Can We Use Product-Form Solution Techniques for Networks with Alternate Path Routing?

Essia H. Elhafsi and Mart Molle
Dept. of Computer Science and Engineering
University of California
Riverside, CA. 92521
{essia, mart}@cs.ucr.edu

D. Manjunath
Dept. of Electrical Engineering
Indian Institute of Technology, Bombay
Powai Mumbai, India 400 076
dmanju@ee.iitb.ac.in

Abstract*— We define a "forking node" to be a service center with one input feeding two outputs (each served by its own queue) under the control of some internal routing policy. We assume that both outputs lead to paths through which an arriving packet can reach its final destination. However, the mean delays are different between the two downstream paths, so the routing policy should favor the path with the lower downstream delay. Using simulation, we compare the performance of this system under a variety of random, deterministic and state-dependent routing policies, including threshold-based and Join-the-Shortest-Queue with bias ($JSQ + b$). Our results show that $JSQ + b$ routing has significantly better performance than any of the alternatives. Moreover, if the input process to the forking node is Poisson, then standard time series analysis techniques show that its two outputs are very close to being independent Poisson processes. Thus, if we can find a sufficiently accurate and efficient "offline" analytical performance model for a $JSQ + b$ forking node, we can extend the applicability of product-form queueing networks to include such forking nodes. For this reason, we present several ways of modeling the performance of a $JSQ + b$ node, using approximations and/or bounds, and compare their results on some example networks.

Keywords: Asymmetric networks; Parallel servers; Threshold routing; State dependent routing; Performance bounds.

I. INTRODUCTION

In this paper we model the performance of networks with routing algorithms that distribute traffic over multiple paths. Using the terminology of queueing network models, our goal is to study *forking nodes* (such as node A in Figure 1), which are service centers with one input and two outputs. More specifically, we wish to find routing policies for forking nodes that: (i) are easily implementable in real network switches; (ii) provide good performance; and (iii) can be incorporated into tractable analytical network performance models.

To simplify the problem, we will make the following assumptions about the environment. Packet arrivals to the forking node form a Poisson stream with rate λ.[1] The two outputs from the forking node, which we refer to as LINK 1 and LINK 2, operate at the same speed. Furthermore, both output links lead to paths by which an arriving packet can reach its final destination. However the remainder of the network transit delays are different for the two paths. Without loss of generality, we assume that the downstream delay for the path beyond LINK i is d_i, $i = 1, 2$, and that $d_1 \geq d_2 \geq 0$. We define $d \geq 0$ as the difference between d_1 and d_2.

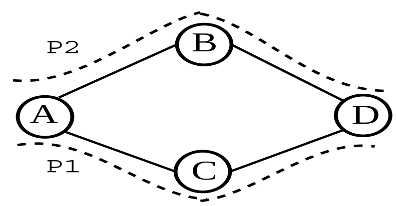


Fig. 1. A network in which there are two paths, P_1 and P_2, from the forking node, A, to the destination node, D. The routing algorithm at node A is aware of the mean delay on each path, and uses this information to determine how much traffic to send over each path.

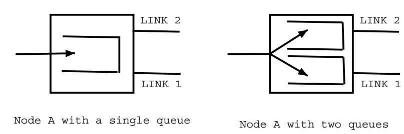


Fig. 2. Queueing options at the forking node A.

There are several options for designing the internal structure of a forking node, as shown in Figure 2. If the node uses a single shared input queue, then the routing decisions can be delayed until a packet is at the head of the queue. Conversely, if the node uses separate output queues for each link, then the routing decisions must be made when a packet arrives. Despite the fact that only the shared queue option can support work conservative scheduling algorithms, network switches generally use separate queues per output port to limit head-of-the-line blocking [14].

In addition to being easy to implement and providing good performance, a good routing policy should also allow us to develop analytically tractable performance models for optimizing the performance of the network. In this context, product-form queueing networks are very attractive, since they can provide precise and detailed performance results with

[1] We are well aware of the extensive literature about long-range dependence in network traffic. However, it has been shown in [11] that the traffic on highly-multiplexed high-speed backbone links looks increasingly like a Poisson process.

relatively low computational complexity [7]. Thus, we will focus our attention on the relationship between routing policies and the product-form solution.

Jackson [10] and Gordon and Newell [9] introduced the product-form property for open and closed queueing networks with exponential interarrival and service time distributions, while Kleinrock [12] was the first to apply such models to the analysis of packet-switched data networks. They have shown that for these networks, a closed-form solution for the equilibrium state probabilities of queueing networks of arbitrary size and complexity exists and it is equal to the product of factors describing the state of each node in the network. This solution is called product-form solution.

The literature with regards to networks having product-form solution is rich [4], [6], [7], [5], [24]; the application to computer systems and their widespread queueing disciplines may be credited to Chandy [7]. In [4], queueing network models were extended to multiple classes of customers and non-exponential distributions. Towsley [20] considers a closed queueing network model with state dependent routing. The routing function is a rational function of the queue length of various downstream queues. He shows that the introduction of routing will preserve the product-form of the equilibrium distribution if the network with no state dependent routing has product-form. Nelson [17] discusses the mathematics leading to the product-form results and the properties of the stochastic process underlying the network model.

However, most practical queueing problems lead to non-product-form networks such as networks with non-exponentially distributed services times, computer systems and networks with asymmetric nodes with simultaneous resource possessions (systems with memory constraints, or with I/O subsystems), models of programs with internal concurrency, and fork-join operations in parallel processing systems. Instead of the costly alternative of simulation, approximate procedures have been considered [5] such as the Flow Equivalent Server (FES) approximation. In general, such techniques are approximations that reduce the model to a similar system that has product-form.

The remainder of this paper is organized as follows. In Section 2, we consider a forking node that uses separate output queues for each link and describe a variety of routing algorithms including static policies (such as simple random routing) and some dynamic threshold-type routing policies (such as join-the-shortest-queue, JSQ). Using simulation, we show that threshold type routing polices perform much better than the alternatives; $JSQ + b$ being the best and random routing the worst. In section 3, we analyze the output processes generated by various routing schemes to determine which ones are compatible with the product-form solution based on Muntz' $M \Rightarrow M$ condition. In section 4, we describe a two-dimensional model of the two queue $JSQ + b$ system and present bounds for the number in each queue and the mean overall delay. In section 5, we consider a threshold-type single queue system as a one-dimensional approximation to the $JSQ+b$ two queue system and compare its behavior to the two dimensional model. This is similar to the two heterogeneous-server system considered by Lin and Kumar [15], Walrand [23], Viniotis and Ephremides [22] and more recently, Koole [13], except in our case we have heterogeneous 'paths' behind each server. In section 6 we present closed-form expressions for the stationary probabilities of the single queue model. In section 7, we present a simple application of our result to a non-product-form queueing network and compute relevant statistics. We conclude with a summary.

