up response of the pendulum starting at nearly the vertically downward position, with the remaining initial conditions zero. Notice that the response is very fast without any initial swinging of the pendulum.

The following remarks are in order.

- It is clear from the simulations that the stabilization mechanism of our controller consists of spinning-up the disk inertia to lift the pendulum, which might impose some unrealistic values to the disk speed. This should be contrasted with the alternative method of [9]—also studied in [1], [3]—where the energy is first pumped-up through a balancing motion before lifting the pendulum. Two drawbacks of the latter approach are the slow convergence and the need to switch the controller close to the upward position. From the theoretical viewpoint both methods also differ, our controller (as well as the one reported in [8]) stabilizes the *equilibrium point*, while the energy-pumping methods stabilizes the homoclinic orbit, hence the need for the switching.
- Although we have solved the stabilization problem of the system (10) with any prescribed saturation of the control, when we come back to the original disk inertia pendulum (9), we have to add $\sin(x_1)$ to the above control. So the above procedure does not give an answer to the problem where the maximal torque that the motor can deliver is smaller than the maximal gravity torque. Simulations and experiments have shown that stability cannot be guaranteed if we impose this saturation limit.

REFERENCES

[1] K. Astrom and K. Furuta, "Swinging up a pendulum by energy control," in *Proc. 13th IFAC World Congr.*, vol. E, San Francisco, CA, 1996, pp. 37–42.

[2] A. Isidori, *Nonlinear Control Systems*, 3rd ed. New York: Springer-Verlag, 1995.

[3] A. Fradkov, "Swinging control of nonlinear oscillations," *Int. J. Control*, vol. 64, no. 6, pp. 1189–1202, 1996.

[4] R. Freeman and P. Kokotović, *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques*. Boston, MA: Birkhäuser, 1996.

[5] M. Krstić, R. Sepulchre, and L. Praly, "On The Measure of Global Invariant Manifold for Fixed Points," University of California, Santa Barbara, CA, Tech. Rep. CCEC 95-0228, 1995.

[6] M. Janković, R. Sepulchre, and P. Kokotović, "Constructive Lyapunov stabilization of nonlinear cascade systems," *IEEE Trans. Automat. Contr.*, vol. 41, pp. 1723–1735, Dec. 1996.

[7] F. Mazenc and L. Praly, "Adding integrations, saturated controls and global asymptotic stabilization for feedforward systems," *IEEE Trans. Automat. Contr.*, vol. 41, pp. 1559–1578, Nov. 1996.

[8] R. Ortega and M. Spong, "Stabilization of underactuated mechanical systems via interconnection and damping assignment," in IFAC Workshop on Lagrangian and Hamiltonian Methods in Nonlinear Systems, Princeton, NJ, March 2000.

[9] M. Spong and L. Praly, "Control of underactuated mechanical systems using switching and saturations," in *Lecture Notes in Control and Information Sciences*. New York: Springer-Verlag, 1997, vol. 222.

[10] E. Sontag, "A universal construction of Artstein's theorem on nonlinear stabilization," *Syst. Control Lett.*, vol. 13, pp. 117–123, 1989.

[11] A. Gelig, G. Leonov, and V Yakubovich, *Stability of Nonlinear Systems with Nonunique Equilibria* (in Russian). Moscow, Russia: Nauka, 1978.

Per-Queue Stability Analysis of a Random Access System

Rocky K. C. Chang and Sum Lam

***Abstract*—In this note, we have extended previous studies of the system stability of buffered ALOHA systems to study an individual queue's stability, i.e., per-queue stability. The main result obtained in this work is a necessary and sufficient per-queue stability condition, which can be computed analytically only for several cases. For other noncomputable cases, we have evaluated several inner and outer bounds. They are generally quite tight for not-so-asymmetric systems.**

***Index Terms*—ALOHA, multiaccess systems, per-queue stability, queue stability ordering, system stability.**

I. INTRODUCTION

Stability analyses of single-resource–multiple-queue systems, such as random access protocols, polling schemes, and token-passing rings, have been studied quite extensively in the past. By stability, we mean that the queue length process of a queue with unlimited buffer space possesses a limiting distribution. Almost all previous studies in this area, however, concern stability of the whole system (system stability). Study of an individual queue's stability (per-queue stability), on the other hand, has hardly received any attention. The per-queue stability problem is more general than the system stability problem, because some queues may remain stable in an unstable system. Therefore, system stability, being a special case of per-queue stability, is inadequate to address the entire stability region of an individual queue. In this note, we consider per-queue stability of a buffered ALOHA system. Our goal is to obtain a necessary and sufficient per-queue stability condition as well as other related results.

So far, only system stability has been studied for the buffered ALOHA system. Computable system stability conditions are well known for two-queue systems and symmetric systems (e.g., see [1], [2]). Szpankowski employed Loynes' theorem and an induction approach to obtain necessary and sufficient system stability conditions for more than two queues, but the conditions are generally noncomputable [2]. Rao and Ephremides, on the other hand, obtained lower bounds for the system stability region using a simple concept of dominance [1]. Luo and Ephremides revisited the same problem and obtained a tighter bound [3]. Their main approach was based on an *instability rank* that helped construct appropriate dominant systems to obtain sufficient conditions. An instability rank, or a stability order, specifies the sequence of queues to become unstable when the system traffic increases according to a certain pattern.

