Congested Banyan Network Analysis Using Congested-Queue States and Neighboring-Queue Effects

David M. Koppelman*
Louisiana State University, Baton Rouge, LA 70803

Abstract: A banyan network analysis technique is presented which more accurately models a congested network than other reported techniques. The analysis is based on the observation that a full queue (within the switching modules making up the network) causes traffic to back up. For a short time after becoming full the free space in the queue is limited to no more than one slot. A queue in such a condition is called *congested.* **Because of blocking the arrival rate to a congested queue is higher; this tends to maintain congestion. The arrival rate to a congested queue's dual is lower as is the service rate for queues feeding the congested queue. These effects are captured in the analysis. The state model used for a queue encodes congested as well as normal operation. Further, the model codes whether the connected next-stage and dual queues are congested. Network throughput computed with the model is closer to that obtained from simulations than other banyan analyses appearing in the literature, including those designed to model congestion. Further, the queue-occupancy distributions are much closer than other analyses, suggesting that the analysis better models conditions in congested banyan networks.**

1 Introduction

Because of their regular structure and their performance-determining role in computation and communication systems, banyan networks are a tempting target of analysis. Several analyses have been published since the networks were described by Peace [13] and Lawrie [10]. Early analyses considered unbuffered or single-buffered banyans [1,7]. Later work considered finite-buffered networks, the type of network considered here. Yoon, Lee, and Liu described one such analysis [15] in which each stage is represented by a single queue modeled by a Markov chain. This analysis, to be referred to as Yoon's analysis here, works well at low and moderate offered-traffic rates. However the predictions are less accurate at rates causing congestion. For many applications this is acceptable, as when congestion is rare. For others, performance at congestion determines system performance, as in a parallel computer running a communication-intensive algorithm. In these cases an accurate model of the network under congestion would help in estimating performance. An accurate model could also help in the development of new network designs or traffic management schemes.

Recent work specifically models banyan networks under congestion. One approach was to develop a model in which the service rate of a queue is dependent upon the fate of the head packet (next packet to be sent) in the previous cycle. The rationale is that a blocked head packet will more likely be blocked again in the next cycle. Such approaches have been used by Lin and Kleinrock [11], Mun and Youn [12], and Hsiao *et al* [4] for finite queues and earlier by Theimer *et al* [14] and Hsiao *et al* [5] for single-slot queues (single-buffered networks).

In the simplest of these analyses, Lin and Kleinrock's [11], the queues are modeled as by Yoon [15], however an effective service rate is used in place of the service rate computed as in [15]. This service rate is derived by considering two cases: the probability of service for a packet first arriving at the head slot and the probability of service for a packet that was blocked by a full queue in the previous cycle. For a packet that first arrives, the stationary next-stage queue-full probability is used in computing the service rate. For a packet which had been blocked, the knowledge that the queue was full is used in computing the service rate. The two service rates are combined to obtain

* This work is supported in part by the Louisiana Board of Regents through the Louisiana Education Quality Support Fund, contract number LEQSF (1993-95)-RD-A-07 and by the National Science Foundation under Grant No. MIP-9410435.