II. ROUTING POLICY SELECTION

1. Unbiased Routing Policies

Unbiased policies do not favor one downstream path over the other. However, they may consider local information about the two queue lengths. In this paper, we consider the set of random, deterministic and state dependent routing policies:

- Random routing: An incoming packet is blindly routed to one of the two queues based on a random coin toss. It is well known that random routing can be used in product form queueing networks.
- Round robin: Incoming packets are routed to the two queues deterministically, according to a strict alternation pattern.
- Join-Shortest-Queue (JSQ): An incoming packet is routed to the output link that has a smaller number of waiting packets at its moment of arrival. This greedy algorithm is known to be socially optimal [25] over all non-preemptive policies that use knowledge of the distribution (but not the actual value) of each customer's service time, i.e., it minimizes the discounted expected sojourn time of the packets in the system and maximizes the discounted number of jobs to complete their service in any specified time interval [26]. It also minimizes the expected total time to complete the processing of all jobs arriving before some fixed time [8]. This policy is known to have a complex sum of products closed form solution [1], [2], [3].
- Virtual waiting time (WT): This scheme is similar to JSQ, except that it knows the actual service times for each customer, and not just their average value.

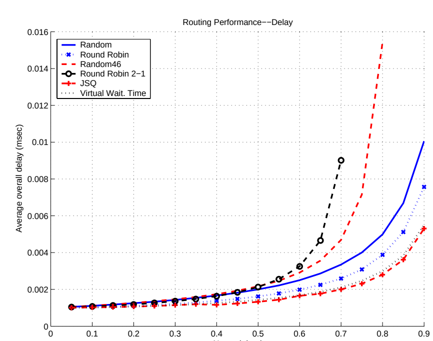


Fig. 3. Performance of unbiased routing policies when the downstream delay behind the queues is the same ($d = 0$).

Figure 3 shows that state dependent routing policies perform well compared to random and round robin routing policies when $d = 0$. We studied different versions of random routing and round robin as shown in Figure 3, which "blindly" favor one of the queues. So rather than splitting the traffic between the queues we force only 40% (33%) of the traffic to use one of the queues using random (round robin) scheme. As a result, their performance decreases drastically especially at high network loads.

However when d is large, which is generally the case, such information should be incorporated in the routing decision. A bias introduced in the routing policy such that packets will prefer the path with the lower downstream delay will improve the performance of the model as we show in the next subsection.

2. Biased Routing Policies

Based on Figure 3, JSQ policy performs better than the alternative routing policies. We will introduce a bias into this policy and compare its performance to the plain JSQ. We refer to this biased routing policy as the $JSQ+b$. We describe $JSQ+b$ in more details.

The routing bias in $JSQ+b$ scheduling provides a significant performance improvement over pure JSQ scheduling when our system model includes an additional source of state-independent delay behind one of the queues.

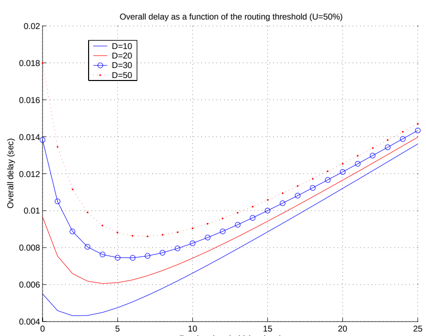


Fig. 4. Reducing the mean delay via threshold routing when there is some extra delay behind Q_1. Network load $\rho = 50\%$, $(d = (10, 20, 30, 50)\ /\mu)$

This improvement is clearly visible in Figure 4, which shows the mean delay as a function of b at $\rho = 0.5$, for several values of the extra delay d behind Q_1 (Q_i is the queue in front of LINK i, $i = 1, 2$). The delay curve seems to be a convex function in b. Similar results were obtained for other network loads ($0 < \rho < 1$) and various downstream delays. Notice that in all cases, the minimum delay occurs for some $b > 0$. Moreover, the optimum value of b/μ is always much smaller than d, and becomes even smaller as ρ increases. It is interesting to note that the optimal solution to the standard parallel queue scheduling problem ($d = 0$) is a "greedy" policy, i.e., JSQ. However, when we generalize the problem by adding some extra delay behind one queue ($d \geq 0$) then the "greedy" policy, i.e., $JSQ+b$ with $b/\mu = d$, is clearly *not* optimal in this case.

We consider the following biased routing policies.

- Join-Shortest-Queue with bias ($JSQ+b$), a dynamic routing policy. We route an incoming packet to the slower path only if the difference in the instantaneous queue lengths between the two queues exceeds b, which we call the routing bias.
- Virtual waiting time with bias ($WT+b$): A new arrival will choose the slower path if the difference in virtual waiting times for the two servers exceeds b.

3. Comparison of the Routing Policies

We compare the routing policies in terms of mean end-to-end delay. We consider a simple three node network and simulate the system for various combinations of d and network loads.

Figure 5, for $d = 20/\mu$ and $0 < \rho < 1$, shows that random routing has the worst delay followed by round robin. We used different versions of random and round robin routing. For random routing we force 40% of the traffic to use the slower path. For round robin we impose that about 33% of the traffic uses the slower path. This has improved their performance especially for low loads. Threshold routing policies however, have a better performance, $JSQ+b$ having the highest followed by the virtual waiting time based routing policy. Similar results are obtained for other values of d. Note that for threshold routing, we compute the delay using the optimal routing bias obtained by simulation.

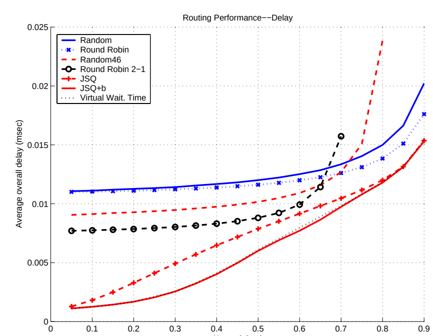


Fig. 5. Comparison of several routing policies. The mean delay in a two-queue system when there is some extra delay behind one of the queues. $(d = 20/\mu)$

III. RELATION TO PRODUCT-FORM

We use the $M \Rightarrow M$ property, due to Muntz [16], to determine whether forking nodes with various routing policies can be incorporated into a product-form queueing network model. A service center satisfies the $M \Rightarrow M$ (Markov implies Markov) property if feeding it a Poisson arrival process causes it to produce a Poisson departure process. If every service center in a queueing network satisfies the $M \Rightarrow M$ property, then we have a sufficient condition for showing that its solution has a product-form. Based on this result, we simulated a forking node and applied time-series analysis to its two output processes to test for goodness of fit

 ISBN: 1-56555-300-4

between the observed interdeparture times distribution and the exponential distribution, and evaluated the autocorrelation and cross correlation functions (as a test for independence). Note that this methodology is *not* intended as a rigorous proof that the product-form does or does not hold for the routing policies we studied. Instead, our goal is to decide whether or not we can use a product-form solution to approximate the performance of a network that includes forking nodes.

