Sticky States in Banyan Network Queues and Their Application to Analysis *

DAVID M. KOPPELMAN **

Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA

ABSTRACT

A finite-buffered banyan network analysis technique designed to model networks at high traffic loads is presented. The technique specially models two states of the network queues: queue empty and queue congested (roughly, zero or one slots free). A congested queue tends to stay congested because packets bound for the queue accumulate in the previous stage. The expected duration of this state is computed using a probabilistic model of a switching module feeding the congested queue. A technique for finding a lower arrival rate to an empty queue is also described. The queues themselves are modeled using independent Markov chains with an additional congested state added. The new analysis technique is novel in its modeling the higher arrival rate to a congested queue and a lower arrival rate to an empty queue. Comparison of queue state distributions obtained with the analysis and simulation show that an important feature of congestion is modeled.

1 INTRODUCTION

A novel model of finite-buffered banyan networks, (multistage networks having exactly one path between each input/output pair), is presented along with an improved banyan analysis method based on the model. The distinguishing feature of the model is that the arrival rate of traffic to a queue (buffer) in the network depends upon the state of the queue itself. In particular, the arrival rate to an empty queue is lower than average while the arrival rate to an almost full queue can be much higher than average. These arrival rates tend to prolong the respective queue conditions, making *sticky* the queue empty state and what will be called the congested state. Since traffic flow in such a network depends strongly on the fraction of time a queue is empty and full, the model captures an important aspect of banyan network behavior.

Many of the banyan network analysis methods appearing in the literature are based on Dias and Jump [1] and Jenq's [6] analyses of single-buffered networks. Their models are based on the observation that traffic arriving at the queues in a stage would be identically distributed and the assumption that the arrivals at a queue are independent of the state of any queue. Using symmetry properties the observation is easily proven; the assumption, while clearly a simplification, yields an easily analyzed model. Each stage is modeled by a single queue. The queue's stationary state distribution can be found in closed form although analysis of the entire network uses iteration. Similar models for finite-buffered banyan networks were presented by Yoon, *et al* [17] and Szymanski and Shaikh [15] and more recently by Ding and Bhuyan [2]. The stationary distribution for the queues in these models can also be found in closed form.

Performance predictions made with these models match simulations closely when the network is offered light to moderate traffic; the predictions diverge, however, as the offered traffic rate increases. The performance of many systems using banyan networks, including parallel computers,

* To appear in *The Journal of Parallel and Distributed Computing*.

** 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.

is determined by the performance of the network at high traffic rates, therefore models of banyan performance which are more accurate at high traffic rates are needed.

One approach to handling heavy traffic is to modify these models so that the service rate of a queue is dependent upon the fate of the head packet (next item in the queue to be served) in the previous cycle. The rationale is that a head packet having been blocked will more likely be blocked again in the next cycle. Such approaches have been tried by Lin and Kleinrock [11], Mun and Youn [12], and Hsiao, *et al* [4] for finite queues and earlier by Theimer, *et al* [16] for the single-buffered networks.

In the simplest of these analyses, Lin and Kleinrock's [11], the queue itself is modeled as by Yoon [17], however an effective service rate is used in place of the service rate computed as in [17]. 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 which was blocked by a full queue at the previous cycle. For the packet which first arrived at the head slot, the stationary queue full probability is used in computing the service rate. For the 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 the effective service rate. This analysis is for output buffered networks, those using nonblocking crossbars with queues at the module outputs where any number of packets entering a module can enter a queue as long as there is space. (The analysis presented here is for input-buffered networks networks which use blocking crossbars.) Lin and Kleinrock's analysis does account for the effect of a full queue on performance, but does so by adjusting the overall service rate. Unlike the analysis 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* [16], and Hsiao, *et al* [4] model each queue with several sets of states. Each set corresponds to the 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. These approaches yield better results, but there are shortcomings. Theimer, *et al* [16] analyzes only single-buffered networks. The results presented in [4] lacked detail, probably because of space restrictions. As will be illustrated below, the queue state distributions obtained using Mun's analysis do not match those obtained from simulation as well as the analysis presented here, especially under light traffic, and so may be less useful in understanding congestion. Mun's analysis can be computationally expensive for large queues since the stationary distribution of the queues cannot, or at least not easily, be found in closed form.

The model presented here differs in several important respects: it is the arrival rate, rather than the service rate which is state dependent; the dependence is on the state of the queue *receiving* the arrivals. This produces a distinctive queue state distribution, similar to those observed in simulations (see the analysis results section below). The arrival rates themselves are computed taking into account the queues receiving the arrivals and the queues feeding them, capturing the effect of a congested queue on the queues feeding it. A closed-form expression gives the stationary distribution of a queue, although iteration is still required to completely analyze the network.

Throughput obtained using the analysis is best for small queue sizes: the results are comparable to those obtained using Mun's analysis. For intermediate queue sizes Mun's analysis yields closer results, but for larger queue sizes, the analysis presented here gives closer and faster results.

The queue state distributions obtained match those obtained from simulations for high queue occupancies, indicating that the congestion effect is accurately modeled. The queue empty state predicted by the model is closer to simulation than Yoon's analysis for all stages and Mun's analysis for stages near the output.

In a related work a queue is modeled using sticky states as well as the state of queues in the next and same stage, quadrupling the number of states [9]. While the predicted throughput is closer than that obtained here, the added complexity is not necessary to model the sticky states induced by congestion.

2 Preliminaries

The network being modeled, which will be called the (n, m, d) *basic network,* is an n-stage banyan network consisting of $m \times m$ modules each having m d-slot queues. The stages are numbered from 0 to $n-1$, with 0 being the first stage. Each stage consists of m^{n-1} modules. The network has m^n inputs which are connected to the first-stage module inputs by links; for all but the last stage, module outputs are connected by links to module inputs in the next stage. Module outputs in the last stage are connected to network outputs. For a banyan network the links can be connected in any way as long as there is exactly one path between every input/output pair [3].

Each module consists of m queues, each with d slots connected to an $m \times m$ crossbar switch. The module inputs connect to the queues; the queue outputs connect to the crossbar; and the crossbar outputs connect to the module outputs. A *packet* is the unit of communication; it consists of a destination (a network output) and data. Each slot can hold one packet (all packets are the same size). The queue uses a first in, first out service discipline; the slot containing, or that would contain, the packet being served is called the *head slot*, the packet is called the *head packet.*

Earlier analyses and the analysis presented here apply to networks having symmetry properties that are easily proven for a subclass of banyan networks called *bidelta* networks [10]. A network is a *delta* network if the module output needed by a packet is uniquely determined only by the stage number and the packet's destination [13]; it is a bidelta network if a similar property holds when routing from outputs to inputs. (For example, in an n-stage omega network using 2×2 modules a packet in stage j uses module output 0 if digit $n-j-1$ of its destination is 0 and output 1 otherwise.) Most banyans of interest are bidelta networks, and the symmetry properties likely hold for many others. See [14] for an elementary introduction and [10] a discussion of the topology of banyan networks.

The network operates in a synchronous clocked fashion; time is divided into *cycles.* In cycle t a packet can move from a queue in one stage to a queue in the next stage if 1) it is at the head of the queue, 2) it is granted use of the appropriate link, and 3) the queue in the next stage has at least one free slot during the cycle. The packet which is granted use of a link is chosen randomly from those contending for its use. The packet meeting these conditions at cycle t will be in the next stage queue starting at cycle $t+1$. A packet at a network input will move into the network in cycle t if there is at least one slot free in the corresponding queue in the first stage. A packet will move from a queue in the last stage to a network output if it is at the head of the queue and is granted the appropriate output.

In the remainder of this paper *Yoon's analysis method* will be used to refer to an analysis which is similar to that of Yoon, Lee, and Liu [17], except that packets move as described above, as later done by Ding [2].

