *Stochastic Analysis of Computer and Communications Systems*, pp. 267–318, 1990. Elsevier Science/North-Holland.

[68] H. Takagi, *Queueing Analysis: A Foundation of Performance Evaluation. Vacation and Priority System, Part 1*, vol. 1. North-Holland, 1991.

[69] H. Takagi, "Analysis and application of polling models," *Performance Evaluation*, pp. 423–442, 2000. LNCS 1769.

[70] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[71] M. Tsatsanis, R. Zhang, and S. Banerjee, "Network-assisted diversity for random access wireless networks," *IEEE Trans. Signal Processing*, vol. 48, pp. 702–711, March 2000.

[72] B. Tsybakov and W. Mikhailov, "Ergodicity of slotted aloha system," *Probl. Pered. Inform.*, vol. 15, no. 4, pp. 73–87, 1979.

[73] R. L. Tweedie, "Criteria for ergodicity, exponential ergodicity and strong ergodicity of markov processes," *J. Appl. Probab.*, vol. 18, pp. 122–130, 1981.

[74] R. L. Tweedie, "Criteria for rates of convergence of markov chain with application to queueing theory," *Papers in Probability, Statistics and Analysis*, pp. 267–318, 1982. Cambridge Press, London.

[75] R. L. Tweedie, "The existence of moments for stationary markov chains," *J. Appl. Probab.*, vol. 20, pp. 191–196, 1983.

[76] V. M. Vishnevskii and O. V. Semenova, "Mathematical methods to study the polling systems," *Automation and Remote Control*, vol. 67, no. 2, pp. 173–220, 2006.

[77] O. Yaron and M. Sidi, "Performance and stability of communication networks via robust exponential bounds," *IEEE/ACM Trans. Network.*, vol. 1, pp. 372–385, June 1993.

[78] R. Zhang, N. D. Sidiropoulos, and M. Tsatsanis, "Collision resolution in packet radio networks using rotation invariance techniques," *IEEE Trans. Commun.*, vol. 50, pp. 146–155, Jan. 2002.