For each of the routing policies, we simulate the system with various utilizations (ρ=10% to 90%) and different path downstream delays (d). For the threshold based routing policies, we study the system using the optimal routing threshold, b^*, that we obtain by simulation. We study the distribution of packet interdeparture times from both queues (Q_1 and Q_2) and show that it can be described by an exponential distribution for certain routing schemes. Furthermore, we show that the interdeparture times are independent for other routing policies. For some policies such as the round robin the results do not show that product-form holds.

1. Interdeparture Times Distribution

To show that the interdeparture times distribution from both Q_1 and Q_2 is exponential we use the complementary distribution function (CCDF). The CCDF is defined as $F^c(t) = 1 - F(t)$, where $F(t)$ is the cumulative distribution function. The CCDF of an exponential distribution with mean $1/\lambda$ is

$$F^c(t) = e^{-\lambda t}. \quad t \geq 0$$

The CCDF of packet interdeparture times is a straight line when the Y axis is plotted in log scale, which corresponds to an exponential distribution.

The CCDF of interdeparture times distribution from Q_1 and Q_2 for a network 70% utilized and using the threshold based routing and random routing schemes are given in Figure 6(a) and (b) respectively. We tested networks with various loads and linear least square fitting shows that the CCDFs for all network utilization we tested can be described by an exponential distribution with confidence over 95% for all routing policies with the exception of the round robin scheme.

Table I gives a more detailed approximation of the interdeparture times distribution from each queue, for various network utilizations, to an exponential distribution in terms of confidence levels (%).

TABLE I

CONFIDENCE LEVEL TO APPROXIMATE THE INTERDEPARTURE TIMES TO AN EXPONENTIAL DISTRIBUTION FOR ALL ROUTING POLICIES

Policy	ρ	$C.I.$ ($d = 20/\mu$) Q_1	Q_2	$C.I.$ ($d = 100/\mu$) Q_1	Q_2
Virtual Waiting Time	10	*[2]	96.0	*	94.8
	20	*	95.2	*	95.6
	30	*	97.9	*	95.4
	40	97.7	94.4	*	94.9
	50	98.3	96.4	97.7	96.6
	60	99.5	95.9	96.0	94.3
	70	98.6	94.6	98.8	95.0
	80	98.4	97.5	99.5	95.1
	90	98.6	95.6	98.4	95.6
JSQ+b	10	*	95.2	*	94.8
	20	*	97.3	*	95.0
	30	*	97.3	*	95.2
	40	97.5	96.7	*	95.4
	50	98.1	97.4	99.1	95.0
	60	99.6	96.3	98.4	96.9
	70	98.9	97.0	98.6	95.8
	80	98.2	97.1	98.7	95.1
	90	98.3	96.6	99.2	95.0
JSQ	10	98.0	95.7	97.3	95.7
	20	96.6	96.4	97.7	97.6
	30	98.4	96.9	98.8	96.2
	40	98.6	96.6	98.1	95.6
	50	99.2	97.2	98.7	97.2
	60	98.0	95.2	97.5	95.3
	70	97.9	95.1	99.2	96.7
	80	98.7	96.8	96.8	95.0
	90	98.0	97.9	98.5	95.0
Random	10	97.7	97.2	98.1	98.0
	20	95.5	95.0	97.9	96.6
	30	95.1	97.0	96.5	96.0
	40	96.8	97.2	96.8	96.5
	50	96.9	96.3	97.7	96.4
	60	97.4	96.5	95.8	95.5
	70	95.2	96.7	97.1	98.2
	80	96.5	96.2	97.1	95.4
	90	97.4	95.9	96.5	95.9
Round Robin	10	93.0	89.8	94.2	93.2
	20	97.0	95.2	90.6	92.6
	30	94.8	93.6	90.1	90.1
	40	96.2	96.9	93.3	87.8
	50	92.4	97.3	91.2	95.3
	60	95.9	91.8	91.7	93.0
	70	95.5	96.0	93.5	95.5
	80	94.7	92.5	96.7	97.2
	90	94.9	96.9	91.9	97.5

Not enough data to compute statistics.

2. Correlation in Interdeparture Times

The interdeparture times correlation is captured by the autocorrelation function (ACF), $\rho_{acf}(k)$ which measures the dependency between a series X_t and a shifted version of itself X_{t+k}:

$$\rho_{acf}(k) = \frac{E[(X_t - \mu_x)(X_{t+k} - \mu_x)]}{\sigma_x^2}$$

where μ_x and σ_x are the sample mean and standard deviation respectively.

The correlation of interdeparture times between queues is also captured by the cross correlation (XCF) which measures the dependency between two different series $X_{1,t}$ and $X_{2,t}$ shifted.

$$\rho_{xcf}(k) = \frac{E[(X_{1,t} - \mu_{x_{1,t}})(X_{2,(t+k)} - \mu_{x_{2,t}})]}{\sigma_{x_{1,t}} \sigma_{x_{2,t}}}$$

where $\mu_{x_{1,t}}$ and $\sigma_{x_{1,t}}$ are $X_{1,t}$'s mean and standard deviation respectively, and $\mu_{x_{2,t}}$ and $\sigma_{x_{2,t}}$ are $X_{2,t}$'s mean and standard deviation respectively.

The closer the $\rho_{acf}(k)$ and $\rho_{xcf}(k)$ are to zero the independent the series are. For various values of the downstream delay d and for all routing policies, we computed the autocorrelation

TABLE II

ERROR(%) IN THE AUTOCORRELATION OF THE INTERDEPARTURE TIMES FROM Q_1 (ϵ_1) AND FROM Q_2 (ϵ_2) AND IN THE CROSS CORRELATION (ϵ_c)