Unlike previous work, our focus in this note is on per-queue stability. Besides describing the system, we present, in Section II, two preliminary results that are essential to obtaining a stability condition, say, for a target queue q_t. We first obtain in Lemma 1 q_t's necessary and sufficient stability condition for a path with a *known* stability order. We then use this result to obtain a criterion for comparing stabilities of any two queues in the system, and the criterion is essentially the same as the one obtained recently by Luo and Ephremides [3]. By combining

Manuscript received February 11, 2000; revised September 14, 2000 and December 5, 2000. Recommended by Associate Editor L. Dai. This work was supported in part by The Hong Kong Polytechnic University Central Research Grant 350/492.

The authors are with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: csrchang@comp.polyu.edu.hk; cssumlam@comp.polyu.edu.hk).

Publisher Item Identifier S 0018-9286(01)08811-0.

the results in Section II, we present, in Section III, q_t's entire stability region. However, only part of the region can be computed analytically. Therefore, we also evaluate several bounds for the noncomputable region. Our conclusions to this note can be found in Section IV.

II. Per-Queue Stability and Stability Ordering

We refer the formal definition of per-queue stability, Loynes' theorem, and the stationary and ergodic requirements for using Loynes' theorem to [2]–[4].

The ALOHA system considered here is the same as that in [2] and [3]. The system consists of a set of n queues. Denote the set as Q, and the ith member as $q_i, i = 1, \ldots, n$. Each queue has infinite buffers to store incoming packets. The packet arrival processes are general, but they must be stationary and ergodic. For q_i, λ_i is the packet arrival rate to the queue, and p_i is the transmission probability when the queue is not empty. In order to ensure a nonempty stability region, we limit the number of queues that employ $p = 1$ to at most one. Otherwise, none of the queues will be stable, regardless of their arrival rates. Let N_i^k be q_i's queue length at the beginning of slot $k \geq 1$ and $\mathbf{\Phi}_Q^k = (N_1^k, \ldots, N_n^k)$. It is well known that $\{\mathbf{\Phi}_Q^k\}_{k=1}^{\infty}$ is an irreducible and aperiodic Markov chain [5]. As a result, the system is stable if and only if the Markov chain is positive recurrent.

Before presenting the two main results in this section, we define several quantities and terminologies that will be used throughout this note. Consider a linear increase in the system traffic, or simply a *path*, according to a given vector $\mathbf{r} = (r_1, \ldots, r_n), r_i \geq 0$. Each operating point on that path can therefore be uniquely defined by an $\lambda^r \geq 0$, such that the point is given by $\mathbf{r}\lambda^r$, i.e., $\lambda_i = r_i\lambda^r, i = 1, \ldots, n$. For a given path $\mathbf{r}$, we define q_i's *critical value*, denoted by $\widehat{\lambda}_i^r \geq 0$, such that q_i is unstable if $\lambda^r > \widehat{\lambda}_i^r$, and q_i is stable if $\lambda^r < \widehat{\lambda}_i^r$ (we do not consider stability at the boundary). We call the point $\mathbf{r}\widehat{\lambda}_i^r$ q_i's *critical point* on $\mathbf{r}$. Clearly, the set of q_i's critical points on all possible paths constitutes the boundary surface of q_i's stability region, and the thickness of this surface is 0. For any two queues $q_i, q_j \in Q$, q_i is said to be *at least as stable as* q_j on $\mathbf{r}$, denoted by $q_j \preceq q_i$, when $\widehat{\lambda}_j^r \leq \widehat{\lambda}_i^r$. That is, either q_j becomes unstable first $\left(\widehat{\lambda}_j^r < \widehat{\lambda}_i^r\right)$ or both queues become unstable simultaneously $\left(\widehat{\lambda}_j^r = \widehat{\lambda}_i^r\right)$. The former is denoted by $q_j \prec q_i$, and the latter by $q_j = q_i$. It is easy to see that the $\preceq$ relation is a partial order as well as a total order on Q.

A stability order for a given path can be generally defined by $q_1 \preceq \cdots \preceq q_n$. Of course, $\preceq$ may be replaced by $\prec$ or $=$ for some paths. For a given path, it is also helpful to define $\mathcal{L}(t)$ and $\mathcal{M}(t)$ as a set of queues that are no more stable than q_t, and a set of queues that are more stable than q_t, respectively. More precisely, $\mathcal{L}(t) = \{q_k \in Q \,|\, q_k \neq q_t \text{ and } q_k \preceq q_t\}$ and $\mathcal{M}(t) = Q - \mathcal{L}(t) - \{q_t\}$. Therefore, q_t is a *most stable queue* (MSQ) if $\mathcal{M}(t) = \emptyset$. Similarly, q_t is a *second most stable queue* (SMSQ) if the queues in $\mathcal{M}(t)$ are as stable as each other. Moreover, q_t is a *least stable queue* (LSQ) if $\mathcal{L}(t) = \emptyset$ or if $q_t = q_k, \forall\, q_k \in \mathcal{L}(t)$. Lastly, we use the term *partition* to refer to a set of paths that share certain common queue stability ordering properties.

