# Relative Stability Analysis of Multiple Queues

Lam Sum
Community College SPACE
The University of Hong Kong
Hong Kong, SAR China
sumlam@hkuspace.hku.hk

Rocky K. C. Chang and Yi Xie
Department of Computing
The Hong Kong Polytechnic University
Hong Kong, SAR China
{csrchang|csyxie}@comp.polyu.edu.hk

## ABSTRACT

In this paper we consider a general class of single-server multiqueue systems in which the stability of any single queue can be essentially determined by the queue's arrival rate and service rate. We refer such class of systems to as Rate Stability ($RS$) multiqueue systems. The $RS$-multiqueue system is general enough to admit different stability definitions and different models. We will present two sets of new results for the $RS$-multiqueue systems. These results extend many previous results on the stability analysis of multiqueue systems.

In the first part, we report that the $RS$-multiqueue systems can be classified into three classes. In each class, any pair of queues exhibits different interaction properties in three aspects: the number of intersection points of their stability boundaries, their possible relative stability relation, and whether a queue can have guaranteed service once becoming unstable.

In the second part, we present a relative stability analysis of two $RS$-multiqueue models: a polling model and a random access model. Moreover, the analysis facilities the absolute stability analysis of the models.

## Keywords

rate-stability multiqueue systems, degree of stability, relative stability, absolute stability, polling models, ALOHA system

## 1. INTRODUCTION

Contention for a single resource occurs in many computer and communication systems. It is common to use a single-server, multiqueue system to model these systems. The main focus of this paper is the stability of a multiqueue system. There are two main stability issues to consider in a multiqueue system: *queue stability* and *system stability.* The former concerns stability of individual queues and the latter the stability of all queues. We refer these two problems collectively to as *absolute stability problems*, because their goals is to find the ultimate stability conditions. In this paper, we take a different approach to the stability problems. Instead of obtaining the stability conditions directly, we first of all consider the *relative stability* properties which address the *interaction* among the queues based on their levels of stability. Unlike the complex analysis involved in a typical absolute stability analysis, a relative stability analysis provides us with a simple characterization for one queue to be more stable than the other or for two queues to be as stable as each other. Such characterization, to our surprise, applies to a large number of systems and even for different stability definitions. The relative stability results will also help yield the queue stability conditions without requiring dominant systems and monotonicity properties. The system stability conditions can also be immediately derived from the queue stability conditions.

The concept of relative stability is more common in control theory [1, 2, 3]. Associated with the concept of relative stability, the *degree of stability* of a control system can determine how large a perturbation is required to produce an unstable system. In the area of stability analysis of multiqueue systems, most of the attention has mainly been given to the absolute stability analysis [4, 5, 6, 7, 8, 9, 10, 11, 12]. Moreover, recently a few results of *stability rank* [13] or *stability ordering* [14, 15] have been reported. However, to the best of our knowledge, no general analysis on the relative stability issues has been reported. It is therefore our intention to provide a general framework to analyze the relative stability issues in this paper, and we use polling systems and buffered ALOHA systems to illustrate our approach.

This paper is organized as follows. In section 2 we describe the $RS$-multiqueue systems. Then we introduce the concepts of degree of stability and relative stability in section 3. By considering how many ways any two queues can be as stable as each other, in section 4 we introduce three classes of $RS$-multiqueue systems. We then provide the relative stability conditions of the queues for one class in section 5. Specifically, we derive conditions under which a queue is less stable than, or as stable as another queue. In section 6, we study the relative stability of two common computer communication network models, namely, a polling system with a limited service policy and the slotted buffered ALOHA system. The results are applied to obtain absolute stability conditions for the polling system in section 7. We finally conclude the paper in section 8.


*Valuetools* '06, October 11-13, 2006, Pisa, Italy


## 2. RATE-STABILITY MULTIQUEUE SYSTEMS

We consider in this paper a general class of multiqueue systems. Consider a multiqueue system consisting of a single server with a constant service rate and a set of queues. Generally, each queue has unlimited buffers. The arrival and service processes at the queues are assumed stationary and ergodic and are mutually independent. Moreover, the service policy at each queue is work-conserving and it does not need to be the same for all queues, e.g. a mixture of service policies. However, a nonzero overhead may be incurred when the server switches from one queue to another, thus making the multiqueue system generally nonwork-conserving.

Generally speaking, a queue is stable if the queue length is finite in some probabilistic sense, e.g., the queue length is bounded with probability one, the first moment exists and is finite, the distribution of the queue length process converge to a limiting distribution function, etc. In particular, we find that in the following two stability definitions, the stability conditions of a queue have something in common. In the first definition [16], a queue is said stable if the queue length process (equivalently, the waiting time process) tends to some honest distribution at all its points of continuity. The stability condition of a queue under such a definition is that if the average arrival rate is less than the average service rate, given that the system satisfies some stationary and ergodic assumptions. The second definition is for a queueing system with deterministic traffic model [17]. It says that a queue is stable if the queue length (or the delay of any packet) is always bounded. The stability condition of a queue under such a definition is again that if the average arrival rate is less than the service rate, given that the arrival process is an envelope process with an average rate and the service rate of the server is constant [18].

The common aspect of the above two stability conditions is that they are solely determined by the average arrival rate and average service rate of the queue, given that the arrival and the service processes satisfy some "stationary" (time invariant) and "ergodic" (average rate exists and is finite) requirements. One further observation is that in a single server queue with constant capacity, the average service rate is the service rate achieved by the server for a given arrival rate. It is not necessarily always equals to the capacity of the server. Intuitively, how fast the server can serve depends on how fast the customers arrive, and the server does not always serve at full speed. We term the service rate that can be achieved when the server works at its full capacity the *maximum achievable service rate* (MASR). In general, the average service rate is not larger than the MASR. The above discussion motivates us to define stability of a queue as the following.