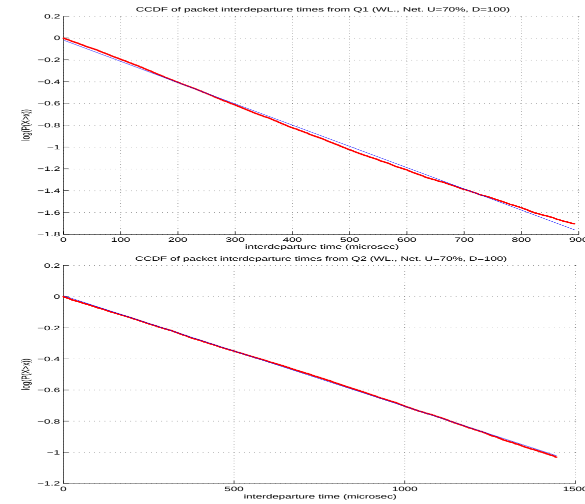
Policy	ρ	$d = 20/\mu$			$d = 100/\mu$		
		ϵ_1	ϵ_2	ϵ_c	ϵ_1	ϵ_2	ϵ_c
Virtual Waiting Time	10	-1[3]	0	-1	-1	0	-1
	20	-1	0	-1	-1	0	-1
	30	0	0	0	-1	0	-1
	40	0	0	0	-1	0	-1
	50	0	0	0	0	0	0
	60	0.5	0	0	0	0	0
	70	0.5	0	0	0.5	0	0
	80	0	0	3.0	1.5	0	1.5
	90	0	0	4.5	0.5	0	1.7
JSQ+b	10	-1	0	-1	-1	0	-1
	20	-1	1.2	-1	-1	0.5	-1
	30	0	2.1	0	-1	0	-1
	40	0	0	0	0	0	2.2
	50	0	0.7	0	0	0	0
	60	0	0.9	0	0	0	0
	70	0	0	0.2	0	0	2.0
	80	0	0	1.5	0	0	1.0
	90	0.5	0	3.2	0	2.9	2.0
JSQ	10	0	0.5	0	0	0	0
	20	0	0.5	0	0	0.5	0
	30	0.5	0.5	0	0	0.5	0
	40	0	0.5	0	0	0.5	0
	50	0	1	0	0	0.5	0
	60	0	1	1.5	0	1	1.0
	70	0.5	1	1.0	0	0.5	1.7
	80	0	0.5	2.0	0.5	0.5	3.0
	90	0	0.5	3.0	0.5	0	3.0
Random	10	0	0	0	0	0	2.7
	20	0	0	2.2	0	0	2.5
	30	0	0	2.2	0	0.5	2.0
	40	0	0.5	3.7	0	0	3.2
	50	0	0	2.7	0	0	2.0
	60	0	0	3.2	0.5	0.5	2.7
	70	0	0	4.5	0	0	2.5
	80	0	0	2.7	0	0	1.0
	90	0	0	4.0	0	0	1.7
Round Robin	10	0	0	3.2	0	0	4.2
	20	0.5	0	5.7	0.5	0	6.5
	30	0	0.5	5.7	0	0.5	5.7
	40	0.5	0.5	3.7	0.5	0.5	7.0
	50	0.5	0.5	4.0	1	1	5.5
	60	0.5	0.5	6.2	1	0.5	7.0
	70	0.5	1.5	7.2	0.5	1.5	7.2
	80	0	0.5	6.7	0.5	0.5	6.0
	90	0	0	6.2	0	0	6.0

Not enough data to compute statistics.

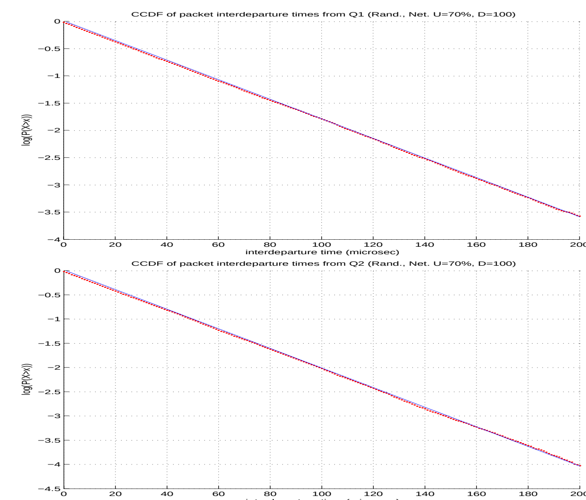
function (ACF) for packet interdeparture times, from Q_1 and Q_2. We also computed the queues cross correlation (XCF). These statistics were computed for 200 lags for over 50,000 consecutive packet interdeparture times.

Figure 7 gives plots of the ACF, of the packet interdeparture times using threshold routing, from Q_1 and Q_2 for a downstream delay d, $d = 100/\mu$. It also shows the XCF of packet interdeparture times from both queues. Similar plots are given in Figure 7 for the random routing policy.

We computed the error, that is the fraction of data outside of the 95% confidence interval bounds, for the auto correlation



(a) CCDF of interdeparture times – $JSQ + b$ routing.



(b) CCDF of interdeparture times – random routing.

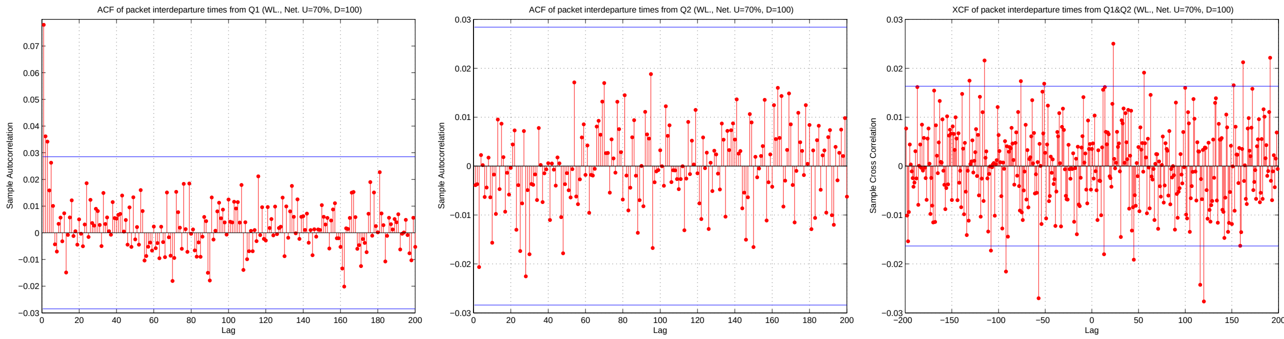
Fig. 6. CCDF of packet interarrival times from Q_1 and Q_2. The distributions can be well approximated by an exponential distribution. ρ = 70%, $d = 100/\mu$

of the interdeparture times from Q_1, Q_2 and for the cross correlation of the interdeparture times from both queues. In all cases, except for the round robin, the error is less than or equal to 5% (Table II), so with a 95% confidence level, our simulation results show that the autocorrelations are very close to zero. Note that in certain cases, not enough data is available to compute the error (due to low load in the system or not enough packets join a particular queue, in this case Q_1).

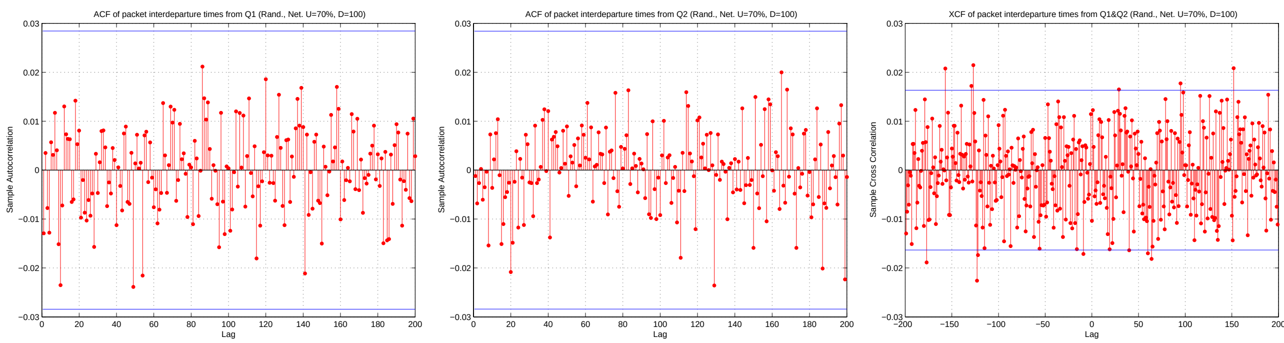
3. Satisfaction of the $M \Rightarrow M$ Property