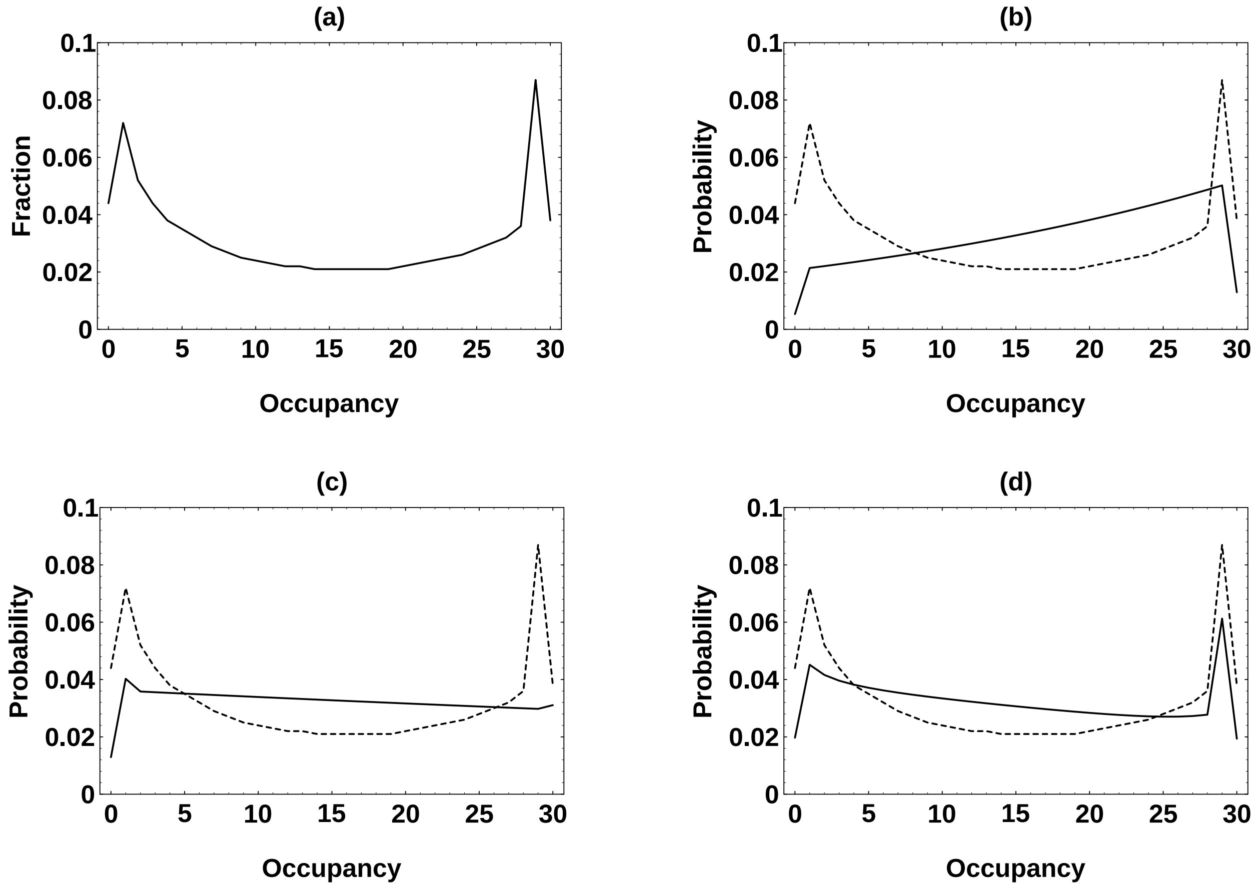

Figure 1. Queue occupancy (number of packets in queue) distributions in the second stage of an 8-stage, 30-slot-buffer banyan network obtained from *(a)* simulation, and simulation (dashed) plotted with results obtained *(b)* using Yoon's analysis, *(c)* using Mun and Youn's analysis, and *(d)* using the analysis described here.

the effective service rate. This analysis is for networks using nonblocking crossbars with queues at the module outputs: any number of packets entering a module can enter a queue if there is space. (The analysis presented here is for blocking crossbars.) This analysis does account for the effect of a full queue on performance, but it does so by adjusting the overall service rate. Unlike the model described here, it does not model the higher arrival rate when a queue is full, which has a strong effect on the queue state distribution.

The analyses presented by Mun and Youn [12], Theimer *et al* [14], and Hsiao *et al* [4] model each queue with several sets of states. Each set corresponds to a possible fate of the head packet in the previous cycle. A different service rate is found for each set, thus modeling, for example, the lower service probability of a packet that had been blocked. The results reported by these investigators come closer to simulated results than those reported by Yoon *et al* [15].

The models reported in [14] and [5] are for single-slot queues, so do not apply to the networks considered here. In [4] the model is incompletely reported. The model described by Mun and Youn does not always give throughput values as close to simulation as the model described here. Further, Mun and Youn's model does not predict queue-occupancy distributions as closely.

The analysis described here is designed to capture congestion's salient features, based upon simulations of congested networks. The central feature of the model is the congested queue.

A queue becomes congested in a cycle in which it is full and a packet destined for the queue is blocked. The arrival rate to a congested queue will be higher than normal because of the packets it blocks; the higher arrival rate tends to prolong congestion. Congestion ends when the queue has one slot free and no packet is ready to move into the queue.