The number of arrivals at network inputs in a cycle is modeled by independent, identically distributed Bernoulli random variables. The *arrival rate* is the expected number of arrivals at an input per cycle. (Equivalently, the probability of one arrival at an input in a cycle.) An arriving packet which does not enter the network is dropped. Each packet arriving at the network has a fixed destination, randomly chosen such that all outputs are equally likely and destination choices

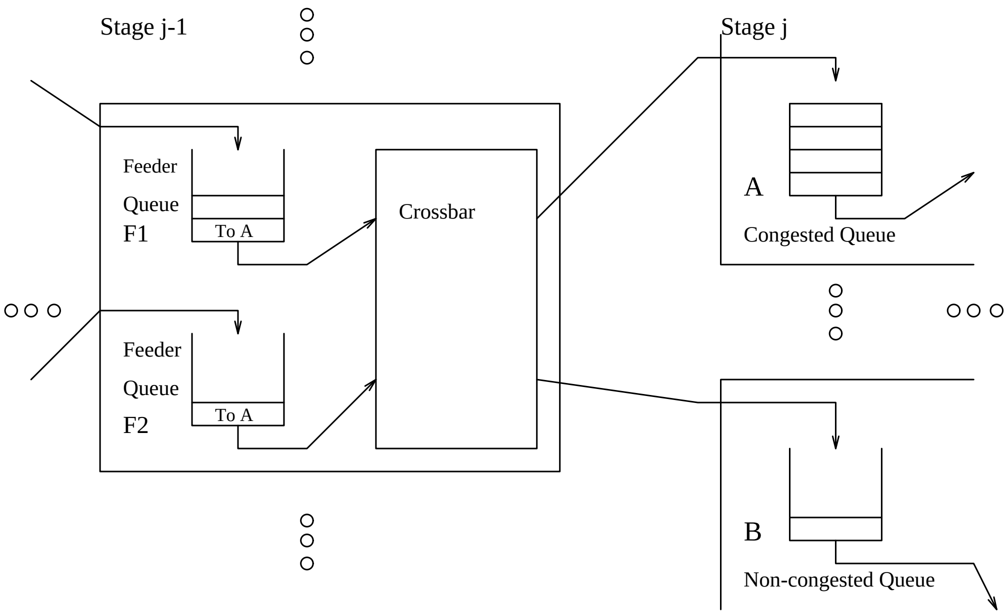

FIG. 1. Illustration of congestion. Queue A, being congested, is full much of the time. This causes packets bound for A to accumulate at the head of the feeder queues, F1 and F2, prolonging congestion. Queue B is not congested

are independent and identically distributed. The *latency* of a network is defined to be the mean or expected number of cycles packets spend in the network. The *throughput* of the network is the mean or expected number of packets leaving the network per output per cycle.

3 Sticky State Model

3.1 Congestion

Under the sticky model a queue can change from a full state to a congested state, in which the arrival rate is much higher due to the accumulation of packets bound for the congested queue. The congested state is self-prolonging or *sticky* because the higher arrival rate is both a consequence of and tends to prolong the state. This situation is illustrated in Figure 1 and defined below.

Definition 1: *A queue in stage $0 < j < n$ of an (n, m, d) basic network is said to be in the* congestion start condition *if it contains d items and the queue head slot of at least one of the queues in the previous stage contains a packet which is to pass through the queue.*

Definition 2: *A queue in stage $0 < j < n$ of an (n, m, d) basic network is said to be in the* congestion end condition *if it contains $d - 1$ items and no queue head slot in the previous stage has a packet which is to pass through the queue.*

Definition 3: *A queue in stage $0 < j < n$ of an (n, m, d) basic network 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.*

An example of a queue going from an uncongested state to a congested condition, and then back to an uncongested condition is presented in the table below.

TABLE I
A Congestion Example

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

Table 1 shows the history of four queues connected to a common crossbar, as illustrated in Figure 1. Queues $F1$ and $F2$ are connected to the crossbar's inputs; queues A and B are connected to the crossbar's outputs. 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 items in the queue is shown, or if A is congested the entry indicates whether A has zero or one slots free, indicated by c_0 or c_1, 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 suffers congestion due to feeder queues $F1$ and $F2$. 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 but no packet is offered.

3.2 The Module Model

An important quantity in the sticky analysis is the expected duration of congestion. The computation of the expected duration of congestion uses a state model, called the module chain, modeling the state of the previous stage module which is connected to the congested queue. The queues in the previous stage module will be referred to as *feeding queues.* The important information captured in this chain is the destination of the head packets in the module.

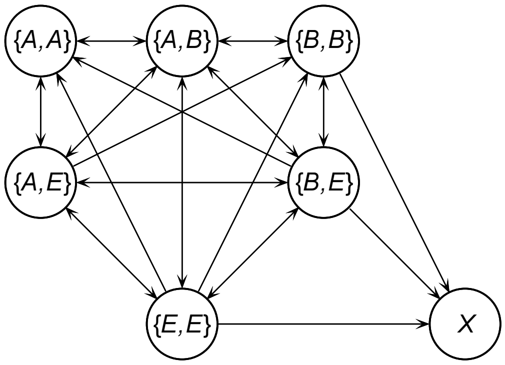

FIG. 2. State transition diagram for a module chain in a network using 2×2 modules. The state X indicates no congestion. The other state labels indicate status of the head slot in each feeder queue, A for a packet bound to the congested queue, B for packet bound to another queue, and E for head slot empty. Transition probabilities are presented in the text

A feeding queue will be modeled as having either zero or at least one packet; if it has at least one packet then it will also have one of two types of destination: one requiring passage through the congested queue and one passing through any other queue. These feeding queue states will be labeled E, A, and B, respectively.

Let the state of the module chain in stage $j-1$ at cycle t be denoted $M_{j-1,t}$ for $1 \leq j < n$. A module state is labeled with multisets (a set which can have more than one copy of an element) of queue states as described above, plus an absorbing state indicating that there is no congestion. (A state is absorbing if the probability of leaving the state is zero.) That is,

$$M_{j-1,t} \in X \cup \{\{q_1, q_2, \ldots, q_m\} \mid q_1 \in \{A, B, E\}, q_2 \in \{A, B, E\}, \ldots, q_m \in \{A, B, E\}\},$$

where X is the absorbing state label. The absorbing state is entered if none of the feeding queues are in state A and the congested queue is not full. For example $M_{j-1,t} = \{A, B\}$ indicates that the module in stage $j-1$ at cycle t has one queue in which the head packet is bound for the congested queue and one queue in which the head packet is bound for the non-congested queue. The module state transition diagram for 2×2 modules appears in Figure 2.

Note that, because a multiset is used, the identity of the queues is not represented. This allows certain states to represent several possible queue state arrangements. For example, a module in an $(n, 2, d)$ network in which one feeding queue's head packet was bound for the congested queue and one was not would be in state $\{A, B\}$ regardless of which queue had the packet bound to the congested queue.

Since each module state for an $(n, 2, d)$ network is a combination of two feeding queue states the transition probabilities must take into account changes in both queues. For example, the change from state $\{A, E\}$ to $\{B, E\}$ could occur because the queue in state A changed to state B or because the queue in state A changed to E and the queue in state E changed to B. When both queues are in state A or B a link will always be granted to one of the queues; since the queues are not distinct it is not necessary to consider the probability that a particular queue is granted a link. (The case in which $m > 2$ is more complex because multiple queues in state B could have their packets passing through multiple links.)

Transition probabilities for the module chain will be described by first describing feeding queues individually. Furthermore, to simplify analysis it is assumed that at most one queue to which a module is connected will be congested.

The arrival processes into all feeding queues are assumed independent. Since an arriving packet's destination is random with a uniform distribution and independent of other packets, the probability that a packet's path will take it through the congested queue is $1/m$. Therefore if a feeding queue is in state E the transition probability to state A is $r_{0,j-1}/m$ and to state B is $r_{0,j-1}(m-1)/m$, where $r_{0,j-1}$ denotes the probability of arrival to an empty queue in stage $j-1$.

The transition out of state A requires that the congested queue be not full. Let v_j denote the service probability for a queue in stage j. It will be shown that the probability a congested queue is not full is approximately v_j. If a feeding queue is in state A the transition probability to E is then $v_j(1-\alpha_{j-1})$, where α_{j-1} is the probability that a queue in stage $j-1$ will not be empty given that a packet moved out of the queue in the previous cycle. The transition probability from A to A is $v_j\alpha_{j-1}/m + 1 - v_j$. The transition probability from A to B is $v_j\alpha_{j-1}(m-1)/m$.

Because the module chain models situations in which one queue is congested, the head packet of a queue in state B will always advance to the next stage. Then, the transition probabilities from B to E, A, and B, are $1-\alpha_{j-1}$, α_{j-1}/m, and $\alpha_{j-1}(m-1)/m$ respectively.