A. Per-Queue Stability Condition With a Known Stability Order

In Lemma 1, we present q_t's stability condition for a partition in which the paths share the same $\mathcal{M}(t)$. We denote this partition by $\mathcal{P}^{\mathcal{M}(t)}$. Moreover, we exclude the paths that $\lambda_t = 0$, because q_t is always stable on these paths, i.e., $\widehat{\lambda}_t^r = \infty$. From (1), $\widehat{\lambda}_t^r$ is actually q_t's success transmission probability in a dominant system, given that the queues in $\mathcal{M}(t)$ are stable. In this dominant system, the queues in $\mathcal{M}(t)$ are identical to those in the original system, but q_t and the queues in $\mathcal{L}(t)$, which are often referred to as *persistent queues*, will generate dummy packets whenever they are empty. We denote this dominant system by $\mathcal{S}^d(\mathcal{M}(t))$, and the queue length process in $\mathcal{S}^d(\mathcal{M}(t))$ by $\left\{\mathbf{\Phi}_Q^{k,d}(\mathcal{M}(t))\right\}_{k=1}^{\infty}$, which is still an irreducible and aperiodic Markov chain. The set-up of this dominant system is similar to that in [3], except for one subtle but important difference. That is, we differentiate the case of $q_i = q_t \preceq q_j$ from the case of $q_i \prec q_t \preceq q_j$ in setting up our dominant systems for q_t, simply because the two $\mathcal{M}(t)$s are different. Luo and Ephremides, on the other hand, did not differentiate them, and we will come back to this point when we discuss bounds in Section III. The equation for solving $\widehat{\lambda}_t^r$ is generally nonlinear (due to $\bar{p}_{\mathcal{M}(t)}^{r,\widehat{\lambda}_t^r}$); therefore, it cannot be solved analytically except for several special cases, which will be discussed in Section III, and bounds that are needed for others. Using Lemma 1 and other arguments, we then show in Lemmas 2–3 that $q_i \preceq (\succeq) q_j$, if and only if $\lambda_i/(p_i/1-p_i) \geq (\leq) \lambda_j/(p_j/1-p_j)$.

Lemma 1: Given an $\mathcal{M}(t)$, q_t's necessary and sufficient stability condition for a path $\mathbf{r} \in \mathcal{P}^{\mathcal{M}(t)}$, with possible exceptions of the boundaries, is given by

$$\lambda^r < \widehat{\lambda}_t^r, \text{ where } \widehat{\lambda}_t^r = \frac{p_t}{r_t}\bar{p}_{\mathcal{M}(t)}^{r,\widehat{\lambda}_t^r} \prod_{q_k \in \mathcal{L}(t)} (1 - p_k) \tag{1}$$

and $\bar{p}_{\mathcal{M}(t)}^{r,\widehat{\lambda}_t^r}$ is the probability that the queues in $\mathcal{M}(t)$ do not transmit in $\mathcal{S}^d(\mathcal{M}(t))$ at the operating point $\mathbf{r}\widehat{\lambda}_t^r$.

Proof: Given an $\mathcal{M}(t)$, we consider a path $\mathbf{r} \in \mathcal{P}^{\mathcal{M}(t)}$. The queues in $\mathcal{M}(t)$ are assumed to be stable. Otherwise, q_t and the queues in $\mathcal{L}(t)$ would become unstable. By applying Loynes' theorem to q_t in $\mathcal{S}^d(\mathcal{M}(t))$, q_t's necessary and sufficient stability condition in this dominant system is given by (1), with the possible exceptions of the boundaries [2], [4].

Next, we show that q_t's stability condition in the original system is also given by (1); that is, q_t's stability behavior remains the same in both systems. First, it is straightforward to show from their dominance relationship that q_t's stability in $\mathcal{S}^d(\mathcal{M}(t))$ implies its stability in the original system $\mathcal{S}$. This can be done by showing that q_t's queue length in $\mathcal{S}^d(\mathcal{M}(t))$ is no less than that in $\mathcal{S}$ for $k \geq 1$ when given identical initial system states. Second, we need to show that q_t's instability in $\mathcal{S}^d(\mathcal{M}(t))$ implies its instability in $\mathcal{S}$. By setting $\lambda^r > \widehat{\lambda}_t^r$, q_t is unstable in $\mathcal{S}^d(\mathcal{M}(t))$. Because of the stability order, the queues in $\mathcal{L}(t)$ are unstable in $\mathcal{S}^d(\mathcal{M}(t))$ as well; that is, $\lim_{k\to\infty} N_i^{k,d} = \infty$ for $q_i \in \mathcal{L}_o \cup \{q_t\}$, where $N_i^{k,d}$ is q_i's queue length in $\mathcal{S}^d(\mathcal{M}(t))$ at the beginning of the kth slot. Then, we apply a theorem from [6] to $\left\{\mathbf{\Phi}_Q^{k,d}(\mathcal{M}(t))\right\}_{k=1}^{\infty}$:

> Consider the Markov chain $\left\{\mathbf{\Phi}_Q^{k,d}(\mathcal{M}(t))\right\}_{k=1}^{\infty}$. Assume that $\lim_{k\to\infty} N_i^{k,d} = \infty$ for each $q_i \in \mathcal{L}(t) \cup \{q_t\}$ if the process starts from any state. Then, $\Pr\{N_i^{k,d} > 1, k \geq 1$, for each $q_i \in \mathcal{L}(t) \cup \{q_t\}\} > 0$ if the system starts at $k = 1$ with the queues in $\mathcal{L}(t) \cup \{q_t\}$ being nonempty.