The effect of congestion can be seen in the queue state distribution plot for a simulated banyan network appearing in Figure 1*(a)*. The plot shows the fraction of time a second-stage queue spends at each level of occupancy, from 0 packets (empty), to 30 packets (full for this network). Because of the congestion effect there is a clear peak at 29 packets. (The peak appears at 29, not 30, packets because a packet will not enter a queue that was full at the beginning of a cycle.) The throughput at and near congestion is determined by this part of the queue state distribution, so modeling it accurately is important.

The queue state distribution for the same network, obtained using the analysis method of Yoon *et al* (adapted for local flow control) appear in Figure 1*(b)*. The distribution does not have the form of Figure 1*(a)*, although the maximum does occur at 29 packets. Note that the probability that a queue holds 29 packets is lower than the corresponding quantity observed in simulation.

The queue state distribution for the same network obtained using the analysis method of Mun and Youn (adapted for local flow control) appear in Figure 1*(c)*. Again, the form does not match that observed in simulation. The full-queue probabilities are higher than Yoon's model predicts; a possible reason for its greater accuracy.

The state distribution obtained through the analysis described here is plotted in Figure 1*(d)*. There is a peak at 29, as observed in simulations. The form of the state distribution appears more like that obtained from simulations than the distributions obtained using the methods of Yoon *et al* and Mun and Youn. Further, the throughput predictions are more accurate than the previous analyses for many configurations.

The analysis described here also accounts for a congested queue's effect on its *dual*. (A queue's dual [defined for all but the first stage] in a network using 2×2 crossbars is the queue in the same stage connected to the same previous-stage crossbar.) The arrival rate to a congested queue's dual is lower (as is the service rate for queues feeding the congested queue).

The remainder of the paper is organized as follows. In Section 2 congestion is defined; in Section 3 the highlights of the model are described. Results appear in Section 4; conclusions appear in Section 5.

2 Congestion Model

The analysis presented here differs from previous analyses in that the arrival rate to a queue depends upon the state of the queue, most importantly on whether the queue is congested. Under the model a queue can change from a full state to what is called a congested state, in which the arrival rate is much higher. The higher arrival rate is due to the accumulation of packets in the previous-stage queues bound for the congested queue. The previous-stage queues will be refered to as *feeder queues*. This situation is illustrated in Figure 2 and formally defined below.

Definition 1: *A queue in stage $0 < j < n$ of an n-stage banyan network using d-slot queues is said to be in the* congestion-start condition *if it contains d packets and the head slot of at least one feeder queue contains a packet which is to pass through the queue.*

Definition 2: *A queue in stage $0 < j < n$ of an n-stage banyan network using d-slot queues is said to be in the* congestion-end condition *if it contains $d-1$ packets and no feeder-queue head slot has a packet which is to pass through the queue.*

Definition 3: *A queue in stage $0 < j < n$ of an n-stage banyan network using d-slot queues is said to be* congested *in cycle t if there exists a cycle $t_s < t$ such that the queue was at the congestion-start condition in cycle t_s and for all $t_s < \tau < t$ the queue was not in the congestion-end condition.*