DEFINITION 1. ***(Rate Stability)*** *Consider a single server queue for which the arrival process and the service process are stationary and ergodic. The queue is said to be stable if its average arrival rate is less than its maximum achievable service rate. The queue itself is called an $RS$-queue.*

If all the queues in a multiqueue system are $RS$-queues, then the system is called a $RS$-multiqueue system. If all the queues in the system are stable, then the system is said stable. From the above, we can see that the $RS$-multiqueue system can implicitly cover different systems and different kinds of stability definition. Noteworthy to mention that in a sample path approach toward stability analysis [19], the stability definition of a queue adopts a similar form, i.e., a queue is called rate stable if the average arrival rate and departure rate of the queue exist and are finite and equal.

We consider that the equilibrium *state* of a single queue is reflected by its service rate. The state of an $RS$-multiqueue system is then reflected by the service rate at each queue, i.e., the service rate vector. The service rate of the system is the sum of all the individual queues' service rates. When all the queues reach their MASRs, in general, the sum equals to the capacity of the server. For an always work-conserving server, this happens when the system traffic is on or beyond the system stability boundary. On the other hand, if a server is in general nonwork-conserving, the sum may be smaller than the server's capacity. Furthermore, the sum remains unchanged only when all queues are in their instability region. Once the system enters and remains in the instability region of all queues, it is easy to observe that the state of the system will not change. This implies that the states of the system in the instability region of all queues is determined by the entry point of that region. Another observation is that when there is only one queue in the system, then the queue has only one instability state. In general, when there are more than two queues in an $RS$-multiqueue system, it is possible for a queue to have different instability states. However, once a queue becomes unstable, it remains in a unstable state if other queue's arrival rate varies nondecreasingly.

# 3. DEGREE OF STABILITY

In an $RS$-multiqueue system, one general question to ask is how stable a queue is. In the following, we use a measure based on the arrival rate of an $RS$-queue to define the *degree of stability* for the queue that gives a normalized distance from the arrival rate to its stability boundary or a normalized distance between the service rate and the MASR. Thus, the smaller the normalized distance is, the less stable a queue is.

DEFINITION 2. ***(Degree of stability)*** *In an $RS$-queue $q$, the degree of stability of the queue is defined as*

$$D(q) \equiv 1 - \frac{\lambda}{C} = 1 - \hat{\rho}, \quad (1)$$

*where $\frac{\lambda}{C} = \hat{\rho}$ and $C$ is the queue's MASR.*

Note that $-\infty < D(q) \leq 1$. Since $C$ is the MASR, $D(q) > 0$ implies that the queue is stable and $D(q) < 0$ implies that the queue is unstable. Moreover, we do not further differentiate the cases of $D(q) < 0$ and $D(q) = -\infty$, which correspond to the cases of $C < \lambda$ and $C = 0$, respectively.

Once the degree of stability of a queue is defined, it is straightforward to use it as a measure for comparing the stability of any two queues. To do so, we define the *relative stability* for any two queues based on $D(q)$ in an $RS$-multiqueue system as follows.

DEFINITION 3. ***(Relative stability of two queues)*** *Consider $q_i$ and $q_j$ in an $RS$-multiqueue system at some equilibrium state.*

1. *$q_i$ is said less stable than $q_j$, denoted by $q_i \prec q_j$, iff $D(q_i) < D(q_j)$, or equivalently $q_i \prec q_j \Leftrightarrow \hat{\rho}_i > \hat{\rho}_j$.*

2. *$q_j$ is said more stable than $q_i$, denoted by $q_j \succ q_j$, iff $D(q_i) > D(q_j)$, or equivalently $q_i \succ q_j \Leftrightarrow \hat{\rho}_i < \hat{\rho}_j$.*

3. *$q_i$ is said as stable as $q_j$, denoted by $q_i \asymp q_j$, iff $D(q_i) = D(q_j)$, or equivalently $q_i \asymp q_j \Leftrightarrow \hat{\rho}_i = \hat{\rho}_j$.*

For convenience, we also say that $q_j$ is *at least as stable as* $q_i$ if either $q_i \asymp q_j$ or $q_i \prec q_j$, and we denote this relation by $q_i \preceq q_j$.

In the following, we study the preliminary condition under which the "as stable as" relation given in Definition 3 holds. To this end, let $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_n)$ be a *traffic point*, where $\lambda_i$ 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 $\Lambda$. We represent the increasing path in its parameterized form, i.e., each point on the path can be represented by $\Lambda = a \cdot K$, where $K$ is the *direction vector* of the line, i.e., $K = (k_1, k_2, ..., k_n)$. And $k_i \in \mathbb{R}^+$ is the *direction component* of $q_i$, and $a$ is a nonnegative real number such that $\lambda_i = a \cdot k_i, \forall i$ (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 increasing path as $L_K$ and a traffic point on the path as $\Lambda_K$. Note that each queue should have a unique boundary point on any increasing path, accordingly, the service rate of the queue at that point should be its MASR. We denote $q_i$'s MASR on $L_K$ by $C_i^K$.

Next we can determine the properties of $L_K$ for which the three relations of relative stability will hold. First, Definition 3 states that if $q_i \preceq q_j$ on $L_K$, then we have $\hat{\rho}_i \geq \hat{\rho}_j$ on the path. That is, the following holds at every traffic point on $L_K$.

$$\frac{\lambda_i}{C_i^K} \geq \frac{\lambda_j}{C_j^K} \Rightarrow \frac{a \cdot k_i}{C_i^K} \geq \frac{a \cdot k_j}{C_j^K} \Rightarrow \frac{k_i}{C_i^K} \geq \frac{k_j}{C_j^K}.$$

As a result, we have the following lemma.