The module chain transition probabilities are found from the feeding queue transition probabilities. Let $Q_{F1,t}$ and $Q_{F2,t}$ be the states of the feeding queues, $F1$ and $F2$, in stage $j-1$ at time t. The probability of a module going from state $\{A,A\}$ to $\{A,B\}$ is then

$$\begin{aligned}
P[\mathbf{M}_{j-1,t} = \{A,B\} \,|M_{j-1,t-1} = \{A,A\}] = \\
P[\mathbf{Q}_{F1,t} = B \mid Q_{F1,t-1} = A] \\
= \frac{m-1}{m}\alpha_{j-1}v_j .
\end{aligned}$$

Note that only one of the head packets could be served, so this equation reflects the one possible transfer. A more complex case is for a transition from $\{A,E\}$ to $\{A,B\}$:

$$\begin{aligned}
P[\mathbf{M}_{j-1,t} = \{A,B\} \,|M_{j-1,t-1} = \{A,E\}] = \\
P[\mathbf{Q}_{F1,t} = A \mid Q_{F1,t-1} = A]\, P[\mathbf{Q}_{F2,t} = B \mid Q_{F2,t-1} = E] + \\
P[\mathbf{Q}_{F1,t} = A \mid Q_{F1,t-1} = E]\, P[\mathbf{Q}_{F2,t} = B \mid Q_{F2,t-1} = A] \\
= \left(1 - v_j + v_j\frac{1}{m}\alpha_{j-1}\right)\frac{m-1}{m}r_{0,j-1} + \frac{1}{m}r_{0,j-1}v_j\alpha_{j-1}\frac{m-1}{m}.
\end{aligned}$$

All of the state transition probabilities for the module chain for stage $j-1$ of an $(n,2,d)$ network appear in the transition matrix below. State labels appear above and to the side of the matrix halves.

$$\mathbf{T}_M = \begin{array}{c} \\ \{A,A\} \\ \{A,B\} \\ \{A,E\} \\ \{B,B\} \\ \{B,E\} \\ \{E,E\} \\ X \end{array}
\begin{array}{c}
\begin{array}{ccc} \{A,A\} & \{A,B\} & \{A,E\} \end{array} \\
\left(\begin{array}{ccc}
1-v_j+\frac{v_j\alpha_{j-1}}{2} & \frac{\alpha_{j-1}(2-2v_j+v_j\alpha_{j-1})}{4} & \frac{(2-2v_j+v_j\alpha_{j-1})r_{0,j-1}}{4} \\
\frac{v_j\alpha_{j-1}}{2} & \frac{\alpha_{j-1}(1-v_j+v_j\alpha_{j-1})}{2} & \frac{(1-v_j+v_j\alpha_{j-1})r_{0,j-1}}{2} \\
v_j(1-\alpha_{j-1}) & (1-\alpha_{j-1})(1-v_j+v_j\alpha_{j-1}) & 1-v_j+\frac{v_j\alpha_{j-1}}{2}-r_{0,j-1}+\frac{3v_jr_{0,j-1}}{2}-v_j\alpha_{j-1}r_{0,j-1} \\
0 & \frac{v_j{\alpha_{j-1}}^2}{4} & \frac{v_j\alpha_{j-1}r_{0,j-1}}{4} \\
0 & v_j\left(1-\alpha_{j-1}\right)\alpha_{j-1} & \frac{v_j(\alpha_{j-1}+r_{0,j-1}-2\alpha_{j-1}r_{0,j-1})}{2} \\
0 & v_j(1-\alpha_{j-1})^2 & v_j\left(1-\alpha_{j-1}\right)\left(1-r_{0,j-1}\right) \\
0 & 0 & 0
\end{array}\right.
\end{array}$$

$$
\begin{array}{cc}
 & \begin{array}{cccc} \{B,B\} & \{B,E\} & \{E,E\} & X \end{array} \\
\begin{array}{l} \{A,A\} \\ \{A,B\} \\ \{A,E\} \\ \{B,B\} \\ \{B,E\} \\ \{E,E\} \\ X \end{array} &
\left.\begin{array}{cccc}
0 & \frac{(1-v_j)\alpha_{j-1}r_{0,j-1}}{4} & \frac{(1-v_j){r_{0,j-1}}^2}{4} & 0 \\
\frac{(1-v_j)\alpha_{j-1}}{2} & \frac{(1-v_j)\alpha_{j-1}r_{0,j-1}}{2} & \frac{(1-v_j){r_{0,j-1}}^2}{2} & 0 \\
0 & \frac{(1-v_j)(\alpha_{j-1}+r_{0,j-1}-2\alpha_{j-1}r_{0,j-1})}{2} & (1-v_j)(1-r_{0,j-1})r_{0,j-1} & 0 \\
\frac{(1-v_j)\alpha_{j-1}}{2} & \frac{(1-v_j)\alpha_{j-1}r_{0,j-1}}{4} & \frac{(1-v_j){r_{0,j-1}}^2}{4} & 0 \\
(1-v_j)(1-\alpha_{j-1}) & \frac{(1-v_j)(\alpha_{j-1}+r_{0,j-1}-2\alpha_{j-1}r_{0,j-1})}{2} & (1-v_j)(1-r_{0,j-1})r_{0,j-1} & 0 \\
0 & (1-v_j)(1-\alpha_{j-1})(1-r_{0,j-1}) & (1-v_j)(1-r_{0,j-1})^2 & 0 \\
v_j & v_j & v_j & 1
\end{array}\right).
\end{array}
$$

3.3 Congestion Duration

The expected duration of congestion is found by combining the expected number of cycles before state X is entered for all possible congestion start states. By definition, $A \in M_{j-1,t}$ holds for a module state at the start of congestion and $X = M_{j-1,t+t_c}$ is the module state at the end of congestion, while $X \neq M_{j-1,\tau}$ for $t \leq \tau < t + t_c$, where t_c is the congestion duration. The states at the start of congestion are computed assuming that before congestion starts both the number of packets in the feeding queues as well as their destinations are independent. The probabilities that a feeding queue will be in states A, B, and E are $(1 - p_{0,j-1})/m$, $(1 - p_{0,j-1})(m-1)/m$, and $p_{0,j-1}$ respectively. Since the queues are assumed independent at this point the probability of the possible congestion start conditions, $\{A,A\}$, $\{A,B\}$, and $\{A,E\}$, are $((1 - p_{0,j-1})/2)^2$, $2((1 - p_{0,j-1})/2)^2$, and $2((1-p_{0,j-1})/2)p_{0,j-1}$, respectively for an $(n, 2, d)$ basic network. The initial state probabilities for stage $j-1$ module states $\{A,A\}$, $\{A,B\}$, and $\{A,E\}$, given that a queue which they are feeding is in the congestion start condition are then $(1 - p_{0,j-1})/(3 + p_{0,j-1})$, $2(1 - p_{0,j-1})/(3 + p_{0,j-1})$, and $4p_{0,j-1}/(3 + p_{0,j-1})$, respectively.

The expected congestion duration is computed by finding the expected number of cycles before absorption given the initial state distribution and the transition probabilities. This same computation also yields the arrival rate to the congested queue, denoted $r_{c,j-1}$. Let $m_{S,j-1,t} = P[\mathbf{M}_{j-1,t} = S]$, the probability that a module will be in state S at cycle t. Let $\vec{m}_{j-1,t} = (m_{\{A,A\},j-1,t}, m_{\{A,B\},j-1,t}, m_{\{A,E\},j-1,t}, m_{\{B,B\},j-1,t}, m_{\{B,E\},j-1,t}, m_{\{E,E\},j-1,t}, m_{X,j-1,t})$ be the state distribution for the module at cycle t for an $(n, 2, d)$ network. Then if congestion starts at $t = 0$ with state distribution

$$\vec{m}_{j-1,0} = \left(\frac{(1 - p_{0,j-1})}{(3 + p_{0,j-1})}, \frac{2(1 - p_{0,j-1})}{(3 + p_{0,j-1})}, \frac{4p_{0,j-1}}{(3 + p_{0,j-1})}, 0, 0, 0, 0 \right),$$

the state distribution of the module at cycle t is $\vec{m}_{j-1,t} = \mathbf{T}_M^t \vec{m}_{j-1,0}$, where $\mathbf{T}_M^t$ is the module transition matrix raised to the power t. The expected number of cycles to absorption in stage j, $t_{c,j}$, (the same as the expected congestion duration) is then

$$t_{c,j} = \sum_{\tau=1}^{\infty} \tau(m_{X,j-1,\tau-1} - m_{X,j-1,\tau}). \tag{1}$$

The arrival rate during congestion is given by

$$r_{c,j} = \frac{1}{t_{c,j}} \sum_{\tau=0}^{\infty} m_{A,j-1,\tau},$$

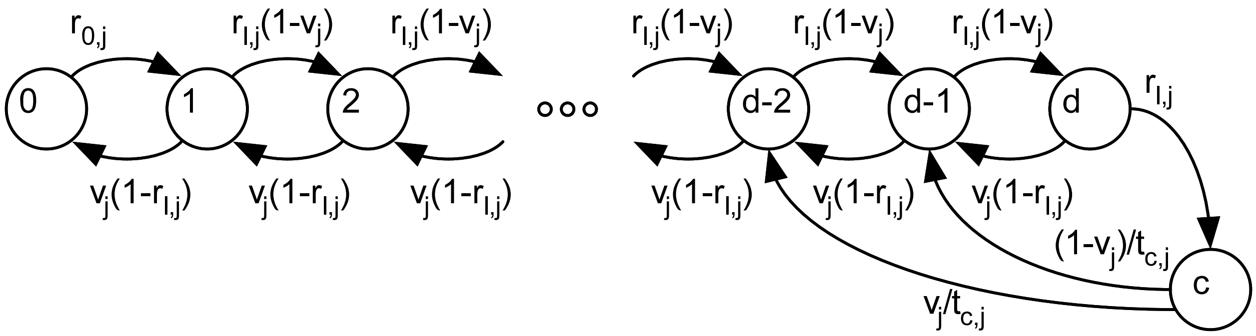

FIG. 3. State transition diagram for a queue in stage j using the sticky model. The congested state, c, indicates that the queue has zero or one slots free. Symbol $t_{c,j}$ denotes the expected congestion duration (determined using the module chain).

where $m_{A,j-1,\tau} = m_{\{A,A\},j-1,\tau} + m_{\{A,B\},j-1,\tau} + m_{\{A,E\},j-1,\tau}$.