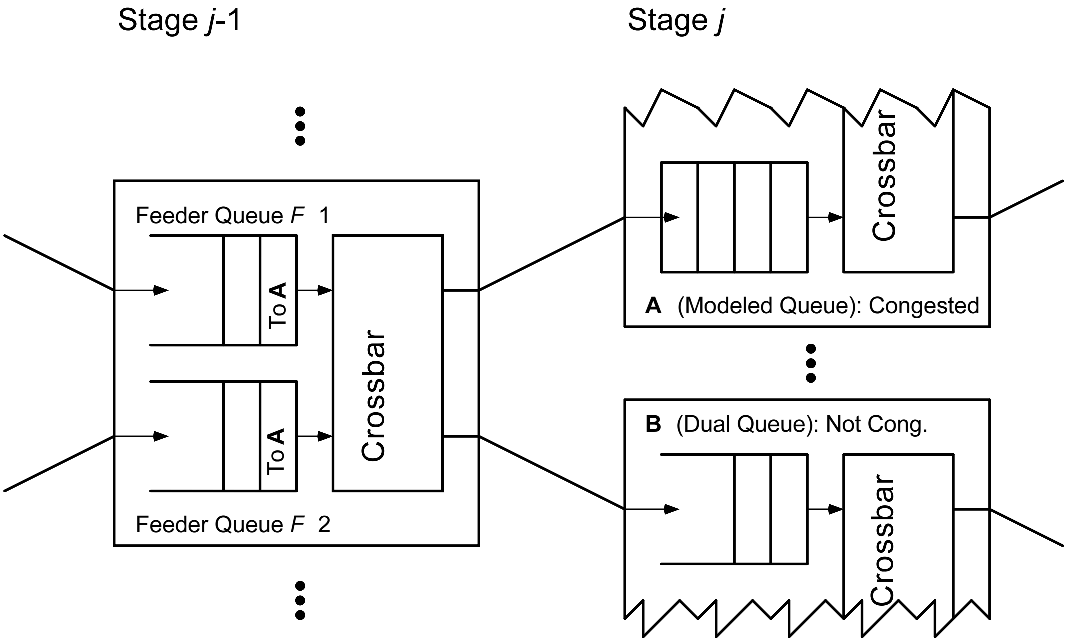

Figure 2. Illustration of congestion. Queue **A**, being congested, is full much of the time. This causes packets bound for **A** to accumulate at the head of the feeder queues, $F1$ and $F2$, prolonging congestion. Queue **B** is not congested

An example of a queue going from an uncongested to a congested condition, and then back to an uncongested condition, appears in the table below. The table shows the history of four queues connected to a common crossbar, as illustrated in Figure 2. Queue **A** suffers congestion. Queues $F1$ and $F2$ are queue **A**'s feeder queues, queue **B** is queue **A**'s dual queue. The condition of each queue over a time interval is shown in the table. For queues $F1$ and $F2$ the next queue on the path of the head packet is shown, **A** or **B**, or **E** if the queue is empty. For queues **A** and **B** either the number of packets in the queue is shown, or if **A** is congested the entry indicates whether **A** has zero or one slots free, indicated by c_1 or c_0, respectively. An asterisk next to the entry indicates that the head packet of the respective queue will move to the next stage at the end of the cycle. Queue **A** is full but not congested up to cycle 2. At cycle 2 $F1$ offers a packet which **A** cannot accept so that congestion starts. Congestion continues until the end of cycle 8, in which there is one slot free in **A** while no packet is offered.

Table 1: A Congestion Example

Queues	Feeder		Fed		Queues	Feeder		Fed	
Cycle	$F1$	$F2$	**A**	**B**	Cycle	$F1$	$F2$	**A**	**B**
0	**A**$*$	**E**	$d-1$	0	5	**A**	**A**$*$	c_1*	$1*$
1	**E**	**B**$*$	d	0	6	**B**$*$	**A**$*$	c_1*	0
2	**A**	**E**	$d*$	$1*$	7	**B**$*$	**E**	c_0*	$1*$
3	**A**$*$	**B**$*$	c_1	0	8	**E**	**B**$*$	c_1	$1*$
4	**A**$*$	**A**	c_0*	1	9	**A**$*$	**E**	$d-1$	1

3 Analysis Overview

The following is an outline of the analysis method; a complete description can be found in

[9]. The network to be modeled is an n-stage input-buffered banyan network [3] with queues having d slots each. Local flow control is used; that is, a packet cannot enter a queue in a cycle unless the queue has a free slot. The slot occupied by a packet leaving a queue in a cycle is considered free in the next cycle. Offered traffic consists of fixed-length packets; each queue slot can hold exactly one packet. Packets arriving at an input that is blocked, due to a full queue, are dropped. The number of arrivals at an input in a cycle is modeled by a Bernoulli random variable. These n random variables are independent and identically distributed. Destinations are randomly chosen for each packet so that they are uniformly distributed over network outputs and independent of other destination choices [6,15].

A network is modeled by n independent Markov chains, one for each stage. The state model encodes information about a queue and its neighbors. The neighbors considered are the previous-stage queues which feed the queue, the next-stage queues to which the queue is connected (considered as a unit) and the dual queue in the same stage. The state of a queue (in all but the first and last stages) encodes whether or not the dual and next-stage queues are congested. If a queue is congested then its state also encodes the destinations of any packets in the heads of the previous-stage queues.

The part of the state coding previous-stage queues is used in modeling congestion. The part coding next-stage queues captures the propagation of congestion towards the inputs. The part coding the dual queue captures the lower arrival rate when the dual is congested. (A simpler model which gives similar results omits the state of the dual queue. That model is not described here.)