Lemma 1. *In an RS-multiqueue system, if the traffic increases according to some direction vector $K$, $q_j$ is at least as stable as $q_i$ on the path $L_K$ if and only if the direction components of the two queues satisfy $\frac{k_i}{C_i^K} \geq \frac{k_j}{C_j^K}$ on $L_K$. That is, $q_i \preceq q_j \Leftrightarrow \frac{k_i}{C_i^K} \geq \frac{k_j}{C_j^K}$.*

# 4. THREE CLASSES OF RS-MULTIQUEUE SYSTEMS

Geometrically, the existence of an increasing path $L_K$ on which $q_i \asymp q_j$ implies that a traffic point on $L_K$ will be the stability boundaries for both queues. That is, the queue stability boundaries of $q_i$ and $q_j$ intersect each other on that particular traffic point. Two logical questions therefore follow: (1) does this $L_K$ exist? (2) if it exists, is it unique in some sense? The answers to the two questions in fact enable us to categorize the RS-multiqueue systems into three different classes.

In the first class, the path for which $q_i \asymp q_j$ does not exist, i.e., $q_i \succ q_j$ always hold for any $\lambda_i > 0$ and $\lambda_j > 0$. Consider a 2-queue system in which a server serves at most $M$ requests from $q_i$ and $q_j$ in each cycle. Furthermore, $q_i$ has a higher priority than $q_j$, and the server can serve all the $M$ requests from $q_i$ alone if it has generated enough requests. Thus, under this service policy, $q_j$ can be *starved* by $q_i$ on any linear increasing path $L_K$. In other words, $q_j$'s stability region is a proper subset of $q_i$'s and there are no intersection points for any two queues' stability boundaries. Consequently, once become unstable, $q_j$ will receive no service from the server and the two-queue system is degenerated into a single-queue system. It also implies that both $q_i$ and $q_j$ will have unique instability states. Another interesting point is that the above arguments are true regardless of which increasing path is. Specifically, consider an instability traffic point of both queues. Start from the origin and consider any increasing path (not necessarily to be linear) that reaches the instability traffic point, we can observe that (1) $q_j$ will always be less stable than $q_i$ or, equivalently, no intersection point of two queues' stability boundaries, (2) $q_j$ will receive no service from the server once it becomes unstable, and (3) both queues have unique instability state at that particular point. We term the property that the instability state of the system at a particular instability traffic point of all queues remains the same regardless of which path is adopted to reach the traffic point from the origin as *Path Independence of Instability Point*, or simply path independence. This relationship can obviously be extended to more than 2 queues. We therefore refer this class to as $RS_0$-multiqueue systems (the "0" refers to the nonexistence of as-stable-as paths) and summarize its properties in the following theorem.

Theorem 1. *An $RS_0$-multiqueue system exhibits the following properties,*

1. *one queue is always less stable than another;*
2. *a less stable queue receives no service from the server once it becomes unstable;*
3. *the system has an unique instability state of all queues;*
4. *an instability traffic point of the system is path independent.*

Proof. Consider any two queues in the system. Then the definition of $RS_0$ class is the geometrical meaning of (1), i.e., no intersection point of the two queues' stability boundaries, and therefore the only conclusion can be drawn is that one queue is always less stable than the other. Otherwise, it implies that there is at least one linear increasing path can cause the two queues to be as stable as each other. If (2) is false, it implies there is a "hole" at the less stable queue's stability boundary. More precisely, assume that the less stable queue receives a guaranteed nonzero service rate $\gamma > 0$ when it becomes unstable. Based on the $RS$-multiqueue system assumptions, as long as the arrival rate of the queue is less than $\gamma$, the queue should be stable. Then it is easy to find an increasing path from the origin such that the less stable queue's traffic component is always less than $\gamma$ but the more stable queue's traffic component grows indefinitely. Such an increasing path will cause the more stable queue to become unstable first, thus contradicting the assumption. Therefore, (2) is also true. Item (3) is also true because once entering into the instability region of all queues, all the less stable queues become unstable already and the system simply degenerates to a single-queue system. The most stable queue receives all the service. For that queue, it obviously has only one single instability state. Item (4) is a direct result from (3) simply by noting that the unique instability state of all queues applies to the whole instability region of all queues. Any path from the origin to the instability region of all queues will result the unique instability state of the system and therefore any instability traffic point of the system is path independent. □

In describing the second class, suppose that there are $n$ queues in the system. In this case, the as-stable-as paths exist, and they are *unique* in the sense that all such paths form a $(n-1)$-dimensional hyperplane. Equivalently, the intersections of the two queues' stability boundaries will be a $(n-1)$-dimensional hyperplane. All three systems to be considered in section VI are examples of this class. In Figures 1(a)-(b), we show the stability regions of the individual queues (the shaded regions) in a 2-queue system. Note that the whole traffic space is partitioned into two regions by a single as-stable-as path (1-dimensional hyperplane). The partitions correspond to $q_i \succ q_j$ and $q_i \prec q_j$. Moreover, each queue is *guaranteed* with a minimum service rate, denoted by $\lambda_{g,i}$ and $\lambda_{g,j}$. We refer this class to as $RS_1$-multiqueue system (the "1" refers to the uniqueness in the sense described above). The following theorem summarizes the properties of an $RS_1$-multiqueue system.

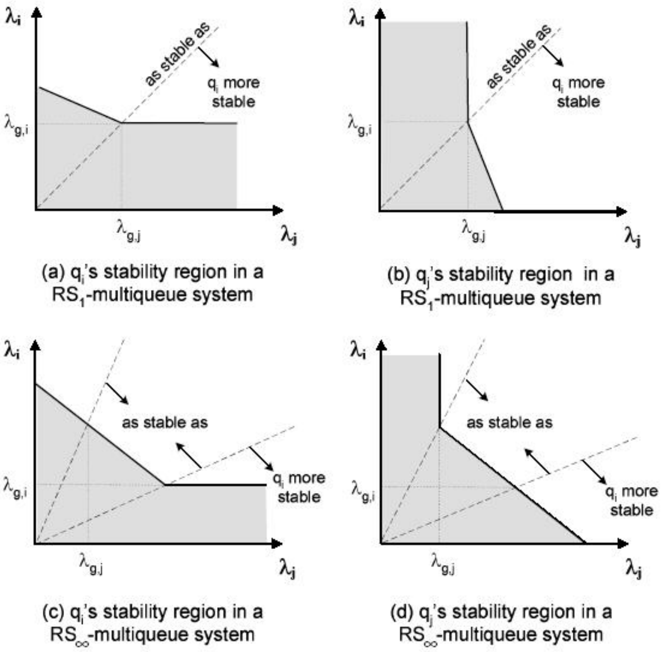



**Figure 1: Queue stability regions for an $RS_1$-multiqueue system of 2 queues.**

THEOREM 2. *An $RS_1$-multiqueue system of n queues exhibits the following properties,*