We showed that for a variety of network utilizations the interdeparture times are exponentially distributed with a confidence level of at least 95%, with the exception of the round robin policy. We also showed that for most policies, the interdepartures from each queue and cross correlations

 ISBN: 1-56555-300-4



(a) Correlation– $JSQ + b$ routing policy



(b) Correlation– random routing policy

Fig. 7. Autocorrelation function of packet interarrival times from Q_1, Q_2 and sample cross correlation (left to right). All the correlation coefficients are within the 95% confidence intervals except for a small number of coefficients. ρ = 70%, $d = 100/\mu$

also pass the test for independence. We conclude that it is very likely that the departure processes from both queues are (or almost are) Poisson. We cannot generalize for sure that the $M \Rightarrow M$ property holds for all policies. However, we can claim with confidence that a network using any of the threshold routing policies we tested can be approximated to having a product-form solution. Furthermore, the threshold routing under test have a much higher performance than random routing that has product-form. Therefore, we will base our further analysis of the system on $JSQ + b$ routing policy.

IV. MODELING THE $JSQ + b$ FORKING NODE

1. Two-Dimensional State Space

Let $Q_i(t)$ be the number of packets in queue i ($i = 1, 2$) at time t, including the packets being transmitted (if any). In this case, we can use $\mathbf{Q}(t) := [Q_1(t), Q_2(t)]$ to represent the state of the system at time t. Clearly, $\mathbf{Q}(t)$ evolves as a two dimensional continuous-time Markov chain (CTMC) in $\mathcal{Z}^{+2}$, with state transitions as shown in Figure 8. In particular, because of the $JSQ+b$ routing policy, an arriving packet will be directed to LINK 2's queue iff $Q_2(t) - Q_1(t) \leq b$. We call the line defined by $Q_2(t) - Q_1(t) = b$ in the state space of $\mathbf{Q}$ its *attractor line*. Notice that for all states that are *not* on this line, a packet arrival transition always moves the state toward the *attractor line*. Thus, for all states to the left of the *attractor line*, an arrival is routed to LINK 2. Similarly, for

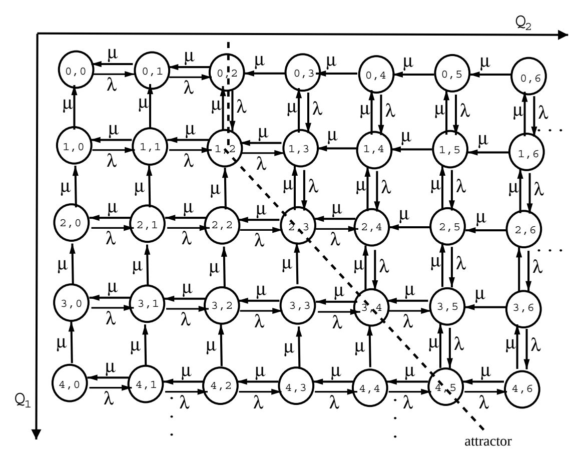
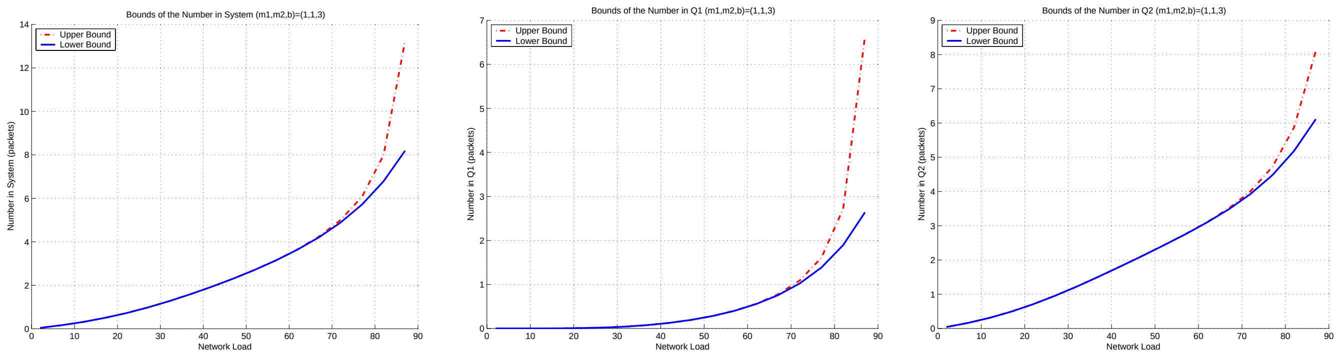


Fig. 8. The transition diagram for the CTMC $\mathbf{Q}(t)$. The *attractor line* is shown for $b = 1$

all states to the right of the *attractor line*, an arrival is routed to LINK 1.

If we assume that $\rho \equiv \frac{\lambda}{2\mu} < 1$, then the Markov chain $\mathbf{Q}(t)$ will be ergodic. Let π_{q_1,q_2} denote the stationary probability of $\mathbf{Q}(t) = [q_1, q_2]$. Given the difficulties of the analysis of the basic JSQ system, we do not anticipate finding an easy solution to the more-general $JSQ + b$ system. Hence we now focus on the problem of finding good approximations, rather

 ISBN: 1-56555-300-4



(a) Bounds of the number in the system (b) Bounds of the number in Q_1 (c) Bounds of the number in Q_2

Fig. 9. Bounds of the number in the queues and in the system for the two queue infinite buffer model when the threshold routing is $b = 3$ and $d = 50/\mu$

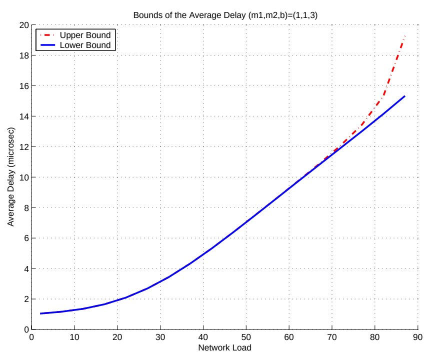


Fig. 10. Bounds of the mean delay in the system for two infinite buffer queues when the threshold routing is $b = 3$ and $d = 50/\mu$

than an exact solution.

2. Approximate Solution via Two-Dimensional Bounds

Following the method of [21], [19], separate Markov chains can be constructed whose solutions can be shown to respectively upper and lower bound the stationary probabilities. In [21], [19], it is shown that suitable modifications of the transitions at the edges of the state spaces can in fact yield tight upper and lower bounds of the stationary probabilities. These stationary probabilities in turn have a closed-form expression that can be easily computed. Performance statistics like mean and variance of the system are also amenable to closed-form and computable solutions. The Markov chains over the truncated state space form a quasi-birth-death (QBD) process that can be solved using matrix geometric techniques [18].