A queue's state is specified with three-tuple (S, D, N). State variable D describes the dual queue, N describes the next-stage queue, and S describes the queue itself. The queue to which a state (*e.g.,* (S, D, N)) corresponds will sometimes be called the *modeled queue*. Thus, S describes the modeled and previous-stage queues while D describes the dual queue. State transitions have three components, corresponding to the elements of the three-tuple. Although every transition includes each of the components, for clarity the components will be discussed separately, as though they were transitions themselves.

3.1 The Modeled Queue

A queue is either congested or not congested; if not congested S is the number of packets in the modeled queue, $S \in \{0, 1, \ldots, d\}$. If congested $S \in \mathcal{Q}_{\mathsf{C}}$, where $\mathcal{Q}_{\mathsf{C}}$ is the set of possible previous-stage *HOL system* states. (HOL system refers to the contents of the head slots of the feeder queues [6].) To reduce the number of states needed symmetric HOL configurations are not duplicated and the number of packets in a queue during congestion is not explicitly coded. This complicates the analysis but also reduces the amount of computation. See [9] for details.

The possible values of S when a queue is congested are

$$\mathcal{Q}_{\mathsf{C}} = \{\, \mathbf{AA}, \mathbf{AB}, \mathbf{AE}, \mathbf{BB}, \mathbf{BE}, \mathbf{EE} \,\}.$$

Symbol $\mathbf{A}$ indicates that a HOL slot holds a packet bound for the modeled queue, $\mathbf{B}$ indicates that a HOL slot holds a packet bound for the dual queue, and $\mathbf{E}$ indicates that a HOL slot is empty. (The identity of the HOL slot is not coded, so there is no need for states such as $\mathbf{BA}$.) Transitions from and among these states are based on the service rate of the modeled queue, and the arrival rate and state distribution of the previous-stage queues.

Consider a transition between states in which S changes from $\mathbf{AE}$ to $\mathbf{EE}$. Three events are implied by this transition: 1) the packet bound for the modeled queue was given service; 2) the packet bound for the modeled queue was not replaced; and 3) there was no arrival in the queue that was empty before the transition. The probability of this transition is computed using the probability that there is space in the modeled queue, the probability of no arrival to an empty queue in the previous stage, and the probability that the previous-stage queue is empty given that a packet left the queue in the previous cycle. Each of these can be computed from the state distribution. Since the congestion status of the next-stage queue is included in the state these probabilities can be computed accurately. For example, the probability that the previous-stage queue is empty after packet departure is computed using a state distribution conditioned on the next-stage queue being congested. Therefore the probability will be lower than it would be if next-stage congestion were not taken into account. See [9] for details.

Consider a transition from **BE** to **EE**. This transition is similar to the previous example with one important difference: the packet is bound for the dual queue. When the dual queue is not congested the probability of space is much higher. Because of this imbalance (under stationary conditions), states with packets bound for the modeled queue will have a higher probability.

Transitions from a congested state to an uncongested state occur when $S \in \{\, \mathbf{BB}, \mathbf{BE}, \mathbf{EE} \,\}$ and if the modeled queue has one empty slot. Since the probabilities of these states computed using the model are lower, especially under heavier traffic, the self-perpetuating nature of congestion is captured. The states following congestion are $S \in \{\, d-2, d-1 \,\}$. Transitions to a congested state occur when $S = d$ (the queue is full) and if there is an arrival. See [9] for details.

The precision obtained by modeling the HOL system during congestion is not needed when a queue is not congested. Therefore for non-congested states S only encodes the number of packets in the queue (rather than encoding a HOL state distribution for each possible number of packets in the queue). Transition probabilities between these states are computed from arrival and service rates in the usual way [15].

3.2 The Dual and Next-Stage Queues

From the perspective of the modeled queue, the dual queue is either congested, labeled **C**; or not congested, labeled **N**. Thus, $D \in \{\mathbf{N}, \mathbf{C}\}$. Because a congested queue will block packets bound for the dual queue the arrival rate to the dual queue will be lower when $D = \mathbf{C}$ than when $D = \mathbf{N}$. When $N = \mathbf{C}$ at least one of the two next-stage queues are congested; when $N = \mathbf{N}$ neither next-stage queue is congested. If a next-stage queue is congested the service rate is reduced.