1. *for any two queues, there exists linear increasing paths such that the two queues exhibit all three relative stability relations, i.e., "less stable than", "as stable as", and "more stable than";*
2. *any single queue receives nonzero guaranteed service from the server even when it becomes unstable;*
3. *the system has an unique instability state of all queues;*
4. *an instability traffic point of the system is path independent.*

PROOF. Note that the definition of the $RS_1$-multiqueue system already guarantees that there exists increasing paths that can cause two queues as stable as each other. To prove the first item is then equivalent to prove that besides the "as stable as" case, it is impossible to have only "less stable than" (or "more stable than") between the two queues.

Without loss of generality, consider two queues $q_i$ and $q_j$ in the $RS_1$ system with $q_i \asymp q_j$ and $q_i \prec q_j$ only. For the purpose of illustration, we project the queue stability boundaries of the two queues into a two-dimensional space, as shown in Figures 2. With the definition of $RS_1$ system and the assumption that $q_i \asymp q_j$ and $q_i \prec q_j$ only, the two queues' stability boundaries can only intersect in a way as shown is Figure 2(a). However, it turns out that the situation illustrated in Figure 2(a) is impossible. Consider any two linear increasing paths on which $q_i$ become unstable first, e.g., $L_1$ and $L_2$ in Figure 2(a). After $L_1$ and $L_2$ go beyond $q_i$'s stability boundary and intersect with $q_j$'s stability boundary at $p_1$ and $p_2$, $q_j$'s stability should not be affected by $q_i$. In other words, once beyond $q_i$'s stability boundary, $q_j$'s stability is independent of $q_i$ on $L_1$ and $L_2$. Therefore, the instability of $q_j$ on both $L_1$ and $L_2$ should be caused by the same value of $\lambda_2$, i.e., the $\lambda_2$ components of $p_1$ and $p_2$ are the same. It implies that the stability boundary of $q_j$ should be a horizontal line parallel to the $\lambda_i$ axis, as shown in 2(b). Now consider any traffic point $p_a$ in the area $A$ shown in Figure 2(b), which corresponds to the instability area of $q_i$ and stability area of $q_j$. Now increase the traffic point and by increasing $\lambda_i$ only and keeping $\lambda_j$ unchanged till reaching $p_b$, we can enter the stability region of $q_i$ $B$ again, and this is obviously not true. Therefore, in an $RS_1$ system, the first item is true.

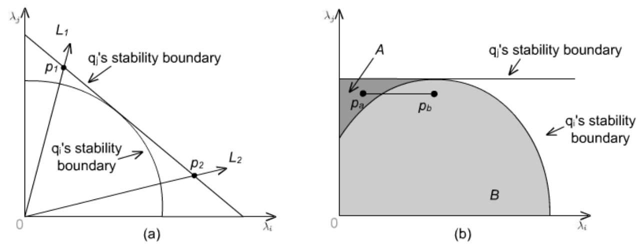



**Figure 2: Proof of $RS_1$ (1).**

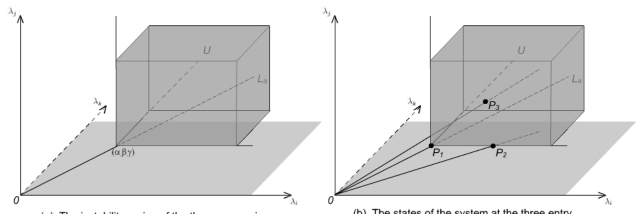
The falseness of (2) implies that there exists at least one linear increasing path $L_1$ on which a queue, say $q_i$, receives no service from the server when it is unstable. The consequence is that on this path, all other queues, stable or not, receive the entire server's capacity, i.e., all other queues have higher priority than $q_i$. However, based on (1) we know that any two queues relate to each other according to one of the three relative stability relations on some increasing path. Then it is possible to find at least one linear increasing path $L_2$ on which $q_i$ is the most stable queue. When $q_i$ become unstable on $L_2$, it still receives some service. The existence of both $L_1$ and $L_2$ implies that the server's service policy for $q_i$ is path dependent, which is generally not true. Therefore, item (2) is true.

To prove (3), we first show that the instability region of all queues of an $RS_1$ system is an open $n$-dimensional rectangle in the traffic space. Based on the definition of $RS_1$, there exists one and only one linear increasing path $L_a$ on which all queues are as stable as one another. In other words, on $L_a$, there exists a traffic point which is the stability boundary of all queues. Assume that traffic point $p_1$ has the coordinate $(\alpha, \beta, ..., \gamma)$. Now consider all linear increasing paths on which $q_1$ is the most stable queue. Based on the arguments in the proof of item (1), on all these paths the stability boundaries of $q_1$ should be independent of all other queues, and therefore these stability boundaries form a $n$-dimensional hyperplane $\Delta$ with $\lambda_1 = x$, where $x$ is a constant. By noting that point $p_1$ is also on the hyperplane, we have the $\Delta : \lambda_1 = \alpha$. The same argument is true for all other queues. Therefore, for each $q_i$, its queue stability boundary in the region, where the queue itself is the most stable queue, is a $n$-dimensional hyperplane with $\lambda_i = y$, where $y$ is a constant, and the value of $y$ can be determined by the point $p_1$. Consequently, the instability region of all queues $U$ is an open $n$-dimensional rectangle such that $U = \{(\lambda_1, \lambda_2, ..., \lambda_n) \in Z^n : \alpha < \lambda_1, \beta < \lambda_2, ..., \gamma < \lambda_n\}$. In Figure 3(a) we show the instability region $U$ for a three-queue $RS_1$ system. The increasing path $L_a$ is the one on which $q_i$, $q_j$, and $q_k$ are as stable as one another, and on $L_a$, the stability boundary of all queues is the traffic point $(\alpha, \beta, \gamma)$.