From the above, we conclude that there is a set of sample paths of positive probability for which q_t and the queues in $\mathcal{L}(t)$ are indistinguishable in $\mathcal{S}$ and $\mathcal{S}^d(\mathcal{M}(t))$. In other words, they are also unstable in the original system. As a result, q_t's instability in $\mathcal{S}^d(\mathcal{M}(t))$ implies its instability in $\mathcal{S}$. □

Before leaving this section, we should also point out the following. If we consider a dominant system in which the set of persistent queues also includes some queues in $\mathcal{M}(t)$, then q_t's stability condition obtained from this dominant system is only sufficient, but not necessary, for q_t to be stable in the original system. This is because q_t's instability in the dominant system is caused by the instability of a more stable queue. On the other hand, if the set of persistent queues does not include q_t and possibly other queues in $\mathcal{L}(t)$, then q_t's stability condition obtained from this dominant system is only necessary, but not sufficient, for q_t to be stable in the original system. The sufficiency cannot be established, because q_t can still increase its arrival rate in the original system while staying in the stable condition.

B. Determining Queue Stability Order

In the following, we use the per-queue stability results obtained in Lemma 1 and other arguments to directly show that stabilities of any two queues can be compared solely based on their $\lambda/(p/1-p)$s. Using the concept of critical points, the proof for Lemma 2 significantly improves the proof for a similar result in our earlier work on polling models [7]. Although this result is essentially the same as the one obtained by Luo and Ephremides, there are a couple of differences. First, we directly show that $\lambda/(p/1-p)$ is both necessary and sufficient to determine the stability order, instead of only the sufficiency shown in [3] (with additional arguments, the necessity can also be established). Second, we explicitly consider the as-stable-as case which was not considered as a separate case in [3] (this can also be done by combining the cases of $\preceq$ and $\succeq$, and by employing additional arguments).

Lemma 2: In the ALOHA system, $q_i = q_j$ for any path for which $r_i/(p_i/1-p_i) = r_j/(p_j/1-p_j)$ holds.

Proof: When $r_i = r_j = 0$ for some paths, $q_i = q_j$ because $\lambda_i^r = \lambda_j^r = \infty$. When either $r_i = 0$ or $r_j = 0$, we also know that the queue with a zero arrival rate is more stable than the other. Therefore, Lemma 2 holds also for this case. In the rest of this proof, we thus assume $r_i, r_j > 0$. Denote the path that $q_i = q_j$ by $\tilde{\mathbf{r}} = (\tilde{r}_1, \ldots, \tilde{r}_n)$. In the following, we first prove that there exists at least an $\tilde{\mathbf{r}}$, and then show that $\tilde{r_i}/(p_i/1-p_i) = \tilde{r_j}/(p_j/1-p_j)$ must hold for such an $\tilde{\mathbf{r}}$.

We first assume that $\tilde{\mathbf{r}}$ does not exist; that is, the stability boundary surfaces of q_i and q_j do not intersect. Because the boundary surfaces are continuous, this assumption implies that either $\widehat{\lambda}_j^r > \widehat{\lambda}_i^r$ or $\widehat{\lambda}_i^r > \widehat{\lambda}_j^r$ holds for all possible paths. In other words, one queue is *always* more stable than the other. This conclusion is obviously not true for the ALOHA system with $r_i, r_j > 0$, thus contradicting the assumption that $\tilde{\mathbf{r}}$ does not exist.

Now we consider an $\tilde{\mathbf{r}}$. Because $q_i = q_j$ on $\tilde{\mathbf{r}}$, their critical points must coincide on $\tilde{\mathbf{r}}$

$$\begin{aligned}\frac{p_i(1-p_j)}{\tilde{r}_i}\bar{p}_{\mathcal{M}(i)}^{\tilde{r},\widehat{\lambda}_i^{\tilde{r}}} &\prod_{q_k\in\mathcal{L}(i)-\{q_j\}}(1-p_k)\\ &= \frac{p_j(1-p_i)}{\tilde{r}_j}\bar{p}_{\mathcal{M}(j)}^{\tilde{r},\widehat{\lambda}_j^{\tilde{r}}}\prod_{q_k\in\mathcal{L}(j)-\{q_i\}}(1-p_k).\end{aligned} \tag{2}$$

Moreover, on this path $\mathcal{M}(i) = \mathcal{M}(j)$, and $\mathcal{L}(i)-\{q_j\} = \mathcal{L}(j)-\{q_i\}$. With the above and $\widehat{\lambda}_i^{\tilde{r}} = \widehat{\lambda}_j^{\tilde{r}}$, $\bar{p}_{\mathcal{M}(i)}^{\tilde{r},\widehat{\lambda}_i^{\tilde{r}}} = \bar{p}_{\mathcal{M}(j)}^{\tilde{r},\widehat{\lambda}_j^{\tilde{r}}}$ for this path. As a result, (2) is simplified to $\tilde{r}_i/(p_i/1-p_i) = \tilde{r}_j/(p_j/1-p_j)$. By noting that this result does not depend on the values of the constant arrival rates for other queues, we conclude that this result is valid for any path in the parameter space. □

Corollary 1: All queues are as stable as each other for the path $r_1/(p_1/1-p_1) = \cdots = r_n/(p_n/1-p_n)$.