As an illustration of this notation, the state of a queue in an idle network is $(0, \mathbf{N}, \mathbf{N})$. State $(5, \mathbf{N}, \mathbf{C})$ indicates a queue holding five packets with at least one of the next-stage queues (to which it connects) in the congested state, while the queue's dual is not congested. State $(\mathbf{AA}, \mathbf{C}, \mathbf{N})$ indicates that the queue and its dual are congested, while the next-stage queues to which it connects are not congested.

A queue undergoes *lateral transitions* when its dual or next-stage queue enters or leaves congestion. For example, $(3, \mathbf{N}, \mathbf{C}) \rightarrow (4, \mathbf{C}, \mathbf{C})$ is a dual-queue lateral transition while $(3, \mathbf{N}, \mathbf{C}) \rightarrow (4, \mathbf{N}, \mathbf{C})$ is not. The dual-queue lateral-transition probabilities are based on the probability that the dual queue starts or ends congestion. Similarly, the next-stage-queue lateral-transition probabilities are based on the probability that one or both of the next-stage queues starts or ends congestion.

The lateral transition probabilities have been derived so that stationary state distributions computed are self-consistent with respect to the amount of time a queue is congested. That is, if the state distribution for a queue indicates that the next-stage queues are congested $x\%$ of the time, then the next-stage queue's state distribution will indicate that a queue or its dual are congested $x\%$ of the time. Achieving this self-consistency complicates the lateral-transition probability expressions. See [9] for details.

3.3 Service Rate

Two service rates are used, one used when there is next-stage congestion and one used when there is not. The service rate is computed so that the number of packets leaving a queue in one stage is equal to the number received in the next stage in a corresponding set of states.

This is done by, in effect, computing the number of packets the next stage expects to receive while in certain states and then setting the service rate so that number of packets is sent during the corresponding states. The service rate computation was chosen so that under stationary conditions the flow rates in all parts of the network are identical. See [9] for details.

3.4 HOL System and Arrival Rates

The part of a queue state modeling the congested HOL system is a central feature of the analysis method. This HOL system serves a secondary function: it is used for computing an arrival rate to the dual queue. A synthetic HOL system is also used for computing an arrival rate for use by an uncongested queue when its dual is also uncongested. The synthetic HOL distribution is determined from queue-occupancy probabilities.

The HOL distributions are used to compute two probabilities, r_{an} and r_{na}. Symbol r_{an} denotes the probability that a packet in a HOL system will be bound for a queue given that there was a packet bound for the queue in the previous cycle. Similarly, r_{na} denotes the probability that no packet in the HOL system will be bound for a queue given that there was a packet bound for the queue in the previous cycle. Note that because an arriving packet may block another packet bound for the same queue, $r_{\mathrm{an}} \geq r_{\mathrm{na}}$.

In general, given only the state of a queue one cannot determine whether there was an arrival in the previous cycle. However, the probability of an arrival in the previous cycle can be determined; further, it is not the same for every state. A modified transition matrix is used to determine the probability that in a given state there was an arrival in the previous cycle. The modified transition matrix is constructed from the regular transition matrix by zeroing all transitions which do not correspond to arrivals.

The previous-cycle-arrival probabilities, along with r_{an} and r_{na} are used to determine arrival rates for use in certain states. Separate arrival rates are computed for empty, full, and one-slot-free queue states because, as observed in simulations, the arrival rates at these states differ from the average. Accurate modeling of these conditions is important because throughput is determined by the amount of time a queue is empty or full.

For example, consider an empty queue. When a queue is empty it is certain that there was no arrival in the previous cycle. Thus an empty queue has a lower arrival rate. The probability that a full queue had an arrival in the previous cycle is higher than average. (If the queue had no arrival, it would also have to have had no service. The probability of the two is smaller than the sum of the alternatives.) Therefore, the probability of an arrival in the full state is higher. Because of the way in which congestion is modeled, the arrival rate for a queue having one slot free also differs from average. See [9] for details.

4 Verification

The analysis was tested by comparing its predictions to the output of a simulator. The analysis and simulator were used on a variety of network and traffic configurations, as were the analysis methods of Yoon *et al* [15] and Mun and Youn [12,16] for comparison. Throughput and queue-occupancy distributions are compared.

4.1 Methodology

A simulator was used to determine the performance of the network. The simulator precisely implements the modeled network and traffic. Simulations were performed for 40 000 cycles; confidence intervals were computed for severe parameter sets (large queue and network sizes and heavy traffic). The confidence intervals were much smaller than the differences with the analytical models. For example, the 95% confidence interval for the throughput of 8-stage, 50-slot-queue networks at $\lambda = .90$ based on three runs was $[.7186, .7190]$, where λ indicates the offered-traffic arrival rate. Simulator output included throughput and queue-occupancy distributions.