Now consider that the system has entered the region $U$ through two different entry points, e.g., $p_1 : (\alpha, \beta, ..., \gamma)$, and $p_2 : (\lambda_1, \beta, ..., \gamma)$, where $\alpha \leq \lambda_1$. Based on the argument that a queue's service rate equals to its arrival rate at its queue stability boundary, and the server's capacity is fixed and should be equal to the sum of the service rates of all queues when all queues become unstable, we can conclude that the service rate of $q_1$ is the same at both $p_1$ and $p_2$. Therefore, the system has the same states at $p_1$ and $p_2$, and has the same states when it enters the instability region of all queues through $p_1$ and $p_2$. $P_2$ is a line, i.e., the intersection points of hyperplanes $\lambda_2 = \beta$, ..., $\lambda_n = \gamma$ in region $U$. The same argument applies to all the entry points of the form $(\alpha, \lambda_2, ..., \gamma)$, ..., and $(\alpha, \beta, ..., \lambda_n)$. Now consider an entry point with the form of $p_3 : (\lambda_1, \lambda_2, \delta, ..., \gamma)$, where $\alpha \leq \lambda_1$ and $\beta \leq \lambda_2$. We can repeat the arguments recursively to show that any entry points with the form $p_3 : (\lambda_1, \lambda_2, \delta, ..., \gamma)$ has the same states as entry points $p_2 : (\lambda_1, \beta, \delta, ..., \gamma)$ (or $(\alpha, \lambda_2, \delta, ..., \gamma)$), and therefore has the same states as $p_1 : (\alpha, \beta, ..., \gamma)$. Repeat the arguments again and at last, we can show that any entry point to the region $U$ with the form $(\lambda_1, \lambda_2, ..., \lambda_n) \in Z^n : \alpha \leq \lambda_1, \beta \leq \lambda_2, ..., \gamma \leq \lambda_n$ will have the same states as $P_1 : (\alpha, \beta, ..., \gamma)$. Therefore, the system in the region $U$ has unique state. In Figure 3(b) we show the three kinds of entry points for a three-queue $RS_1$ system. The system states are the same at $p_1$ and $p_2$s, as well as at $p_2$s and $p_3$s. It implies that the system has unique state at all entry points to the region $U$. Therefore, in region $U$, the system has unique state.

Item (4) is a direct result of item (3). In fact, any traffic point in the instability region causes the system at the same unique state. Then obviously any traffic point in the instability region is path independent. □

The third class turns out to be the most complex. Same as the $RS_1$-multiqueue systems, as-stable-as paths exist for this class. However, this class loses the uniqueness. More precisely, the as-stable-as paths are *everywhere* in the sense that they are on an infinite number of hyperplanes. In the following theorem, we prove that for any two queues, if the intersection points of their stability boundaries are more than one, then the total number of the intersection points must be uncountably infinite. We therefore refer this class to as $RS_\infty$-multiqueue systems.




(a). The instability region of the three queues is an open rectangle.

(b). The states of the system at the three entry points $P_1$, $P_2$, and $P_3$ are the same.

**Figure 3: Proof of $RS_1$ (3).**

THEOREM 3. *In an $RS_\infty$-multiqueue system the intersection points of any two queues' stability boundaries are uncountably infinite.*

PROOF. If the number of intersection points of any two queues' stability boundaries is more than one, we just need to prove the number cannot be countable. To do so, we consider a counterexample of an $RS$ system with only two queues $q_i$ and $q_j$. If the number of the intersection points of the two queues is more than one and are countable, then in general we have two cases of their stability boundaries as shown in Figure 4(a) and (b). In Figure 4(a), $q_i$'s stability boundary is always within $q_j$'s stability in a certain region, while in Figure 4(b) the queues' stability boundaries cross over each other in a certain region. Now consider in Case 1, two increasing paths $L_a$ and $L_b$ go beyond $q_i$'s stability boundary and intersect with $q_j$'s stability boundary at points $p_a$ and $p_b$. Using the argument that after $q_i$ become unstable, $q_j$'s stability should be independent of $q_i$; therefore, the $\lambda_j$ components at $p_a$ and $p_b$ should be the same, which implies that $q_j$'s stability boundary should be parallel to $\lambda_i$'s axis, as shown in Figure 4(c). However, it is easy to find that Figure 4(c) is impossible because an increasing path from $p_1$ to $p_2$ in Figure 4(c) will cause $q_i$'s state changed from an unstable state to a stable state. Similarly, if the two queues' stability boundaries cross over each other, then they should be like the one shown in Figure 4(d), which is again impossible because an increasing path from $p_1$ to $p_2$ can also cause $q_i$'s state to change from an unstable state to a stable state. □

Based on discussion above, we can first determine for a given $RS$ system its class based on the definition of the different classes. The second method is to see whether the system will have a unique state or not in the instability region of all queues. Both $RS_0$ and $RS_1$ classes have unique system state while $RS_\infty$ does not have. Then we can check whether each queue in the system has guaranteed service or not in order to distinguish the class between $RS_0$ and $RS_1$. In an $RS_\infty$ system, there will be an infinite number of entry points to the instability region of all queues. On these entry points the system states may be different; therefore the system will have different states in the instability region of all queues. For example, consider a 2-queue system in which a cyclic server always serves at most $M_i$ requests from $q_i$ and at most $M_j$ requests from $q_j$ in every cycle. Besides, the server can also serve additional $M_a$ requests in every cycle