Proof: By applying Lemma 2 and the partial ordering properties of the $\preceq$ relation. □

Lemma 3: $q_j \prec (\succ) q_i$ for any paths that satisfy $r_j/(p_j/1-p_j) > (<) r_i/(p_i/1-p_i)$.

Proof: We again assume $r_i, r_j > 0$. Moreover, we consider only the case of $q_j \prec q_i$; the other case can be similarly proved. We consider a partition, in which the paths satisfy $r_j/(p_j/1-p_j) > r_i/(p_i/1-p_i)$. First, we can always find a path in this partition that gives $q_j \prec q_i$ by setting r_i to a sufficiently small value. Second, we claim that either $q_j \prec q_i$ or $q_i \prec q_j$ holds for all the paths in this partition. If this is not the case, the two stability boundaries should have at least one intersection. This then implies that there is at least an $\tilde{\mathbf{r}}$ in this partition, but we know from Lemma 2 that this conclusion is invalid. As a result, $q_j \prec q_i$ for all paths in this partition. □

Although q_t with a zero arrival rate is technically considered an MSQ, we will not consider this case further. Instead, in the rest of this note, we define an MSQ as one that has a nonzero arrival rate. Moreover, we will treat an ALOHA system with some zero arrival rates as one of a lower dimension.

III. Stability Conditions and Bounds

By combining the results obtained in Section II, in Theorem 1 we obtain the entire stability region for any queue in the system. However, the stability region can be computed only for several cases, as given in Corollary 2. For other cases, we obtain, in Corollary 3, separate sufficient and necessary conditions.

A. Stability Conditions and Simple Bounds

Theorem 1: q_t's entire stability region is given by

$$\bigcup_{\text{all possible } \mathcal{M}(t)s}\left\{\mathcal{P}^{\mathcal{M}(t)} \mid \lambda^r < \widehat{\lambda}_t^r,\ \text{where } \widehat{\lambda}_t^r = \frac{p_t}{r_t}\bar{p}_{\mathcal{M}(t)}^{r,\widehat{\lambda}_t^r}\prod_{q_k\in\mathcal{L}(t)}(1-p_k),\ \forall\,\mathbf{r}\in\mathcal{P}^{\mathcal{M}(t)}\right\}$$

where $\mathcal{M}(t) \neq \emptyset$ and $\mathcal{P}^{\mathcal{M}(t)}$ can be obtained from Lemmas 2–3.

Proof: By combining Lemmas 1–3. □

Corollary 2:

i) In a symmetric ALOHA system for which $p_i = p$ and $\lambda_i = \lambda, \forall\, i$, any queue in the system is stable if and only if

$$\lambda < p(1-p)^{n-1}. \tag{3}$$

ii) If q_t is an MSQ in a certain partition, then it is stable for any path $\mathbf{r}$ in the partition if and only if

$$\lambda^r < \frac{p_t}{r_t}\prod_{q_k\in Q-\{q_t\}}(1-p_k). \tag{4}$$

iii) If q_m is the only MSQ, and q_t is an SMSQ in a certain partition, then q_t is stable for any path $\mathbf{r}$ in the partition if and only if

$$\lambda^r < \frac{\prod_{q_k\in Q-\{q_m,q_t\}}(1-p_k)}{\frac{r_t}{p_t}+\frac{r_m}{1-p_t}}. \tag{5}$$

Proof: For case i), because all queues are as stable as each other, $\mathcal{M}(t) = \emptyset$ for any q_t and any path. As a result, $\bar{p}_{\mathcal{M}(t)}^{r,\widehat{\lambda}_t^r} = 1$, and (1) becomes (3).

For case ii), $\bar{p}_{\mathcal{M}(t)}^{r,\widehat{\lambda}_t^r} = 1$ for any path in the partition because $\mathcal{M}(t) = \emptyset$; as a result, (1) is reduced to (4).