3.4 The Queue Model

The queues are modeled using Markov chains. The chains for stages 1 through $n-1$, which will be referred to as the *sticky queues* or *queues* for brevity, each have $d+2$ states. The states are labeled $\{c, 0, 1, \ldots, d\}$, where c is a symbol indicating the congested state. Let Q_j denote the state of a queue in stage j. If the queue is in state $Q_j \in \{0, 1, \ldots, d\}$ then it contains Q_j items and is not congested. If it is in state c then it is congested and may contain d or $d-1$ items.

The state transition diagram for the queue appears in Figure 3; for clarity the self loops are omitted. The chain has one more state, c, than an M/M/1/d queue model; that state is entered if there is an arrival when the queue is full, the amount of time in the state is determined by the expected congestion duration, $t_{c,j}$, described above. By definition, at the last cycle of congestion there are $d-1$ items in the queue, so the first state after congestion would be for $d-2$ or $d-1$ items, depending upon whether the head packet of the queue completes service. The queue in the first stage is modeled as in Yoon's analysis.

A closed form expression for the stationary distribution of queue state probabilities is readily obtained using standard balanced flow techniques. (See [5,8] for a description of these techniques.) Let symbol $p_{i,j}$ for $0 < j < n$ and $0 \le i \le d$ denote $P[\mathbf{Q}_j = i]$, the probability of a queue containing i items and not being congested. The symbol $p_{c,j}$ will denote $P[\mathbf{Q}_j = c]$, the probability of a queue being congested. As with Yoon's analysis, $p_{i,0}$ will denote $P[\mathbf{Q}_0 = i]$ for $0 \le i \le d$, the probability that the queue in stage 0 contains i items. Let $r_{l,j}$ denote the probability of arrival to a queue in stage j when the queue is neither empty nor congested. Then stationary distribution of queue states is given by

$$p_{0,j} = \left[1 - \frac{r_{0,j}}{r_{l,j}(1-v_j)} + \frac{r_{0,j}t_{c,j}\left(r_{l,j}(r_{l,j}+2/t_{c,j})(v_j-1) - v_j/t_{c,j}\right)\omega_j{}^{d-2}}{v_j\left(2r_{l,j}v_j - r_{l,j} - v_j\right)}\right.$$
$$\left.\frac{r_{0,j}\left(1-\omega_j{}^{d-1}\right)}{r_{l,j}\left(1-v_j\right)\left(1-\omega_j\right)}\right]^{-1}$$
$$p_{S,j} = \begin{cases} \frac{r_{0,j}\omega_j{}^i p_{0,j}}{r_{l,j}(1-v_j)}, & \text{if } 0 < S < d-1; \\ \frac{r_{0,j}(-r_{l,j}-v_j+r_{l,j}v_j)\omega_j{}^{d-2}p_{0,j}}{v_j(-r_{l,j}-v_j+2r_{l,j}v_j)}, & \text{if } S = d-1; \\ \frac{r_{l,j}r_{0,j}(v_j-1)\omega_j{}^{d-2}p_{0,j}}{v_j(-r_{l,j}-v_j+2r_{l,j}v_j)}, & \text{if } S = d; \\ \frac{t_{c,j}r_{l,j}{}^2 r_{0,j}(v_j-1)\omega_j{}^{d-2}p_{0,j}}{v_j(-r_{l,j}-v_j+2r_{l,j}v_j)}, & \text{if } S = c; \end{cases} \tag{2}$$

for $0 < j < n$ where $\omega_j = \frac{r_{l,j}(1-v_j)}{v_j(1-r_{l,j})}$.

In the queue chain congestion is modeled as one state; it will be important to distinguish between a queue being congested with one slot free and being congested while full. These two conditions will be referred to as *substates,* denoted c_1 and c_0 and their respective probabilities will be denoted $p_{c_1,j}$ and $p_{c_0,j}$ for a queue in stage j. These probabilities can be approximated by assuming that congestion duration is infinite. In that case and with a non-zero service rate there will be balanced flow between c_1 and c_0 and so $p_{c_1,j}(1-v_j) = p_{c_0,j}v_j$. This equation makes use of the fact that in a congested queue at all but the last cycle of congestion there will always be an arrival in state c_1. Solving this equation yields

$$\begin{aligned} p_{c_0,j} &= (1-v_j)p_{c,j} \\ p_{c_1,j} &= v_j p_{c,j}. \end{aligned}$$

From the second of these equations it can be seen that the probability of a queue having one slot free given that it is congested is approximately v_j. Finally, let $p_{f,j} = p_{d,j} + p_{c_0,j}$, denote the probability that a queue in stage $0 < j < n$ is full.

3.5 State and Rate Relationships

The arrival rate when a queue is empty, $r_{0,j}$, $0 < j < n$, is computed taking into account the knowledge that there was no arrival at the previous cycle. Let a queue in stage j be empty at cycle $t : Q_{j,t} = 0$. For an $(n, 2, d)$ basic network the state of the module which feeds the queue must be one of $\{\{A, A\}, \{A, E\}, \{E, E\}\}$, where a feeding queue in this module would be in state B if it had a packet bound for the empty queue. Let M' be the state of a module, similar to M but without the absorbing state. In each case, the probability that the module would have a packet which is to pass through the empty queue at the next cycle, $P[B \in \mathbf{M}'_{j-1,t} \mid M'_{j-1,t-1}]$, can be computed. Assume that the states of the queues making up the module are independent. Then $r_{0,j}$ is a sum of these probabilities weighted by the probability of each of the corresponding states. The probabilities for an $(n, 2, d)$ network appear in the table below. The first column shows a module state for each row, the second column gives the conditional probability of the row's module state, and the third column shows the probability that there will be a packet bound for B in the next cycle given the row's state.

TABLE II

States of a Module Feeding an Empty Queue

$M'_{j-1,t-1}$	$P[\mathbf{M}'_{j-1,t-1} \mid B \notin \mathbf{M}'_{j-1,t-1}]$	$P[B \in \mathbf{M}'_{j-1,t} \mid M'_{j-1,t-1}]$
$\{A,A\}$	$\frac{(1-p_{0,j-1})^2}{4} \frac{4}{(1+p_{0,j-1})^2}$	$(1-p_{f,j})\alpha_{j-1}/2$
$\{A,E\}$	$2p_{0,j-1}(1-p_{0,j-1})\frac{1}{2}\frac{4}{(1+p_{0,j-1})^2}$	$1-\left(1-\frac{r_{0,j-1}}{2}\right)\left(1-(1-p_{f,j})\alpha_{j-1}\frac{1}{2}\right)$
$\{E,E\}$	$p^2_{0,j-1}\frac{4}{(1+p_{0,j-1})^2}$	$1-\left(1-\frac{r_{0,j-1}}{2}\right)^2$

The resulting arrival rate is

$$
\begin{aligned}
r_{0,j} = &\sum_{S\in\{\{A,A\},\{A,E\},\{E,E\}\}} P[\,\mathbf{M}'_{j-1,t-1} = S \mid B \notin \mathbf{M}'_{j-1,t-1}\,]\; P[\,B \in \mathbf{M}'_{j-1,t} \mid \mathbf{M}'_{j-1,t-1} = S\,] \\
= &\Bigg[\frac{\alpha_{j-1}\,(1-p_{f,j})\,(1-p_{0,j-1})^2}{8} \\
&+ (1-p_{0,j-1})\,p_{0,j-1}\left(1-\left(1-\frac{\alpha_{j-1}\,(1-p_{f,j})}{2}\right)\left(1-\frac{r_{0,j-1}}{2}\right)\right) \\
&+ {p_{0,j-1}}^2\left(1-\left(1-\frac{r_{0,j-1}}{2}\right)^2\right) \\
&\Bigg]\frac{4}{(1+p_{0,j-1})^2}.
\end{aligned} \tag{3}
$$

The arrival rate to a queue is state dependent. An *overall arrival rate,* denoted r_j, will be needed. It is defined to be the unconditional probability of a packet being ready to enter a queue in stage j. By definition, it can be found in terms of the queue state distribution and the other arrival rates:

$$
r_j = r_{0,j}p_{0,j} + r_{l,j}(1-p_{0,j}-p_{c,j}) + r_{c,j}p_{c,j}. \tag{4}
$$

Since the service rate for a queue is independent of the state of the queues it feeds the overall arrival rate can be found using only the probability that a feeder queue is empty, as in Yoon's analyses. Thus,

$$
r_j = \begin{cases} 1-\left(1-\frac{1-p_{0,j-1}}{m}\right)^m, & \text{if } 0<j<n; \\ \lambda, & \text{if } j=0, \end{cases} \tag{5}
$$

where λ is the arrival rate to the network. Note that the queue state distribution is a function of $r_{0,j}$, $r_{l,j}$, $t_{c,j}$, and, v_j. The rate $r_{l,j}$ can be determined given $r_{0,j}$, $t_{c,j}$, v_j, r_j, and expressions for $p_{0,j}$ and $p_{c,j}$.