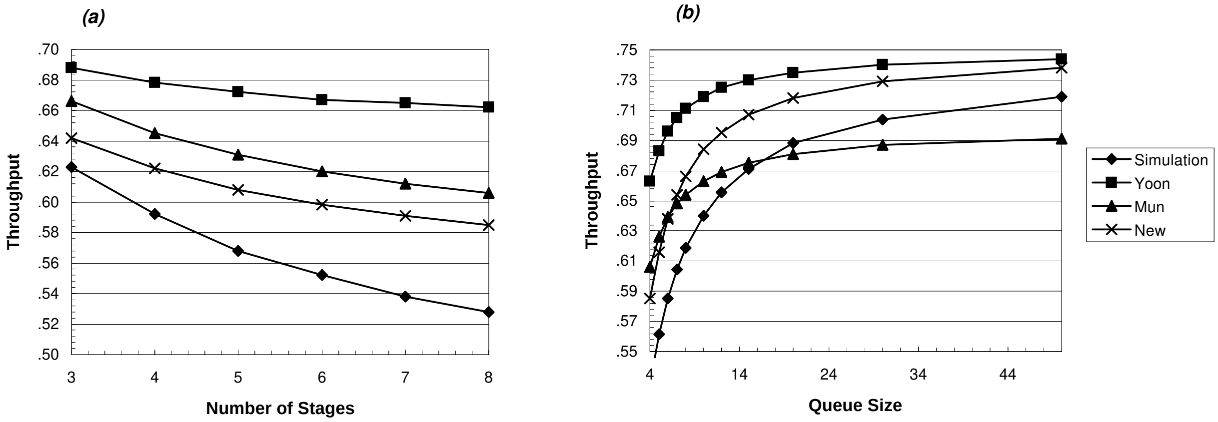

Figure 3. Throughput v. *(a)* number of stages for 4-slot queue and *(b)* queue size for 8-stage networks.

The analyses were used to determine the networks' throughput and queue occupancy distributions. The analysis methods of Yoon *et al* [15] and Mun and Youn [12,16] were adapted to use local flow control, to be comparable to the analysis presented here. (See [2,16] for a description of these variations.) Each was run for enough cycles to obtain results sufficiently close to stationary.

4.2 Comparisons

At low arrival rates all analyses predicted the simulated network throughput closely; those results are not shown here. At higher arrival rates most analyses overestimated throughput. The effect of network size can be seen in Figure 3*(a)* where throughput is plotted for n-stage, 4-slot-queue networks for $3 \leq n \leq 8$ with $\lambda = .9$. The analysis described here computes the throughput most closely matching simulated throughput. Mun and Youn's analysis also performs substantially better than that of Yoon *et al.* In Figure 3*(b)* throughput is plotted for 8-stage, d-slot networks for $4 \leq d \leq 50$ with $\lambda = .9$. The analysis described here computes throughput closest to simulated values for small and large queue sizes. Mun and Youn's analysis performed better for moderate queue sizes. In all cases Yoon's analysis overestimates throughput.

A goal of the work and basis of the analysis reported here is to accurately model the queue occupancy distribution during congestion. The analysis comes much closer than any of the other analyses to doing so. In Figure 1 appear the occupancy distributions obtained from simulation and the three analyses for the second stage of an 8-stage, 30-slot network with $\lambda = .9$. The analysis was motivated by the observation that the state distribution of queues in simulated networks have peaks at 1 and $d - 1$ packets. As described in the introduction, the analysis described here comes much closer to matching these peaks.

5 Conclusions

A banyan network analysis method designed to accurately model congestion has been presented. A congested state for a queue is defined and incorporated in the queue's state model. The state model encodes the state of the queue as well as its neighbors. This allows the effect of congestion on the queue's neighbors to be modeled. State-dependent arrival and service rates are used. Comparisons to simulation and other analyses show the model can predict throughput more closely than other models. Further, it models queue occupancy distribution much more closely than other models.