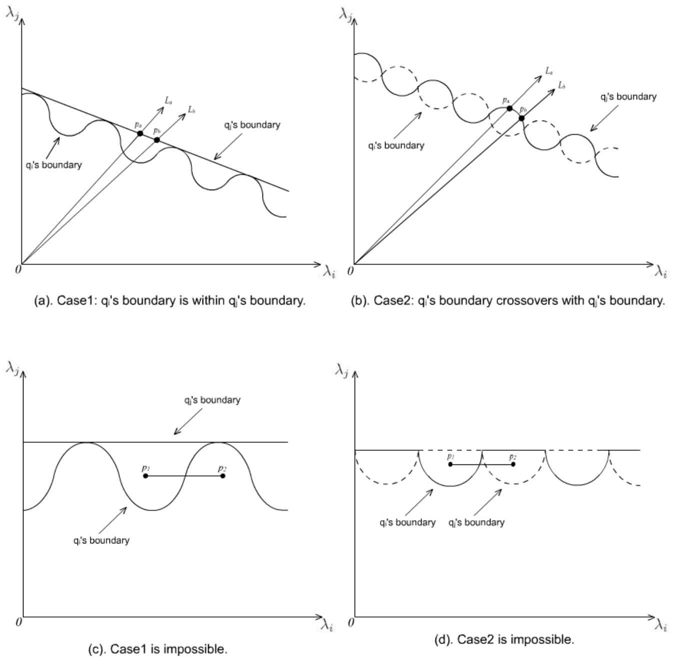



**Figure 4: Proof of $RS_\infty$.**

which can be freely grabbed by either queue. The stability regions for the individual queues for this type of service policy are shown in Figures 1(c)-(d). Compared with Figures 1(a)-(b), each queue is similarly guaranteed with a certain service rate. However, unlike the previous case, there are infinite number of as-stable-as paths. That is, while the $RS_1$-multiqueue system has a as-stable-as path, the $RS_\infty$-multiqueue system has a as-stable-as *region*. Furthermore, the traffic space for this space still has the $q_i \succ q_j$ region and the $q_i \prec q_j$ region. Another interesting point of the $RS_\infty$ class is that a queue may or may not have guaranteed service once becoming unstable. In the above example, both queues have guaranteed service. On the other hand, consider a polling system with an unlimited service policy. This system belongs to $RS_\infty$ class because the service provided to each queue is not limited, which implies that there is no guaranteed service rate. Consequently, a queue may or may not receive service when it becomes unstable, i.e., different states in the instability region. In other words, $\lambda_{g,i} = \lambda_{g,j} = 0$ in Figures 1(c)-(d). As a result, it is easy to see that the two stability regions are the same, and the $q_i \succ q_j$ region and the $q_i \prec q_j$ region do not exist.

In Figure 5, we also show the system stability region for both $RS_1$-multiqueue system and $RS_\infty$-multiqueue system within which both queues are stable. In (a), the stability region is concave which is typical for polling systems with limited service policies. However, some other systems, such as buffered ALOHA, would have a convex stability region. In the remaining of this paper, we restrict ourselves to the $RS_1$-multiqueue systems, and leave the other two classes to the forthcoming papers.

# 5. RELATIVE STABILITY FOR THE $RS_1$-MULTIQUEUE SYSTEMS

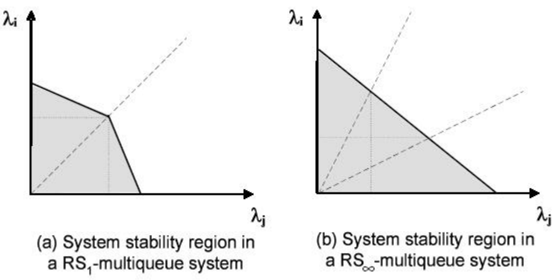



**Figure 5: System stability regions for an $RS_\infty$-multiqueue system of 2 queues.**

From the discussions above, we can derive more general results for the $RS_1$-multiqueue systems, to be presented in the next two theorems.

THEOREM 4. *In an $RS_1$-multiqueue system, $q_i$ and $q_j$ are as stable as each other if and only if their direction components satisfy $\frac{k_i}{k_j} = c_{i,j}$. That is, $q_i \asymp q_j \Leftrightarrow \frac{k_i}{k_j} = c_{i,j}$, where $c_{i,j}$ is a constant which is determined by the system parameters.*

PROOF. For an $RS_1$-multiqueue system with $n$ queues, if $q_i \asymp q_j$ on any two paths $L_{K1}$ and $L_{K2}$, then $L_{K1}$ and $L_{K2}$ belong to a unique $(n-1)$-dimensional hyperplane. For the traffic space and linear increasing paths under consideration, the analytical form of such a hyperplane satisfies $k_i/k_j =$ constant. By denoting the constant as $c_{i,j}$, we complete the proof. □

THEOREM 5. *In an $RS_1$-multiqueue system, $q_i \prec (\succ) q_j \Leftrightarrow \frac{k_i}{k_j} > (<)\, c_{i,j}$.*

PROOF. As the two cases are symmetric, we prove only the case of $q_i \prec q_j$. 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. 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 Theorem 4 that this conclusion is not true. As a result, $q_i \prec q_j$ holds for all paths in this partition. □

An important consequence of Theorem 4 and Theorem 5 is that the relation of two queues based on their degrees of stability in an $RS_1$-multiqueue system are not affected by other queues. The constant $c_{i,j}$ can be easily determined by finding a simple increasing path $L_K$ on which $q_i \asymp q_j$. Therefore, $c_{i,j} = C_i^K / C_j^K$, where $C_i^K$ and $C_j^K$ are the MASRs of $q_i$ and $q_j$ on $L_K$, respectively.

Moreover, we can extend the result that all the as-stable-as paths for 2 queues form a $(n-1)$-hyperplane. In general, all the as-stable-as paths for $m \leq n$ queues form a $(n-m+1)$-hyperplane. Therefore, there is only a single path for which all queues are as stable as each other, as stated in the following corollary.