Define the *flow rate* to be the expected number of packets that pass through a link per cycle. For the basic network and the sticky model at equilibrium flow rate and throughput are the same, otherwise they may differ. The flow rate in the sticky model can be found by summing the arrival rates over queue states, except states in which the queue is full. Let ρ_j be the flow rate into a queue in stage $j > 0$, then

$$\rho_j = r_{0,j} p_{0,j} + r_{l,j}(1 - p_{0,j} - p_{c,j} - p_{d,j}) + (1 - 1/(t_{c,j} v_j)) p_{c_1,j}.$$

This equation is similar to (4), the difference being that cases where the queue is full are not considered. For the first stage the flow rate is given by $\rho_0 = \lambda(1 - p_{d,0})$.

It will be shown that the arrival rate to a queue in the congested state with one slot free is $1 - 1/(t_{c,j} v_j)$. Since v_j is approximately the fraction of time a congested queue has one slot free, $t_{c,j} v_j$ is the expected number of cycles a queue will have one slot free during a period of congestion. As long as the queue is congested and not in the congestion end condition there will be an arrival whenever one slot is free whereas in the congestion end condition there will not be an arrival. The expected number of arrivals is thus $t_{c,j} v_j - 1$. Dividing by $t_{c,j} v_j$ yields the arrival rate to a one-slot-free congested queue, $1 - 1/(t_{c,j} v_j)$.

Recall that α_j denotes the probability that a queue in stage j is not empty in a cycle, given that a packet left that queue in the previous cycle. It can be found by considering the probability that the queue would be empty: $(1 - r_{l,j}) p_{1,j} / (1 - p_{0,j})$. Then considering balanced flow between a queue's empty and other states:

$$\alpha_j = 1 - \frac{(1 - r_{l,j}) p_{1,j}}{1 - p_{0,j}} \qquad \text{for } 0 < j < n.$$

The service rate for a queue is independent of the state of the queue. It is computed based upon the state distribution of the queues which it feeds and the other queues in the module. One way of finding the service rate is by dividing the flow rate computed at the following stage by the occupancy probability,

$$v_j = \frac{\rho_{j+1}}{(1 - p_{0,j})}, \tag{6}$$

for $0 \leq j < n - 1$. For the last stage the service rate equation used by [1,17] is used:

$$v_{n-1} = \frac{\left(1 - \left(1 - \frac{1 - p_{0,n-1}}{m}\right)^m\right)}{1 - p_{0,n-1}}. \tag{7}$$

3.6 Computation of Equilibrium State

The equations above describe the sticky model at equilibrium; they can also be used to find the equilibrium state given only the network description and arrival rate. The system is initialized with the queues in the empty state and all arrival rates zero, except $r_0 = \lambda$. First, the overall arrival rate is computed using (5) using the queue state distributions from the previous iteration (or the initial queue state distributions at the first iteration). Similarly, the service rate is computed using (6,7) as appropriate. To approach equilibrium under certain circumstances this quantity will have to be damped; let v_j be the service rate computed at the t'th iteration and v'_j, ρ'_{j+1}, and $p'_{0,j}$ be the respective quantities computed at iteration $t - 1$. Then

$$v_j = \frac{\frac{\rho'_{j+1}}{(1 - p'_{0,j})} + v'_j}{2};$$

note that at equilibrium this is equivalent to (6).

Next the previously computed queue state distributions and a previously computed value of $r_{0,j-1}$ (or r_0 for the second stage) is used to compute $r_{0,j}$ using (3) for $0 < j < n$. The expected congestion duration, $t_{c,j}$, is found using the previously computed state distributions and the new values of $r_{0,j-1}$ and v_j, and $r_{l,j-1}$ for $0 < j < n$ using (1).

Finally, the new queue state distributions and $r_{l,j}$ are found using (2,4) for $0 < j < n$. This could be solved in closed form for certain d, but the results presented in the paper were computed numerically. For the first stage, which is the same as the corresponding stage in Yoon's analysis, standard equations such as

$$p_{0,j} = \left(1 + \frac{1-{\omega_j}^d}{(1-\omega_j)\,(1-v_j)} - \frac{1}{1-v_j} + \frac{r_j\,{\omega_j}^{d-1}}{v_j}\right)^{-1},$$

$$p_{q,j} = \begin{cases} p_{0,j}\frac{\omega_j^q}{1-v_j}, & \text{if } 0 < q < d; \\ p_{0,j}\frac{\omega_j^{d-1}}{1-v_j}\frac{r_j(1-v_j)}{v_j}, & \text{if } q = d; \end{cases}$$

where $\omega_j = \frac{r_j(1-v_j)}{v_j(1-r_j)}$, can be used.

The results obtained are iterated until the difference between the results computed on subsequent iterations is sufficiently small, using some criteria. The criteria used for this study was that $p_{0,n-1}$ had to vary by less than 1% (after the first $n-1$ cycles in which it would remain at one). For the results reported here only 50 to 60 iterations were necessary to yield almost three digits of precision. The accuracy of the results will be discussed below.

4 Analysis Results

The predictions of Yoon's analysis, Mun's analysis, and sticky state analysis are compared with data taken from simulation. The simulator perfectly simulates the basic network; simulations were performed for 40,000 or more cycles, confidence intervals were computed for severe parameter sets (large queue and network sizes and heavy traffic) and were much smaller than the differences with the analytical models. For example, the 95% confidence interval for the throughput of $(8,2,50)$ networks at $\lambda = .90$ based on three runs was $[.7186, .7190]$. Similarly, there was only insignificant differences in queue state "probabilities" between simulator runs.

The sticky state model was motivated by the observation that the state distribution of queues in simulated basic networks have peaks at 1 and $d-1$ items. The peaks can be seen at occupancy 7 in Figure 4*(a)* where the queue state distributions in each stage of a simulated $(5,2,8)$ network are plotted. The ability of the sticky analysis to model these peaks is shown in Figure 4*(c)* where the peaks also appear. Distributions for Yoon's and Mun's model appear in Figure 4*(b)* and *(d)*; the peaks are not evident.

The sticky model is less effective at predicting the low end of queue state distributions in congested networks. This can be seen in Figure 4 where the sticky model and Yoon's model underpredict. Mun's model, in contrast, overpredicts the low end of the distribution, especially in the last stages of the network. The sticky model is better at predicting the queue-empty state than Yoon's model, which is important in determining throughput.

Queue state distributions in moderately loaded networks, arrival rate .7, appear in Figure 5. In these plots the effectiveness of using a different arrival rate based on an empty queue can be seen: The distributions obtained using the sticky model and simulation closely match. The distribution in all stages obtained using Yoon's model are identical.