We followed the procedure in [19] to compute bounds of the stationary probabilities and consequently bounds of the number in the queues and in the system, which are shown in Figure 9. The bounds are very tight at low and average loads.

We also compute bounds of the mean overall delay based on a model we discuss in subsection IV-4. The results are shown in Figure 10. Note that the bounds approximate the actual delay very well, especially for low and average network loads.

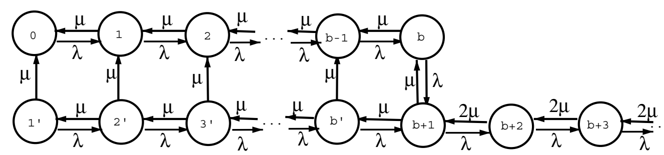


Fig. 11. The CTMC for the single queue model with threshold b.

3. Approximating the System by a Single-Queue Model

Due to the complexity of the two-dimensional system described in the previous section, we now consider the alternative strategy of approximating the system by a simpler one with similar performance, rather than approximating the solution to the original system. In this case, we consider a single-queue model shared by two links and serviced by a threshold routing policy. Each link has a single server, and all the service rates are identical and equal to μ. The path through LINK i is assumed to have a downstream delay of d_i, $i = 1, 2$ where $d_1 \geq d_2 \geq 0$. We assume that the queue has an infinite buffer capacity and that the arrival process is Poisson with rate λ. New arrivals join the queue and are scheduled for transmission as follows.

- If LINK 2 is free, a packet is transmitted on this link.
- If LINK 2 is busy but LINK 1 is free, then a packet is transmitted on LINK 1 only if the queue length is greater than the threshold b.

The CTMC given in Figure 11 describes the transition behavior of the system. Let $Q(t)$ denote the system state at time t where $Q(t)$ takes on values as follows:

$$Q(t) = \begin{cases} 0 & & \text{Queue is empty} \\ i, & i \geq (b+1) & i \text{ in system; both links busy} \\ i, & 0 < i \leq b & i \text{ in system; LINK 2 busy} \\ i', & 1 < i \leq b & i \text{ in system; both links busy} \\ i', & i = 1 & i \text{ in system; LINK 1 busy} \end{cases}$$

4. Comparing Single-Queue vs Two-Queue Results

Before we describe efficient solution techniques for the single-queue model, we must first demonstrate that approx-

 ISBN: 1-56555-300-4

imating the system in this way gives us performance results of sufficient accuracy in comparison to the two-dimensional model. Thus, we now compare the results of these two models to see how well they agree in terms of predicted mean delay and the optimal routing bias.

We define T to be the average end-to-end delay from the forking node to the target destination. Let f be the proportion of time that LINK 1 will be idle. Then f can be expressed as follows:

$$f = \begin{cases} \sum_{i=0}^{b} \pi_i & \text{single queue model} \\ \sum_{j=0}^{\infty} \pi_{0,j} & \text{two queue model} \end{cases}$$

Notice that $(1-f)\mu$ is the mean departure rate along the slower path. On the other hand, the combined departure rate through both paths will be λ if the system is ergodic. Thus, a fraction $\frac{(1-f)\mu}{\lambda}$ of the packets entering the forking node will be subject to the additional downstream delay on the slower path.

We let N be the average number of packets within the forking node, including the ones in service. Using Little's law we have the following overall delay model:

$$T = \frac{N}{\lambda} + (1-f)\frac{\mu}{\lambda}d \qquad (1)$$

Numerically we compute the steady-state probabilities of both models by brute force (i.e., $\pi = \pi P$, where P is the transition matrix of the corresponding CTMC) to get the probability LINK 1 is idle. Based on Equation (1) we compute the overall mean delay for various network loads (ρ=50, 70 and 80%).

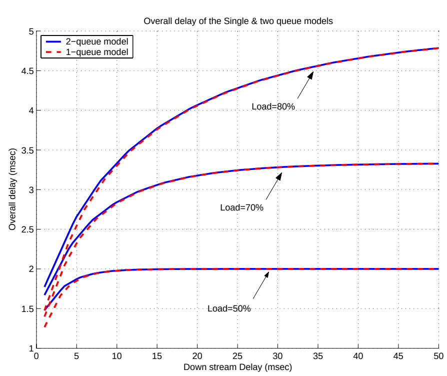


Fig. 12. Comparative results: overall mean delay of the single queue and the two queue models ($\rho = 50, 70, 80\%$).

Figure 12 gives a plot of both models overall mean delay as a function of the downstream delay. For high loads and high d, the delays are very close. For low loads and low values of d, the delays are not as close as for high loads but the gap is acceptable. In all cases, the single queue delay are lower than the two queue delays which is expected.

We also tested both system on how well they estimate the optimum routing bias. We use Equation (1) to find the optimum routing bias (b^* that minimizes the mean delay) for various network loads. Though Equation (1) does show a direct relationship between T and b, but T is a function of b since f depends on b.

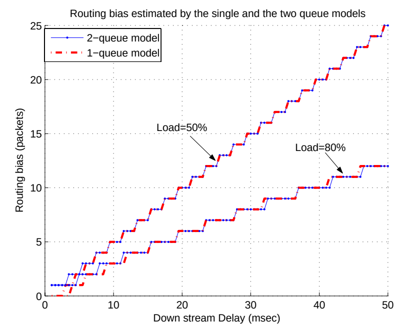


Fig. 13. Comparative results: optimum routing bias estimation of the single queue and the two queue models ($\rho = 50, 80\%$).

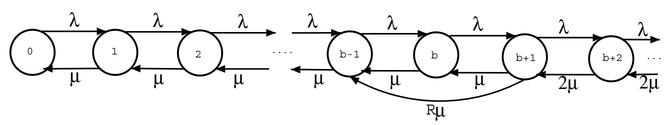


Fig. 14. The truncated CTMC to obtain bounds for the number in the system and the mean delay for the single queue model. When $R = 0$, the CTMC provides upper bounds and when $R = 1$ it provides lower bounds. The routing threshold is b.

Figure 13 presents a plot of both systems estimates of b^*. It is clear that the estimates are very close; in the worst case they are off by 1 packet. Note that when the delay is zero, it is clear that the optimal routing bias is also zero. In this case, packets will be subject to the same delay joining either queues. As a result, the single queue system can be modeled as an $M/M/2$ queueing system. For the two queue system arrivals will be evenly split between both queues.

V. Approximate Solution to the Single Queue Model

Lin and Kumar [15] solved a similar problem and found complex expressions for the stationary probabilities. Therefore, we are looking for approximations or, preferably, tight bounds with simpler expressions that provide better insight and allow for easy computation. Further, a closed-form expression for the mean number in the system and hence the mean delay seems elusive. In the following we derive tight lower and upper bounds for the stationary probabilities of the continuous time Markov chain (CTMC) given in Figure 11. For $\rho \equiv \frac{\lambda}{2\mu} < 1$, the Markov chain is ergodic.