For case iii), q_m's nonempty probability at q_t's critical point in $\mathcal{S}^d(\{q_m\})$ is given by $(\widehat{\lambda}_t^r r_m)/p_m\prod_{q_k\in Q-\{q_m\}}(1-p_k)$. Hence, for any path $\mathbf{r}$ in the partition,

$$\begin{aligned}\bar{p}_{\mathcal{M}(t)}^{r,\widehat{\lambda}_t^r} = &\left(1-\frac{\widehat{\lambda}_t^r r_m}{p_m\prod_{q_k\neq q_m}(1-p_k)}\right)\\ &+(1-p_m)\left(\frac{\widehat{\lambda}_t^r r_m}{p_m\prod_{q_k\neq q_m}(1-p_k)}\right).\end{aligned}$$

That is, q_m is either empty or not transmitting when it is not empty. After substituting the expression of $\bar{p}_{\mathcal{M}(t)}^{r,\widehat{\lambda}_t^r}$ into (1) as well as some manipulations, we arrive at (5). □

Corollary 3:

i) If q_m is the only MSQ and q_t is not an SMSQ in a certain partition, then q_t is stable for any path $\mathbf{r}$ in the partition if (5) holds.

ii) If q_t is stable in a certain partition characterized by a given $\mathcal{M}(t)$, then (6) holds for any path $\mathbf{r}$ in the partition

$$\lambda^r < \frac{\prod_{q_k \in \mathcal{L}(t)}(1-p_k)}{\frac{r_t}{p_t} + \frac{1}{1-p_t}\sum_{q_k \in \mathcal{M}(t)} r_k}. \tag{6}$$

Proof: For case i), as discussed at the end of Section II-A, a sufficient stability condition can be obtained from a dominant system in which its set of persistent queues is a superset of $\{q_t\} \cup \mathcal{L}(t)$. The smallest of such a superset, for which exact stability conditions can be obtained, is given by $Q - \{q_m\}$, where q_m is the only MSQ. As a result, (5) is a sufficient stability condition for this case.

For case ii), by substituting an upper bound of $\bar{p}^{r,\widehat{\lambda}_t^r}_{\mathcal{M}(t)}$ for any $\mathbf{r} \in \mathcal{P}^{\mathcal{M}(t)}$ that is adapted from [3]

$$\bar{p}^{r,\widehat{\lambda}_t^r}_{\mathcal{M}(t)} < 1 - \widehat{\lambda}_t^r \frac{\sum_{q_k \in \mathcal{M}(t)} r_k}{\prod_{q_k \in \mathcal{L}(t)\cup\{q_t\}}(1-p_k)}$$

into (1), we arrive at (6). □

By combining cases ii) and iii) in Corollary 2, we can obtain exact per-queue stability conditions for all queues for the following cases: (1) two-queue systems, (2) n-queue systems with $q_1 = q_2 = \cdots = q_n$, and (3) n-queue systems with $q_1 = \cdots = q_{n-1} \prec q_n$. Moreover, for other cases, we can always obtain stability conditions for MSQs and for SMSQs in some cases. As for the bounds, both the sufficient and necessary conditions are expected to be tight if the queue under consideration is closer to the SMSQ. The necessary condition is also sufficient when q_t is an MSQ, i.e., case (ii) of Corollary 2, or when q_t is an SMSQ and there is only one MSQ, i.e., case iii) of Corollary 2.

Lastly, in this section, we present, in Corollary 4, a system stability condition for a path, which, according to the stability order, is equivalent to an LSQ's stability condition. Separate sufficient and necessary system stability conditions can also be obtained by applying (5) to an LSQ and (6) to an LSQ, respectively.

Corollary 4: On a path $\mathbf{r}$, for which $Q_{\mathrm{LSQ}} \subseteq Q$ is a set of LSQs, the ALOHA system is stable, with possible exceptions of the boundaries, if and only if (7) is satisfied by a $q_i \in Q_{\mathrm{LSQ}}$.

$$\lambda^r < \widehat{\lambda}_i^r, \text{ where } \widehat{\lambda}_i^r = \frac{p_i}{r_i}\bar{p}^{r,\widehat{\lambda}_i^r}_{Q-Q_{\mathrm{LSQ}}} \prod_{q_k \in Q_{\mathrm{LSQ}}-\{q_i\}} (1-p_k). \tag{7}$$

The entire system stability region is given by

$$\bigcup_{\text{all possible } Q_{\mathrm{LSQ}}\text{s}} \left\{\mathcal{P}^{Q-Q_{\mathrm{LSQ}}} \mid (7) \text{ holds for a } q_i \in Q_{\mathrm{LSQ}}, \quad \forall\, \mathbf{r} \in \mathcal{P}^{Q-Q_{\mathrm{LSQ}}}\right\} \tag{8}$$

where $Q_{\mathrm{LSQ}} \neq \emptyset$.

Proof: By applying Theorem 1 to an LSQ, we obtain (7). For the second part, we divide the set of all possible paths into partitions, each of which is characterized by a unique Q_{LSQ}. The entire system stability region is, therefore, a union of the stability regions of LSQs in those partitions, which is given by (8). □