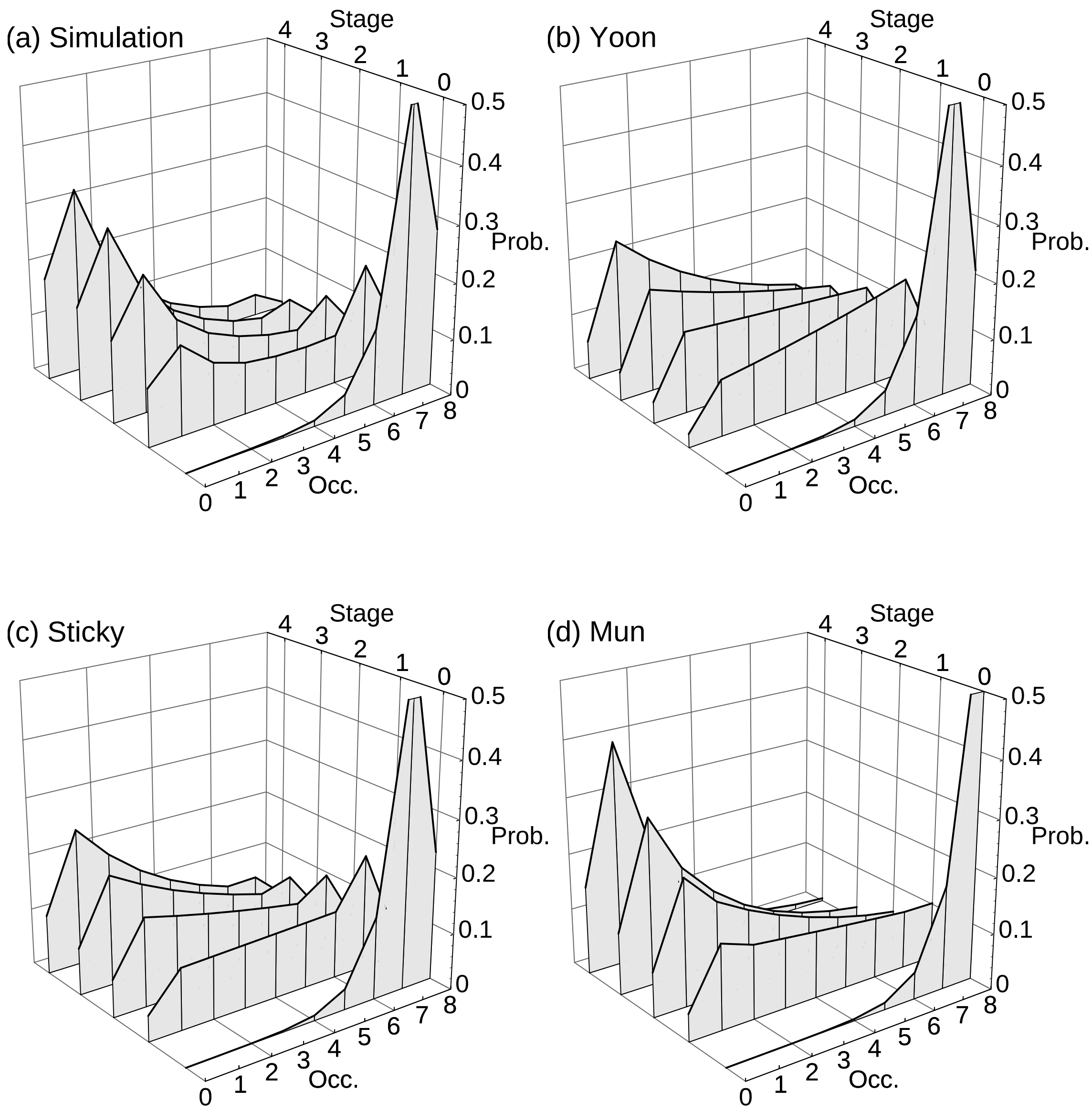

FIG. 4. Queue occupancy (number of items in queue) distributions in each stage of an (5,2,8) basic network offered traffic at rate .9 obtained from *(a)* simulation; *(b)* analysis using Yoon's model, *(c)* analysis using the sticky model, and *(d)* using Mun's model

Note that the distributions obtained for the first stage using simulation, sticky, and Yoon's models are similar. In all analyses the arrivals to the first stage are Bernoulli, in both the sticky and Yoon's analyses service in the first stage is geometric. The closely matching distributions suggest that geometric service is a good approximation. The moderate traffic state distributions predicted using Mun's model less resemble simulation than the others, nevertheless the throughput prediction of Mun's and the other models are very accurate at moderate and low arrival rates.

The effect of network size can be seen in Figure 6, where state distributions in $(3,2,8)$, $(5,2,8)$, and $(8,2,8)$ networks offered traffic at .9 determined by each of the analyses are plotted. The distributions for the same stages in corresponding networks are similar (*e.g.,* the distribution

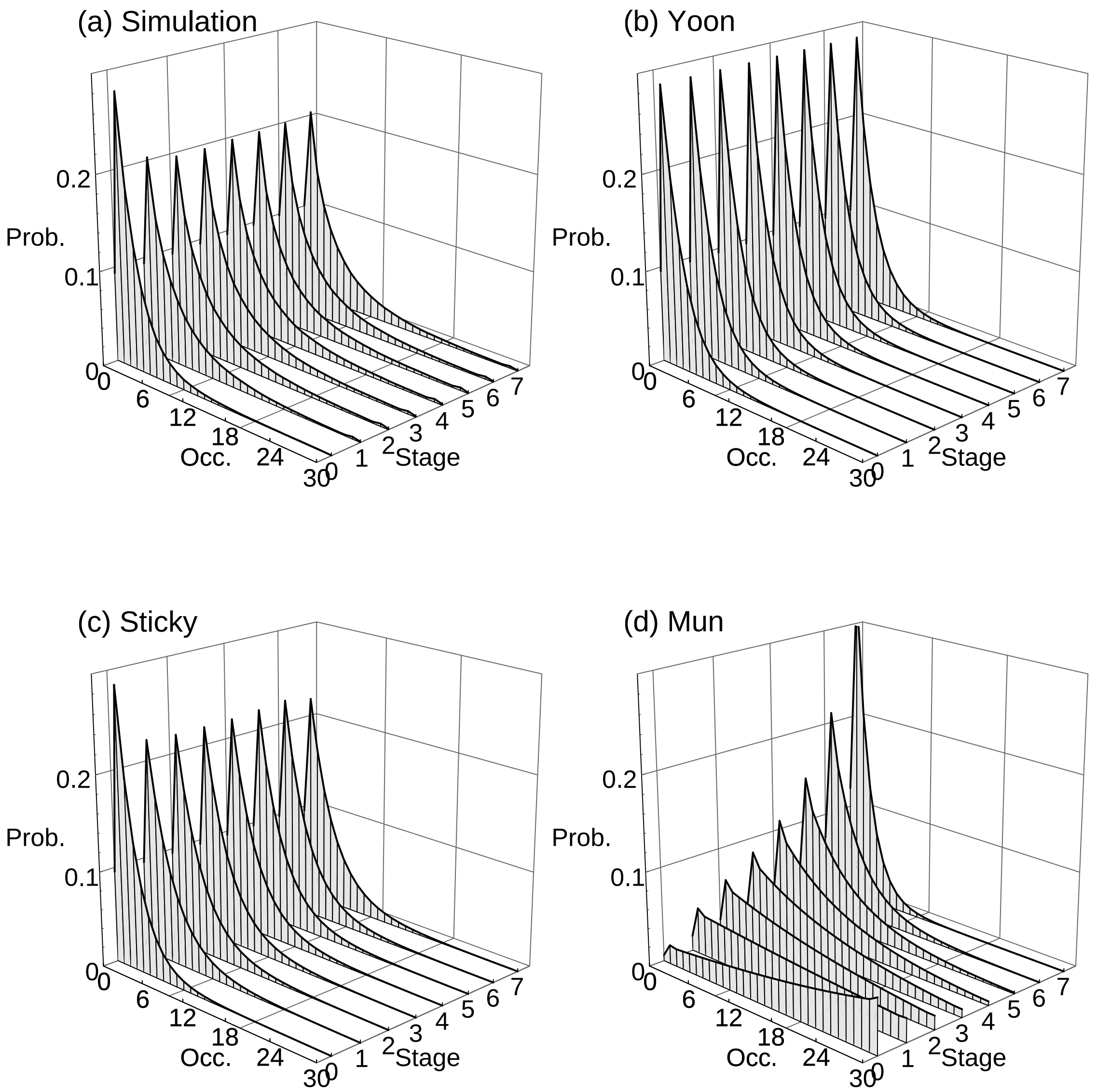

FIG. 5. Queue occupancy (number of items in queue) distributions in all stages of an (8,2,30) basic network offered traffic at rate .7 obtained from *(a)* simulation; *(b)* analysis using Yoon's model, *(c)* analysis using the sticky model, and *(d)* using Mun's model

in stage 1 is similar in $(3,2,8)$, $(5,2,8)$, and $(8,2,8)$, networks). At higher stage numbers the distribution is skewed increasingly towards the one-slot-filled state. This effect is underestimated by sticky and Yoon's analysis and is overestimated by Mun's.[1] The sticky model captures the congestion effect at each size.

The effect of queue size can be seen in Figure 7 where state distributions in $(8,2,3)$, $(8,2,8)$, and $(8,2,30)$ networks offered traffic at .9 determined by each of the analyses are plotted.

[1] To eliminate the possibility that the high slot-one occupancy probabilities are due to an insufficient number of iterations, the states were initialized so that all states (except zero) had equal probability; the distributions were the same.