Two other models based on congestion have been developed by the author, both models use fewer states. One model is similar to the one reported here, except that the state contains no information about the dual queue, resulting in about half the number of states. In fact, the predictions of that model are close to those described here. Since the model described here is a superset of the simpler model and since the model described here might lead to a more accurate model, the simpler model was not described despite its greater efficiency. The other model based on congestion is much simpler: the queue state contains no information about the dual or next-stage queues. An advantage of the model is that a queue's stationary distribution can be found in closed form [8]. (Iteration is still required to compute expected congestion duration and to solve the entire network.) This analysis does not provide as accurate predictions as the analysis described here, but it is computationally more efficient, especially for large queue sizes.

Although the predictions of the analysis described here are closer than those reported by others, there is still room for improvement. Examination of detailed simulator and analysis output reveals areas in which the simulation and analysis diverge. The traffic flow during congestion is higher in analysis than in simulation. This may be caused by insufficient correlation of congestion in adjacent stages. A revised model could increase correlation by having a next-stage-recently-congested state. A recently congested queue would likely soon be congested again. Another difference between analysis and simulation is the amount of time that a queue and its dual are simultaneously congested; it is greater in analysis. Further work then might be undertaken on a more accurate model of HOL/queue interaction during congestion.

6 References

[1] D. M. Dias and J. R. Jump, "Analysis and simulation of buffered delta networks," *IEEE Transactions on Computers*, vol. 30, no. 4, pp. 273–282, April 1981.

[2] J. Ding and L.N. Bhuyan, "Finite buffer analysis of multistage interconnection networks," *IEEE Transactions on Computers,* vol. 43, no. 2, pp. 243-247, February 1994.

[3] L. R. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," in *Proceedings of the International Symposium on Computer Architecture,* 1973, pp. 21–28.

[4] S. H. Hsiao, C. Y. R. Chen, K. C. Nwosu, and D. Meliksetian, "Performance analysis of finite-buffered multistage interconnection networks," IEEE International Conference on Communications, pp. 53–57, 1993.

[5] S. H. Hsiao and C. Y. R. Chen, "Performance analysis of single-buffered multistage interconnection networks," *IEEE Transactions on Communications,* vol. 42, no. 9, pp. 2722-2729, September 1994.

[6] J.Y. Hui, "Switching and traffic theory for integrated broadband networks," Boston: Kluwer Academic Publishers, 1990.

[7] Y. C. Jenq, "Performance analysis of a packet switch based on single–buffered banyan network," *IEEE Journal on Selected Areas in Communications,* vol. 1, no. 6, pp. 1014–1021, June 1983.

[8] D. M. Koppelman, "Sticky states in banyan network queues and their application to analysis," submitted to the *Journal of Parallel and Distributed Computing.*

[9] D. M. Koppelman, "Congested banyan network analysis using congested-queue states and neighboring-queue effects," Tech. Report LSU-ECE-95-163, Louisiana State University, Baton Rouge, LA, September 1995. Available via ftp://gate.ee.lsu.edu/pub/koppel/pubs/sticky.ps.gz

[10] D. H. Lawrie, "Access and alignment in an array processor," *IEEE Transactions on Computers,* vol. 24, no. 12, pp. 1145–1155, December 1975.

[11] T. Lin and L. Kleinrock, "Performance analysis of finite–buffered multistage interconnection networks with a general traffic pattern," ACM SIGMETRICS, pp. 68–78, May 1991.

[12] Y. Mun and H.Y. Youn, "Performance analysis of finite buffered multistage interconnection networks," *IEEE Transactions on Computers,* vol. 43, no. 2, pp. 153-162, February 1994.

[13] M.C. Pease, III, "The indirect binary n-cube microprocessor array," *IEEE Transactions on Computers,* vol. 26, no. 5, pp. 458-473, May 1977.

[14] T. H. Theimer, E. P. Rathgeb, and M. N. Huber, "Performance analysis of buffered banyan networks," *IEEE Transactions on Communications,* vol. 39, no. 2, pp. 269–277, February 1991.

[15] H. Yoon, K. Y. Lee, and M. T. Liu, "Performance analysis of multibuffered packet-switching networks in multiprocessor systems," *IEEE Transactions on Computers,* vol. 39, no. 3, pp. 319–327, March 1990.

[16] H. Y. Youn and Y. Mun, "On multistage interconnection networks with small clock cycles," *IEEE Transactions on Parallel and Distributed Systems,* vol. 6, no. 1, pp. 86-92, January 1995.