B. A Tighter Sufficient Condition and Numerical Results

To obtain a tighter sufficient per-queue stability condition, we borrow from [3] a lower bound for a nontransmission probability and adapt it for $\bar{p}^{r,\widehat{\lambda}_t^r}_{\mathcal{M}(t)}$. In [3], Luo and Ephremides considered a given stability order in the form of $q_1 \preceq \cdots \preceq q_n$, and skillfully obtained a sufficient stability condition for each queue from a dominant system. When applying their result to our per-queue stability condition, we have made two modifications. The first one is to distinguish the as-stable-as case from the less-stable-than case, because per-queue stability conditions are different for these two cases. The second is to apply the result to a path originating from the origin; that is, the queues' arrival rates increase at the same time on this path instead of only one queue's arrival rate increasing. Our approach facilitates the construction of the entire stability region. After making the modifications, we have for a path $\mathbf{r}$

$$\widehat{\lambda}_t^r \geq \underline{\widehat{\lambda}}_t^r, \text{ where } \underline{\widehat{\lambda}}_t^r = \frac{\prod_{q_k \in \mathcal{L}(t)}(1-p_k)}{\frac{r_t}{p_t} + \frac{1}{2}\sum_{q_j \in \mathcal{M}(t)} \left(\frac{p_j}{\underline{\widehat{\lambda}}_j^r r_j}\prod_{q_k \in \mathcal{L}(t)}(1-p_k) + \frac{1}{1-p_t}\right) r_j}. \tag{9}$$

Therefore, $\lambda_t < \underline{\widehat{\lambda}}_t^r$ ensures q_t's stability on path $\mathbf{r}$. We further note that (9) gives exact per-queue stability results for the three cases considered in Corollary 2. Moreover, if multiple queues are as stable as each other, their values of $\underline{\widehat{\lambda}}_t^r$ are the same.

TABLE I
NUMERICAL RESULTS FOR A THREE-QUEUE ALOHA SYSTEM

Cases	q_1	q_2	q_3
1. $r_1 = r_2 = 1, r_3 = 2$, $p_i = 0.5\ \forall i$	$\lambda^r < 0.125^*$	$\lambda^r < 0.1250^*$	(9): $\lambda^r < 0.1000$ (6): $\lambda^r < 0.1250$
2. $r_1 = r_2 = 1, r_3 = 4$, $p_i = 0.5\ \forall i$	$\lambda^r < 0.125^*$	$\lambda^r < 0.1250^*$	(9): $\lambda^r < 0.0714$ (6): $\lambda^r < 0.0833$
3. $r_1 = 1, r_2 = 2, r_3 = 3$, $p_i = 0.5\ \forall i$	$\lambda^r < 0.125^*$	$\lambda^r < 0.0833^*$	(9): $\lambda^r < 0.0714$ (5): $\lambda^r < 0.0625$ (6): $\lambda^r < 0.0833$
4. $r_1 = 1, r_2 = 3, r_3 = 6$, $p_i = 0.5\ \forall i$	$\lambda^r < 0.125^*$	$\lambda^r < 0.0625^*$	(9): $\lambda^r < 0.0455$ (5): $\lambda^r < 0.0357$ (6): $\lambda^r < 0.0500$
5. $r_1 = 1, r_2 = 1.5, r_3 = 2$, $p_1 = 0.5, p_2 = p_3 = 0.6$	$\lambda^r < 0.080^*$	$\lambda^r < 0.0800^*$	(9): $\lambda^r < 0.0750$ (6): $\lambda^r < 0.1043$ (6^+): $\lambda^r < 0.0800$

In Tables I and II, we present numerical results to evaluate the bounds for per-queue stability conditions: (5) and (9) for a sufficient condition, (6) for a necessary condition, and $(6)^+$ for an improved necessary condition, which will be explained later. Note that (5) can be applied only when there is only one MSQ on a path; therefore, bounds from (5) are missing from some cases in the tables. On the other hand, the results indicated by * are exact. The numerical results can be summarized as follows.

- Sufficient conditions computed by (9) are generally tighter than those from (5), except for q_6 in case 3 of Table II.
- By comparing case 3 and case 4 in Table I, sufficient conditions computed by (5) become loose when queues are more asymmetric or when the differences in $\lambda(1-p)/p$ increase. Similar results are also observed in Table II by comparing cases 3 and 4. In the less asymmetric case (case 3), (5) and (9) yield very similar results.
- Necessary conditions computed by (6) are tighter for more stable queues. For example, in case 5 of Table I, q_3's necessary stability boundary even lies outside q_2's stability boundary. A simple way of improving the bound's performance is to find the most inner bound by taking into consideration necessary conditions of the more stable queues. In this example, q_3's necessary condition is therefore given by $\lambda^r < 0.08$. Using this approach, the necessary conditions for the cases in Table II can also be improved significantly. They are labeled (6^+).