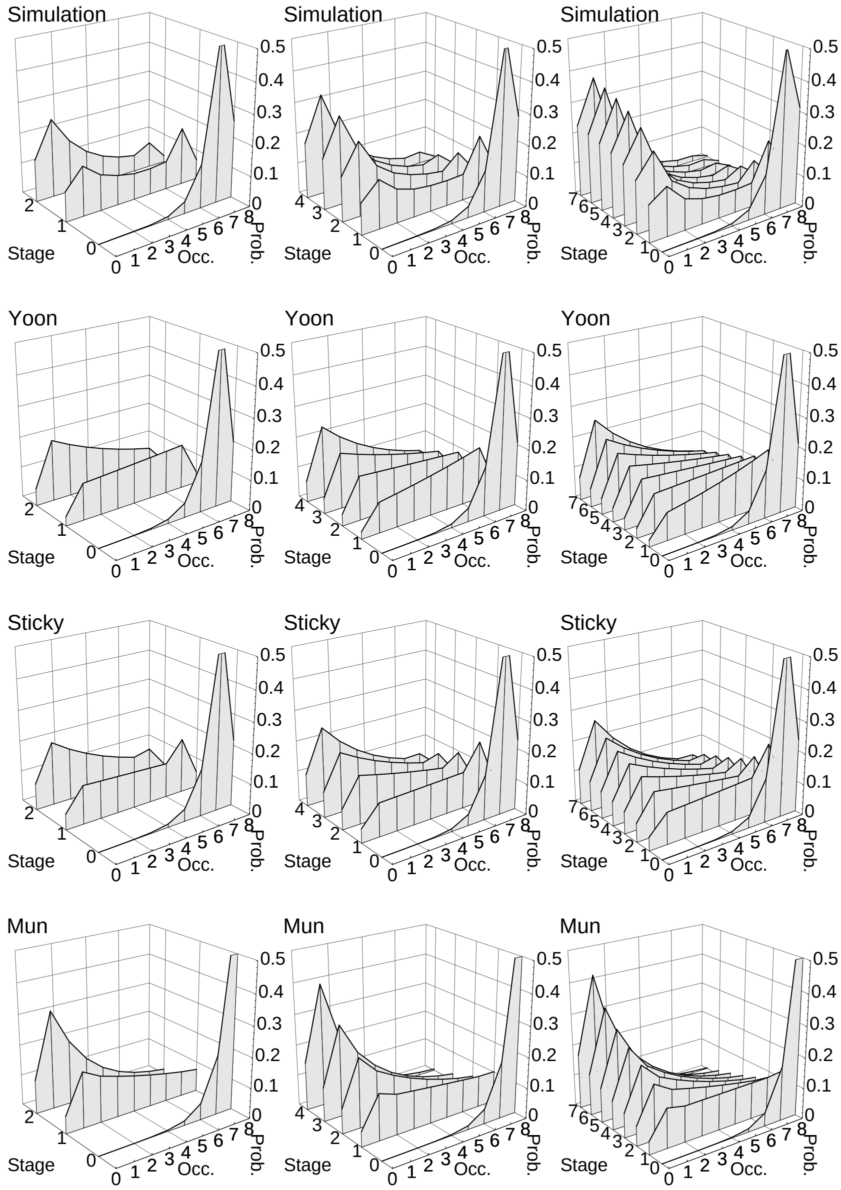

FIG. 6. Queue occupancy (number of items in queue) distribution comparisons of $(3,2,8)$, $(5,2,8)$, and $(8,2,8)$ networks offered traffic at rate .9.

The change in empty-state probability with stage in simulated networks is apparent at all sizes. This effect is not captured using Yoon's or the sticky analysis where the queue states are more evenly spread. Mun's analysis underestimates the change in early stages and overestimates it in later stages. The congestion effect is captured by the sticky model at all queue sizes illustrated (this is difficult to see at size 3).

The effect of arrival rate can be seen in Figure 8 where state distributions in $(8,2,8)$ networks offered traffic at arrival rates .6, .65, .7, and .8 are plotted. At rate .6 the four analyses give similar distributions, with sticky analysis closest to simulation. As arrival rate and congestion increase the peaks at $d-1$ become apparent. As in the other cases, Mun's analysis overestimates low occupancy states while the sticky and Yoon's analysis underestimate them.

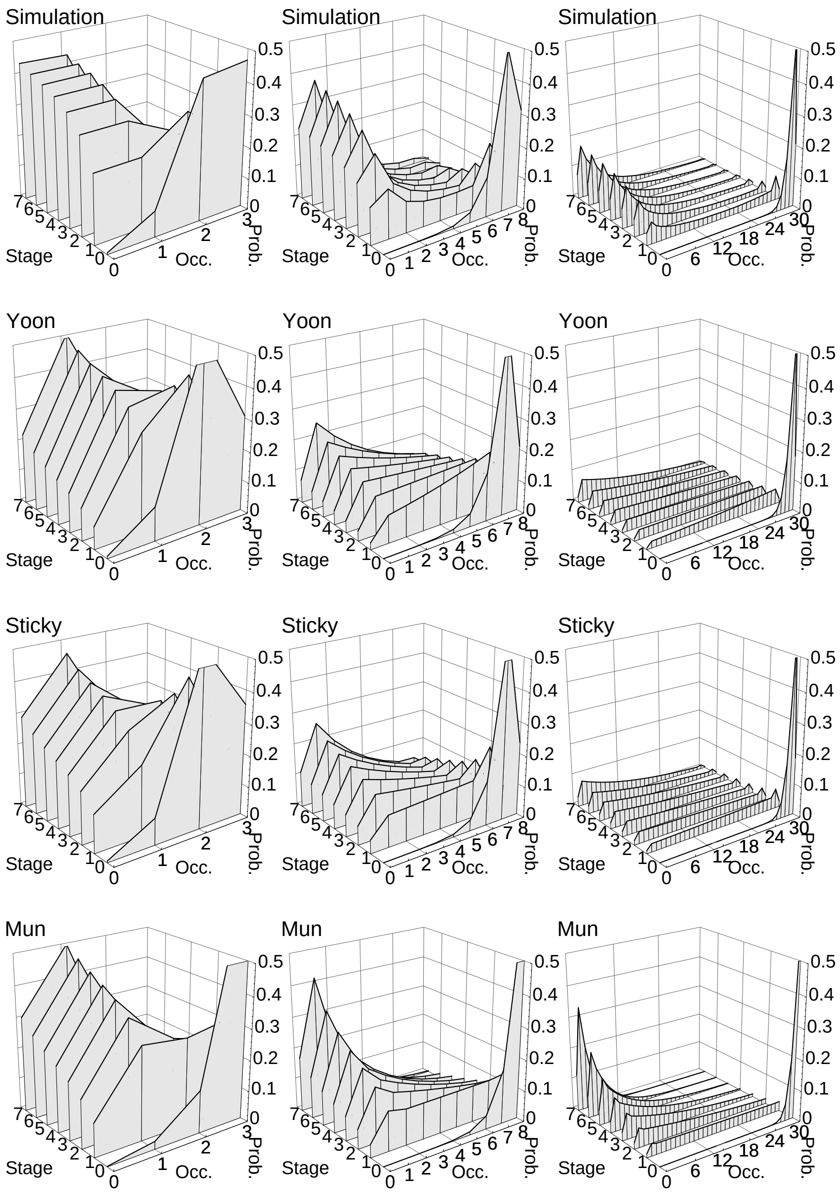

FIG. 7. Queue occupancy (number of items in queue) distribution comparisons of $(8, 2, 3)$, $(8, 2, 8)$, and $(8, 2, 30)$ networks offered traffic at rate .9.