1. Upper Bounds for the Number in the System

We truncate the state space such that when there are $b+1$ packets in the system, a departure from LINK 2 will not be allowed until a departure from LINK 1 occurs. Following the arguments of [19], this provides a sample path based upper bound for the number in the system. Solving the global balance equations of the Markov chain given in Figure 14 when $R = 0$ gives an upper bound of the stationary probabilities, $\{\overline{\pi}_i\}$, and consequently an upper bound of the number in the system and the overall mean delay. $\{\overline{\pi}_i\}$ are expressed as follows:

$$\overline{\pi}_i = \begin{cases} \overline{\pi}_0 (2\rho)^i & \text{for } 1 \leq i \leq b+1 \\ \overline{\pi}_0 2^{b+1} \rho^i & \text{for } i \geq b+2 \end{cases}$$

where,

$$\overline{\pi}_0 = \frac{(1-\rho)(1-2\rho)}{1-\rho(1+(2\rho)^{b+1})}$$

The upper bound, $\overline{N}$, of the mean queue length is:

$$\overline{N} = \overline{\pi}_0 \left[(2\rho)^{b+1} \rho \left(\frac{(b+1)\beta + 1}{\beta^2} + \frac{2((b+1)\alpha - 1)}{\alpha^2} \right) + \frac{2\rho}{\alpha^2} \right]$$

where, $\alpha = 2\rho - 1$ and $\beta = 1 - \rho$

2. Lower Bounds for the Number in the System

We truncate the state space such that when there are $b+1$ in the queue, a departure from LINK 2 will force a departure from LINK 1. The CTMC of the system is given in Figure 14 when $R = 1$. The solution of the stationary probabilities of the Markov chain, following the argument in [19], provides lower bounds of the stationary probabilities of the original system and consequently lower bounds of the number in the system and the overall mean delay.

Let $\{\underline{\pi}_i\}$, be the stationary probabilities of the Markov chain (Figure 14 when $R = 1$), $\{\underline{\pi}_i\}$ are expressed as:

$$\underline{\pi}_i = \begin{cases} \underline{\pi}_0 (2\rho)^i & \text{for } 0 \leq i \leq b-1 \\ \underline{\pi}_0 (\frac{2^{b-1}}{1+\rho}) \rho^{i-1} & \text{for } i \geq b \end{cases}$$

where

$$\underline{\pi}_0 = \left(\frac{1-(2\rho)^b}{1-2\rho} + \frac{(2\rho)^{b-1}}{1-\rho^2} \right)^{-1}$$

The lower bound, $\underline{N}$ of the mean queue length is:

$$\underline{N} = \underline{\pi}_0 \left[\frac{2\rho - (2\rho)^{b+1}}{(1-2\rho)^2} - \frac{b(2\rho)^b}{1-2\rho} + \frac{(2\rho)^{b-1}}{1-\rho^2} \left(b + \frac{\rho}{1-\rho} \right) \right]$$

To assess the tightness of the bounds, in Figure 15 we plot $\overline{N}$ and $\underline{N}$ as a function of ρ. The bounds are very tight for low and high network loads.

3. Bounds for the Mean Overall Delay

Based on the computed upper and lower bound values of the stationary probabilities we compute upper and lower bounds of the mean departure rate from LINK 1 ($\overline{f}$, $\underline{f}$). Having computed upper and lower bound of the number in the system ($\overline{N}$, $\underline{N}$) we use Equation (1) and combine similar bounds to obtain bounds of the mean overall delay. Figure 16 shows a plot of the delay bounds. As in the case of the number in the system, the bounds are very tight for high network loads.

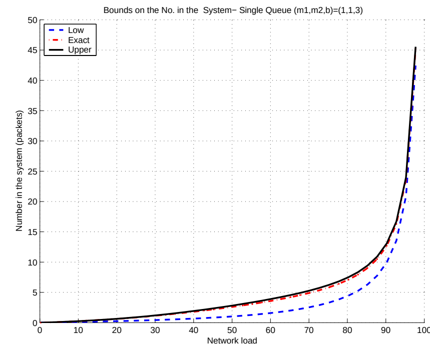


Fig. 15. Bounds on the number of packets in the single queue model. The routing bias $b = 3$, $d = 50/\mu$.

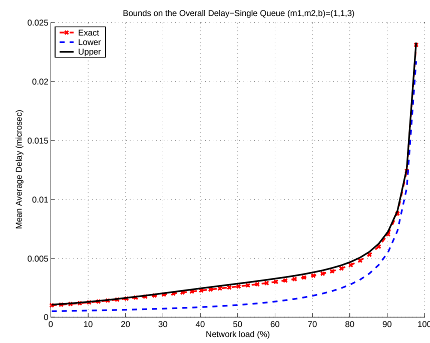


Fig. 16. Bounds on the mean delay in the single queue model. The routing bias $b = 3$, $d = 50/\mu$

VI. APPLICATION TO QUEUING NETWORKS

To illustrate our approach, we now describe the application of our methodology to the simple queueing network shown in Figure 17. In this example, the network is non-product-form because of forking nodes A and B, which use output queueing in combination with state-dependent $JSQ + b$ routing. However, we assume that the remainder of the network satisfies the conditions for a product-form solution. We assume that all the nodes run at the same speed of μ except for C and E which run at a speed of 2μ. To validate the accuracy of our results, we also simulated the system.

1. Computational Method

Starting from the assumption that the network has a product-form solution, we find the flows in each link in the usual way. We then calculate the mean number in system for each service center — using standard results for product-form queues everywhere except at the forking nodes A and B, where we invoke our one- and two-dimensional approximations to estimate the queue length at node A and node B, N_A and N_B, and the departure rates λ_{AC}, λ_{AB} from A's two output links and λ_{BD} and λ_{BE} from B's two output links. In the case of the two queue forking nodes N_{AB} and N_{AC} refer to the individual queues length at node A. Similarly, N_{BD} and N_{BE} refer to the individual queues length at node B These statistics are computed using the upper and lower bound approximations

 ISBN: 1-56555-300-4

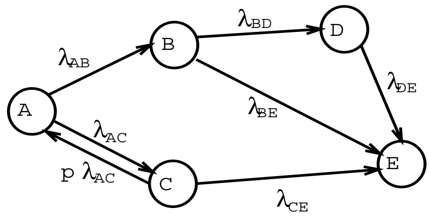


Fig. 17. A network in which there are two forking nodes, A and B. There are several paths, from node A, to the destination node, E. The routing algorithm at node A and node B is aware of the mean delays on the paths, and uses this information to determine how much traffic to send over each path. When $p > 0$ we have a network with feedback.

we described in Section V. If these calculated departure rates from the forking nodes do not match our previously estimated flows, then we update the flows on each link and recompute the number in system for each service center. Finally, we compute the overall number in the system (N) as the sum of the number in the individual nodes, and apply Little's law to obtain the mean delay (T).