COROLLARY 1. *In an $RS_1$-multiqueue system, there is only one increasing path $L_K$ for which all queues are as stable as one another and this $L_K$ is given by $\frac{k_1}{C_1^K} = \frac{k_1}{C_2^K} = \cdots = \frac{k_n}{C_n^K}$.*

Equipped with the general relative stability results, we will next apply them to derive the actual criteria for comparing queues based on their degrees of stability for three specific systems. The result is an order of queues, and the queues at both ends of the stability order are referred to as *least stable queue* (LSQ) and *most stable queue* (MSQ). Generally, there could be multiple LSQs and multiple MSQs on a path. For the case in Corollary 1, all queues are MSQs as well as LSQs.

# 6. RELATIVE STABILITY ANALYSIS FOR TWO SYSTEMS

## 6.1 Polling Systems with Limited Service

Consider a polling system with $n$ $RS$-queues and a single server. Besides, each queue has an unlimited buffer. The arrival rate at $q_i$ is $\lambda_i$. The service times are also generally distributed with the same mean value $b$. The server polls the queue in a predefined order. The server employs a *gated M-limit service policy.* That is, the server can serve those requests which it finds at $q_i$ upon its arrival, up to $M_i$ of them. Any requests arriving after the start of the service can only be served in the server's next visit. Assume that there is a random walk time for the server to switch from $q_i$ to $q_{i+1}$ with $s_i$ as the mean value. Define $EC$ as the mean *cycle time* which is time interval elapsed between two consecutive server visits to a queue. It is well known that $EC$ is independent of the choice of the reference queue [7].

Each $q_i$ in the system at any state has a guaranteed service rate of $M_i/EC$. To obtain the conditions under which $q_i$ is more stable than or as stable as $q_j$, according to Theorem 4 and Theorem 5, we only need to find out the value of $c_{i,j}$ for $q_i$ and $q_j$. Because any $L_K$ on which $q_i \asymp q_j$ will provide us the same value of $c_{i,j}$, we consider the path on which $q_i \asymp q_j$ and that they are the only MSQs. The result is stated in the following theorem.

THEOREM 6. *In a polling system with gated limited service policies, $q_i \succeq q_j$ on an increasing path $L_K$ iff $\frac{k_i}{k_j} \leq \frac{M_i}{M_j}$.*

PROOF. We consider a $L_K$ on which $q_i \asymp q_j$ and $q_i \succ q_k, k \neq i, j$. Then the values of $C_i^K$ and $C_j^K$ are given by $M_i/(\sum_l (M_l b + s_l))$ and $M_j/(\sum_l (M_l b + s_l))$, respectively. This is so because when both $q_i$ and $q_j$ achieve their MASRs ($M_i$ and $M_j$ per cycle time), all other queues have already become unstable, according to the given relative stability among the queues. Together with the assumption that $q_i \asymp q_j$ and Theorem 4, $L_K$ satisfies $\frac{k_i}{M_i} = \frac{k_j}{M_j}$. As a result, $q_i \asymp q_j \Leftrightarrow \frac{k_i}{k_j} = \frac{M_i}{M_j}$. Based on Theorem 5, we also have $q_i \succeq q_j \Leftrightarrow \frac{k_i}{k_i} \leq \frac{M_i}{M_j}$. □

## 6.2 Slotted Buffered ALOHA System

In a slotted buffered ALOHA system, $n$ queues share a broadcast channel (server). The channel is time slotted, and a slot duration is equal to a packet transmission time. Each nonempty queue $q_i$ transmits a packet with a nonzero probability $p_i$ in a slot. If two or more queues transmit simultaneously, then a collision occurs and the packets must be retransmitted in the future. If exactly one packet is transmitted in a slot, then a successful transmission takes place. By using a similar approach as for the polling model, we obtain the following results.

THEOREM 7. *In the slotted ALOHA system, $q_i \succeq q_j$ on an increasing path $L_K$ iff $\frac{k_i(1-p_i)}{p_i} \leq \frac{k_j(1-p_j)}{p_j}$.*

PROOF. We consider again an increasing path $L_K$ on which $q_i$ and $q_j$ are as stable as each other, and both are the only MSQs. Then, the values of $C_i^K$ and $C_j^K$ are given by $p_i \Pi_{l \neq i}(1-p_l)$ and $p_j \Pi_{l \neq j}(1-p_l)$, respectively. They represent the successful transmission probabilities of $q_i$ and $q_j$, given that all other queues are never empty. Again, this is so, because other queues are already unstable when $q_i$ and $q_j$ have attained their MASRs. Together with the assumption that $q_i \asymp q_j$ and Theorem 4, the $L_K$ satisfies $\frac{k_i(1-p_i)}{p_i} = \frac{k_j(1-p_j)}{p_j}$. Based on Theorem 5, we conclude that $q_i \succeq q_j \Leftrightarrow \frac{k_i(1-p_i)}{p_i} \leq \frac{k_j(1-p_j)}{p_j}$. □

# 7. ABSOLUTE STABILITY ANALYSIS

In this section we apply the relative stability results to the analysis of absolute stability for the polling system. We start with the queue stability problem and the results can later be used to obtain the system stability conditions. In do so, we consider the stability condition of a target queue $q_t$ on a linear increasing path $L_K$. By applying the Loynes' theorem to $q_t$ [16], the queue is stable if $\lambda_t < C_t^K$ and unstable if $\lambda_t > C_t^K$. The remaining task is therefore to obtain $C_t^K$.