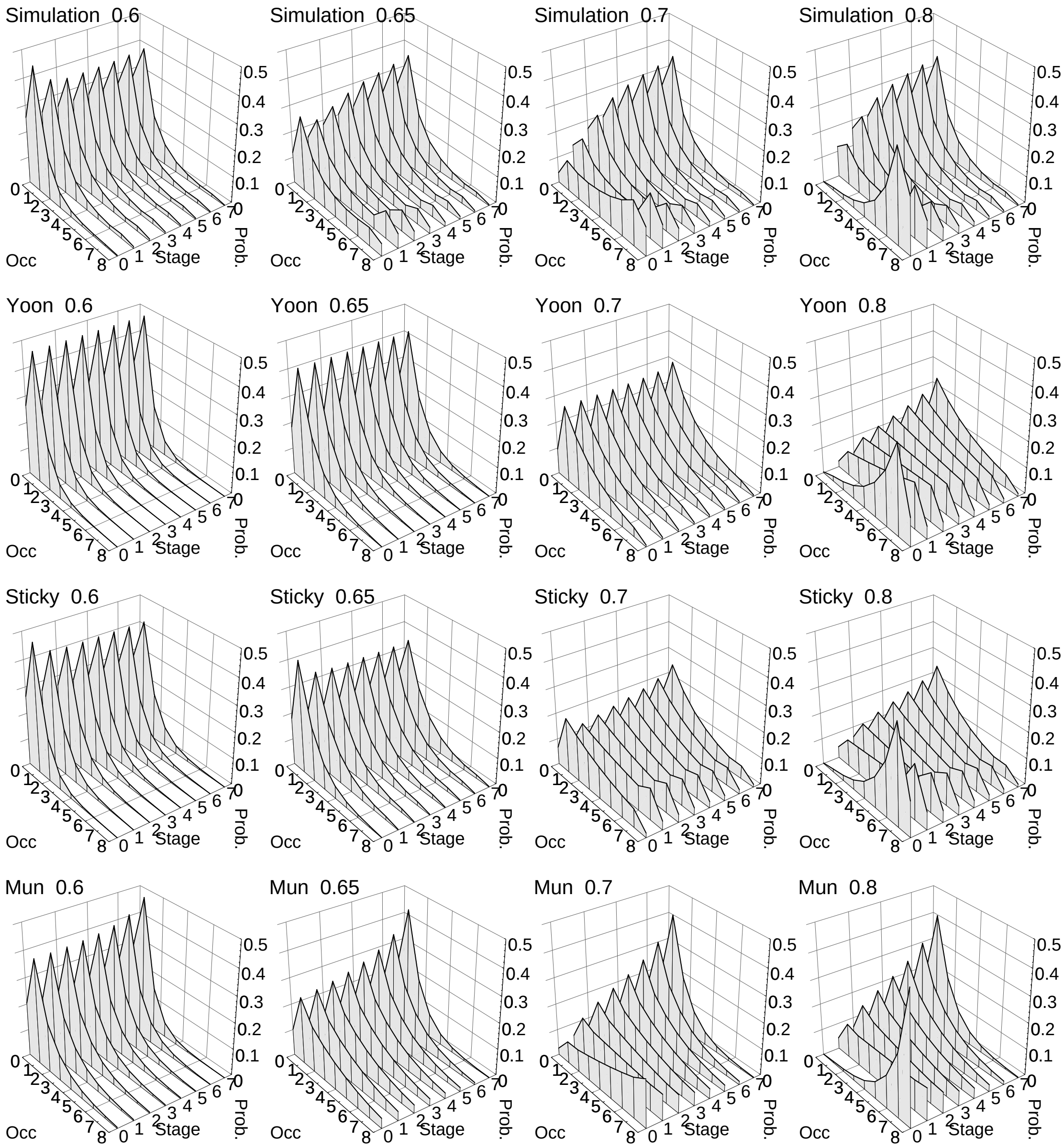

FIG. 8. Queue occupancy (number of items in queue) distribution comparisons of $(8, 2, 8)$, networks offered traffic at rates .6, .65, .7, and .8.

TABLE III
Throughput and Latency of Simulations and Analyses

Queue Size	Arrival Rate	Simulation Thpt.	Simulation Lat.	Fully Ind. Thpt.	Fully Ind. Lat.	Sticky Thpt.	Sticky Lat.
3	.90	.48	20.2	.62	19.9	.58	20.4
5	.90	.56	28.0	.68	30.4	.65	30.5
7	.90	.60	35.9	.71	41.3	.68	40.8
10	.90	.64	47.8	.72	57.9	.70	56.6
20	.90	.69	87.0	.74	113.6	.73	109.4
30	.90	.70	124.0	.74	169.5	.73	162.4
4	.10	.10	9.3	.10	9.2	.10	9.2
4	.30	.30	10.3	.30	10.1	.30	10.1
4	.48	.48	13.6	.48	11.9	.48	12.3
4	.60	.53	20.0	.60	15.2	.59	16.9
4	.72	.53	22.6	.66	22.3	.62	23.6
4	.80	.53	23.4	.66	24.1	.62	24.7
4	.99	.53	24.4	.66	25.4	.63	25.7

TABLE IV
Throughput v. Network Size

Network Size	Simulation	Fully Ind. Analysis	Sticky Analysis	Mun's Analysis
3	.62	.69	.66	.66
4	.59	.68	.65	.65
5	.57	.67	.64	.63
6	.55	.67	.63	.62
7	.54	.67	.63	.61
8	.53	.66	.63	.61

The values of throughput obtained with the sticky model more closely match the simulated values than Yoon's analysis. This is true over a variety of queue sizes, network sizes, and offered traffic rates. The throughput and latency for $(8, 2, 4)$ networks with various arrival rates and queue sizes obtained through simulation and with Yoon's and sticky analyses appears in Table 3. The throughput obtained with the sticky model is closer to simulated values than Yoon's model for all simulations performed. The latency found for some networks having small queue sizes using Yoon's model is closer to simulated values than the sticky model. As can be seen by inspection of Figure 8 this overestimate is due in part to underestimating the fraction of time a queue will have few items.

Throughput versus network size obtained from simulation, the sticky model, Yoon's model, and Mun's model is tabulated in Table 4. The sticky model outperforms Yoon's analysis for each case, although Mun's analysis gives the same or slightly closer predictions.

5 Conclusion

Better predictions of queue state distributions notwithstanding, the difference between the sticky model and simulations is significant and increases with network size. Based upon examination of simulation results and other analysis methods the author believes that the lower throughput in simulated systems is due to congested queues inducing congestion in queues for several stages towards the inputs. The belief is based on the observation that at the onset of congestion in a queue the queue which it feeds is congested more frequently than average. This effect was partially incorporated into a model by determining the service rate for a queue feeding a congested queue and a correlation between congested queues in adjacent stages; from this adjusted service rates were found. See [9]. Although the results were better (at some cost in complexity) correlation of congestion between adjacent stages did not match simulations to a satisfying degree.

The sticky model yields much better results than presented here if a slower service rate is used in the module chain. As of yet the slower service rate has not been quantified using a plausible model, so presentation of these better results will be omitted.

Another observed effect which might be exploited is the variation in service and arrival rates with queue occupancy. In simulations of the basic network, arrival rate, after zero items, slowly increases with queue occupancy. Similarly, service rate slowly drops, with a sharp drop when the queue is full. If such rates could be suitably modeled (without adding states as described above) a relatively simple yet accurate network model might result.

The effect of nonuniform traffic, including hotspots, can be analyzed by straightforward modification of the sticky model. In such analyses offered traffic to an N-input, N-output network is specified using an $N \times N$ *load matrix* which specifies the arrival rate of messages at each input destined to each output [7]. Each queue in the network is modeled (rather than a single representative queue for each stage), arrival and service rates are computed separately for each link.

6 Bibliography

[1] Dias, D.M., and Jump, J.R. Analysis and simulation of buffered delta networks. *IEEE Transactions on Computers.* **30**, 4 (April 1981), 273–282.

[2] Ding, J., and Bhuyan, L.N. Finite buffer analysis of multistage interconnection networks. *IEEE Transactions on Computers.* **43**, 2 (February 1994), 243–247.

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

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

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

[6] Jenq, Y.C. Performance analysis of a packet switch based on single-buffered banyan network. *IEEE Journal on Selected Areas in Communications.* **1**, 6 (June 1983), 1014–1021.

[7] Kim, H.S., and Leon-Garcia, A. Performance of buffered banyan networks under nonuniform traffic patterns. *IEEE Transactions on Communications.* **38**, 5 (May 1990), 648-658.

[8] Kleinrock, L. Queueing systems. New York: John Wiley & Sons. 1975.

[9] Koppelman, D.M. Congested banyan network analysis using congested-queue states and neighboring-queue effects. *IEEE/ACM Transactions on Networking.* **4**, 1 (February 1996), 106-111.

[10] Kruskal, C.P., and Snir, M. A unified theory of interconnection network structure. *Theoretical Computer Science.* **48** (1986), 75–94.

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

[12] Mun, Y., and Youn, H.Y. Performance analysis of finite buffered multistage interconnection networks. *IEEE Transactions on Computers.* **43**, 2 (February 1994), 153–162.

[13] Patel, J.A. Performance of processor-memory interconnections for multiprocessors. *IEEE Transactions on Computers.* **30** (1981), 771-780.

[14] Siegel, H.J., Nation, W.G., Kruskal, C.P., and Napolitano, jr., L.M. Using the multistage cube topology in parallel supercomputers. *Proceedings of the IEEE.* **77**, 12 (1989), 1932–1953.

[15] Szymanski, T., and Shaikh, S. Markov chain analysis of packet-switched banyans with arbitrary switch sizes, queue sizes, link multiplicities and speedups.*Proc. of IEEE Infocom.* April 1989, vol. 3, pp. 960–971.

[16] Theimer, T.H., Rathgeb, E.P., and Huber, M.N. Performance analysis of buffered banyan networks. *IEEE Transactions on Communications.* **39**, 2 (February 1991), 269–277.

[17] Yoon, H., Lee, K.Y., and Liu, M.T. Performance analysis of multibuffered packet-switching networks in multiprocessor systems. *IEEE Transactions on Computers.* **39**, 3 (March 1990), 319–327.