TABLE II
NUMERICAL RESULTS FOR A SIX-QUEUE ALOHA SYSTEM

Cases	q_1	q_2	q_3	q_4	q_5	q_6
1	$\lambda^r < 0.0156^*$	$\lambda^r < 0.0156^*$	$\lambda^r < 0.0156^*$	$\lambda^r < 0.0156^*$	$\lambda^r < 0.0156^*$	(9): $\lambda^r < 0.0114$
						(6): $\lambda^r < 0.0806$
						(6^+): $\lambda^r < 0.0156$
2	$\lambda^r < 0.0156^*$	$\lambda^r < 0.0156^*$	$\lambda^r < 0.0156^*$	$\lambda^r < 0.0156^*$	$\lambda^r < 0.0156^*$	(9): $\lambda^r < 0.0113$
						(6): $\lambda^r < 0.0735$
						(6^+): $\lambda^r < 0.0156$
3	$\lambda^r < 0.0156^*$	$\lambda^r < 0.0142^*$	(9): $\lambda^r < 0.0136$	(9): $\lambda^r < 0.0126$	(9): $\lambda^r < 0.0112$	(9): $\lambda^r < 0.0095$
			(5): $\lambda^r < 0.0130$	(5): $\lambda^r < 0.0120$	(5): $\lambda^r < 0.0112$	(5): $\lambda^r < 0.0104$
			(6): $\lambda^r < 0.0174$	(6): $\lambda^r < 0.0240$	(6): $\lambda^r < 0.0357$	(6): $\lambda^r < 0.0556$
			(6^+): $\lambda^r < 0.0142$	(6^+): $\lambda^r < 0.0142$	(6^+): $\lambda^r < 0.0142$	(6^+): $\lambda^r < 0.0142$
4	$\lambda^r < 0.0156^*$	$\lambda^r < 0.0130^*$	(9): $\lambda^r < 0.0120$	(9): $\lambda^r < 0.0111$	(9): $\lambda^r < 0.0098$	(9): $\lambda^r < 0.0084$
			(5): $\lambda^r < 0.0112$	(5): $\lambda^r < 0.0098$	(5): $\lambda^r < 0.0087$	(5): $\lambda^r < 0.0078$
			(6): $\lambda^r < 0.0149$	(6): $\lambda^r < 0.0195$	(6): $\lambda^r < 0.0278$	(6): $\lambda^r < 0.0417$
			(6^+): $\lambda^r < 0.0130$	(6^+): $\lambda^r < 0.0130$	(6^+): $\lambda^r < 0.0130$	(6^+): $\lambda^r < 0.0130$
5	$\lambda^r < 0.0156^*$	$\lambda^r < 0.0142^*$	(9): $\lambda^r < 0.0136$	(9): $\lambda^r < 0.0136$	(9): $\lambda^r < 0.0136$	(9): $\lambda^r < 0.0123$
			(5): $\lambda^r < 0.0130$	(5): $\lambda^r < 0.0130$	(5): $\lambda^r < 0.0130$	(5): $\lambda^r < 0.0120$
			(6): $\lambda^r < 0.0174$	(6): $\lambda^r < 0.0174$	(6): $\lambda^r < 0.0174$	(6): $\lambda^r < 0.0625$
			(6^+): $\lambda^r < 0.0142$	(6^+): $\lambda^r < 0.0142$	(6^+): $\lambda^r < 0.0142$	(6^+): $\lambda^r < 0.0142$

Case 1: $r_i = 1,\ i = 1, \cdots, 5, r_6 = 1.2; p_i = 0.5\, \forall i$

Case 2: $r_i = 1,\ i = 1, \cdots, 5, r_6 = 1.8; p_i = 0.5\, \forall i$

Case 3: $r_1 = 1, r_i = 1 + (i-1) \times 0.2,\ i = 2, \cdots, 6; p_i = 0.5\, \forall i$

Case 4: $r_1 = 1, r_i = 1 + (i-1) \times 0.4,\ i = 2, \cdots, 6; p_i = 0.5\, \forall i$

Case 5: $r_1 = 1, r_2 = 1.2, r_i = 1.4, i = 3, \cdots, 5, r_6 = 1.6; p_i = 0.5\, \forall i$

IV. CONCLUSION

We have obtained an exact per-queue stability condition for any queue on a given path. The stability boundaries for several cases, including two-queue systems, symmetric systems, and some special paths, are linear. Therefore, exact analytical conditions can be obtained for them. For other cases, the boundaries are nonlinear and cannot be obtained analytically. Therefore, we have also evaluated several inner and outer bounds. They are generally quite tight for not-so-asymmetric systems. Future work in this area includes designing tighter bounds by exploiting geometric properties of the stability region.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their useful comments.

REFERENCES

[1] R. Rao and A. Ephremides, “On the stability of interacting queues in a multiple-access system,” *IEEE Trans. Inform. Theory*, vol. 34, pp. 918–930, May 1988.

[2] W. Szpankowski, “Stability conditions for some distributed systems: Buffered random access systems,” *Adv. Appl. Prob.*, vol. 26, pp. 498–515, 1994.

[3] W. Luo and A. Ephremides, “Stability of N interacting queues in random-access systems,” *IEEE Trans. Inform. Theory*, vol. 45, pp. 1579–1587, May 1999.

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

[5] B. Tsybakov and W. Mikhailov, “Ergodicity of slotted ALOHA system,” *Probl. Peredachii Infor.*, vol. 15, pp. 73–87, 1979.

[6] L. Georgiadis, W. Szpankowski, and L. Tassiulas, “Stability analysis of quota allocation access protocols in ring networks with spatial reuse,” *IEEE Trans. Inform. Theory*, vol. 43, pp. 923–937, May 1997.

[7] R. K. C. Chang and S. Lam, “A novel approach to queue stability analysis of polling models,” *Perf. Eval.*, vol. 39, pp. 27–46, Mar. 2000.