All the queues can be categorized into three groups based on the outcomes of comparing their degrees of stability with $q_t$'s on $L_K$. The $\mathcal{M}_t$ set consists of the queues that are more stable than $q_t$. On the other hand, $\mathcal{L}_t$ set consists of the queues that are less stable than $q_t$. Finally, the $\mathcal{A}_t$ set consists of the queues that are as stable as $q_t$. The $\mathcal{M}_t$ and $\mathcal{L}_t$ sets can be empty. With the 3 sets of queues, we can obtain the mean cycle time when $q_t$ attains its MASR as follows.

$$EC = \sum_{q_l \in (\mathcal{M}_t \cup \mathcal{A}_t)} (u_l b + s_l) + \sum_{q_l \in \mathcal{L}_t} (M_l b + s_l), \qquad (2)$$

where $u_l \leq M_l$ is the average number of requests served at $q_l \in (\mathcal{M}_t \cup \mathcal{A}_t)$. In Eq. (2), the first term corresponds to the time incurred by a set of stable queues whereas the second term correspond to the time incurred by a set of unstable queues. Since the queues in $\mathcal{M}_t \cup \mathcal{A}_t$ are stable, we have $u_l = \lambda_l EC, q_l \in \mathcal{M}_t \cup \mathcal{A}_t$. By substituting $u_l$ into Eq. (2) we obtain

$$EC = \frac{\sum_{q_l \in \mathcal{L}_t} M_l b + s}{1 - \sum_{q_l \in (\mathcal{M}_t \cup \mathcal{A}_t)} \lambda_l b}, \qquad (3)$$

where $s = \sum s_l$. By applying Loynes' theorem, $q_t$ is stable on $L_K$ $\lambda_t < \frac{M_t}{EC}$.

Based on the results above for a path, we can now consider a set of paths that have the same sets of $\mathcal{M}_t, \mathcal{L}_t$, and $\mathcal{A}_t$. If $\Gamma_o \equiv (\mathcal{M}_{t,o}, \mathcal{L}_{t,o}, \mathcal{A}_{t,o})$ is a particular set of $\mathcal{M}_t, \mathcal{L}_t$, and $\mathcal{A}_t$, we denote the set of paths that satisfy $\Gamma_o$ by $P(\Gamma_o)$. And, we have the following results.

LEMMA 2. *The target queue $q_t$ is stable on $L_K \in P(\Gamma_o)$, where $\Gamma_o \equiv (\mathcal{M}_{t,o}, \mathcal{L}_{t,o}, \mathcal{A}_{t,o})$, if*

$$\lambda_t < \frac{M_t}{EC}, \qquad (4)$$

*where*

$$EC = \frac{\sum_{q_l \in \mathcal{L}_{t,o}} M_l b + s}{1 - \sum_{q_l \in (\mathcal{M}_{t,o} \cup \mathcal{A}_{t,o})} \lambda_l b}. \quad (5)$$

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

PROOF. From the above discussion, we know that $EC$ is well defined. $q_t$'s average service rate is therefore given by $\frac{u_t}{EC}$ and it reaches the maximum when $u_t = M_t$. By Loynes' theorem, the queue is stable if Eq. (4) holds.

THEOREM 8. *$q_t$'s stability region in the whole traffic space is given by $\cup_{all\ possible\ \Gamma_o} R(\Gamma_o)$, where $R(\Gamma_o)$ is $q_t$'s stability region for the set of paths $P(\Gamma_o)$.*

The above theorem is a direct consequence of Lemma 2.

An approach to deriving the system stability conditions from the queue stability conditions is to consider the set of paths for which $q_i$ is the LSQ, denoted by $P_i^l$, and to obtain the corresponding stability region, denoted by $R_i^l$. Since the system stability boundary point on a path is the same as a LSQ's boundary point, the system stability region can be expressed as a union of $R_i^l, i = 1, \ldots, n$. Therefore, we have the following system stability result.

THEOREM 9. *The polling system is stable in the region $\cup_{i=1,\ldots,n} R_i^l$, where $R_i^l$ is given by $\cup_{L_K \in P_i^l} \{\lambda_i < \frac{M_i}{EC}\}$, where $EC = \frac{s}{1-\sum_{i=1,\ldots,n} \lambda_i b}$.*

PROOF. According to Lemma 2, $q_i$ is stable on a path in $P_i^l$ if $\lambda_i < \frac{M_i}{EC}$. Thus, we can obtain $R_i^l$. Since each queue is a LSQ in some nonempty partition, the entire system stability region is a union of $R_i^l, i = 1, \ldots, n$. □

Instead of performing a set union operation as in the last method, one can also take an intersection all $n$ queue stability regions to yield the system stability region, because only the regions correspond to the LSQs will remain after the intersection operation.

# 8. CONCLUSIONS

In this paper we have introduced the concepts of degree of stability and relative stability for a single-server, multiqueue system. The degree of stability measures how stable a queue is. By comparing the queues' degrees of stability, we can determine the relative stability relation among the queues. We have analyzed and derived very general relative stability conditions for a class of $RS$-multiqueue systems. In particular, based on the number of ways that two queues can be as stable as each other, we classify the $RS$-multiqueue systems into three classes and provide properties for each class. Both the approach and the models are general enough to cover many different systems. We have illustrated the approach to a polling system with a limited service policy and a buffered ALOHA system. Lastly, we have shown how the relative stability results can facilitate an absolute stability analysis.

## Acknowledgment

The work described in this paper was partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 5146/01E). We also thank the anonymous reviewers for their helpful and critical comments.

# 9. REFERENCES

[1] K. Ogata, *Discrete-Time Control Systems*, 2nd ed. Prentice Hall, 1994.

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

[3] K. Ogata, *Modern Control Engineering*, 4th ed. Prentice Hall, 2001.

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

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

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

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

[8] 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.

[9] O. Ibe and X. Cheng, "Stability conditions for multiqueue systems with cyclic service," *IEEE Trans. Auto. Control*, vol. 33, no. 1, pp. 102–103, Jan. 1988.

[10] 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.

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

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

[13] 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.

[14] R. Chang and S. Lam, "A novel approach to queue stability analysis of polling models," *Performance Evaluation*, vol. 40, no. 4, pp. 27–46, March 2000.

[15] ——, "Per-queue stability analysis of a random access system," *IEEE Trans. Automatic Control*, vol. 46, no. 9, pp. 1466–1470, Sept 2001.

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

[17] R. Cruz, "A calculus for network delay, part I: network elements in isolation," *IEEE Trans. Information Theory*, vol. 37, no. 1, pp. 114–131, Jan 1991.

[18] C. Chang, "Stability, queue length, and delay of deterministic and stochastic queueing networks," *IEEE Trans. Auto. Control*, vol. 39, no. 5, pp. 913–931, May 1994.

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