2. Numerical Results

Recall that our analysis in Section IV considered two different approximation strategies for modeling a $JSQ + b$ forking node: (i) *approximating the solution to the exact 2-dimensional system* by truncating its state space to leave us with upper and lower bounds that are more easily computed; and (ii) *approximating the system as a much simpler 1-dimensional threshold queue model* which has similar behavior but is much easier to solve (at least to give us upper and lower bounds). For completeness, the results shown in Tables III and IV are designed to illustrate the effects of both approximation strategies. In all cases, we use the same parameters for the underlying queueing network model. However, for all results shown in Table III we have approximated each forking node as the 1-dimensional threshold queue from strategy (ii), whereas in Table IV we have represented each forking node by the "correct" 2-dimensional model. Therefore, the discrepancy between the analytical and simulation results *within each table* shows the approximation error that results from assuming the product-form solution algorithm holds in a queueing network where some of the nodes use state-dependent routing (i.e., single queue with threshold queueing in Table III, and two queues with $JSQ + b$ scheduling Table IV). Conversely, the discrepancy *between tables* shows the approximation error from modeling $JSQ+b$ scheduling as a 1-dimensional threshold queueing system.

The results given in Tables III and IV show that the estimated number in the system for the forking nodes computed via both methods are in good agreement, and they also fit very well the "reference value" obtained by simulation. Actually they bound the actual values provided by the simulator. Similar results are obtained for the overall mean delay.

These results supports our hypothesis that a non-product-form queueing network can be very well approximated by a product-form queueing network.

TABLE III

STATISTICS GATHERED FROM A NON-PRODUCT QUEUING NETWORK (FIGURE 17) THAT IS APPROXIMATED TO A PRODUCT FORM ONE. SINGLE QUEUE AT THE FORKING NODES A AND B; ROUTING IS $JSQ + 2,\ \rho = 75\%$

Statistics	Simulation	Lower Bound	Upper Bound
N	9.30	8.876	10.952
N_A	3.42	3.144	4.767
N_B	0.904	0.8247	1.044
N_C	1.240	1.142	0.896
N_D	0.713	0.7656	1.2456
N_E	3.006	3	3
λ_{AB}	0.3171	0.2891	0.3698
λ_{AC}	0.6828	0.7108	0.6301
λ_{BD}	0.0579	0.0367	0.0250
λ_{BE}	0.2591	0.2525	0.3448
λ_{CE}	0.6828	0.7108	0.6301
λ_{DE}	0.0579	0.0367	0.0250
$T(msec)$	0.006209	0.00592	0.00730

(a) Case of $p = 0$

Statistics	Simulation	Lower Bound	Upper Bound
N	10.664	9.9452	12.734
N_A	4.188	4.08	6.105
N_B	1.2316	0.892	1.279
N_C	1.309	1.022	0.8070
N_D	0.9387	0.9512	1.5429
N_E	2.999	3	3
λ_{AB}	0.351	0.325	0.4045
λ_{AC}	0.649	0.675	0.5955
λ_{BD}	0.0728	0.044	0.0330
λ_{BE}	0.3465	0.281	0.3715
λ_{CE}	0.649	0.674	0.5954
λ_{DE}	0.0728	0.044	0.0330
$T(msec)$	0.0071	0.00663	0.00848

(b) Case of $p > 0$

VII. CONCLUSION

In this paper, we focused our attention on problem of modeling the performance of queueing networks that contain forking nodes. We assume that packets entering the forking node can be directed to one of two different output links, according to a dynamic routing policy. Moreover, we considered a variety of routing policies, including random, deterministic and state-dependent (based on its own internal state, i.e., the instantaneous queue lengths for its two outputs). We also allowed the routing algorithm to consider the mean delays along alternate paths from the forking node to a common final destination.

The main impact of this work is an existence proof: forking nodes that use high performance state-dependent routing policies can be included in tractable analytical models of network performance by approximating them as a service center in a standard product form network. More specifically, we compared the performance of different routing policies, and found that state-dependent policies (such as Join-the-Shortest-Queue) provided the best performance and random routing the worst. We also showed that biasing the routes in favor of the faster path leads to a significant performance improvement,

TABLE IV

STATISTICS GATHERED FROM A NON-PRODUCT QUEUING NETWORK (FIGURE 17) THAT IS APPROXIMATED TO A PRODUCT FORM ONE. TWO QUEUES AT THE FORKING NODES A AND B; ROUTING IS $JSQ+2,\ \rho = 75\%$

Statistics	Simulation	Lower Bound	Upper Bound
N	9.40	9.05	10.94
N_A	4.49	4.206	6.049
N_{AB}	1.51	1.236	4.20
N_{AC}	2.98	2.97	1.849
N_B	1.0245	0.8995	1.033
N_{BD}	0.0995	0.0435	0.0607
N_{BE}	0.925	0.856	0.9723
N_C	0.780	0.903	0.803
N_D	0.104	0.0417	0.0579
N_E	3.01	3	3
λ_{AB}	0.3904	0.3671	0.4059
λ_{AC}	0.6096	0.6329	0.5941
λ_{BD}	0.0496	0.0267	0.0365
λ_{BE}	0.3408	0.3404	0.3694
λ_{CE}	0.6096	0.6329	0.5941
λ_{DE}	0.0496	0.0267	0.0365
$T(msec)$	0.006265	0.006033	0.00729

(a) Case $p = 0$.

Statistics	Simulation	Lower Bound	Upper Bound
N	11.09	10.393	12.67
N_A	5.85	5.42	7.51
N_{AB}	2.133	1.49	2.63
N_{AC}	3.72	3.92	4.87
N_B	1.217	1.01	1.218
N_{BD}	0.137	0.057	0.0894
N_{BE}	1.08	0.955	1.128
N_C	0.885	0.9389	0.862
N_D	0.144	0.0205	0.0805
N_E	3.01	3	3
λ_{AB}	0.452	0.3826	0.435
λ_{AC}	0.5479	0.6173	0.5651
λ_{BD}	0.0669	0.033	0.0497
λ_{BE}	0.385	0.349	0.3851
λ_{CE}	0.5479	0.6173	0.5651
λ_{DE}	0.0669	0.033	0.0497
$T(msec)$	0.00741	0.0069	0.00838

(b) Case $p > 0$

but the optimal bias is not a simple function of the difference in delays between the two alternate paths.

Among all routing policies we considered, $JSQ + b$ consistently gave us the best performance. Unfortunately, even though it would be easy to implement in a network switch, $JSQ + b$ routing is not easy to analyze, even in isolation. Therefore, the main contribution of this paper is to develop and test a simple approximation method that includes a $JSQ + b$ forking node as a service center within the framework of a standard product-form queueing network model. By applying standard time-series analysis techniques to the outputs of a $JSQ + b$ service center, we showed that its "offline" behavior comes very close to satisfying Muntz' sufficient condition service centers that produce a product form solution. We also presented numerical examples to show the effectiveness of our method on various networks.

REFERENCES

[1] I. Adan. Explicit solutions for queueing problems. Memoradum COSOR 97-05, Eindhoven university of technology, April 1997.

[2] I. Adan, J. Wessels, and W. Zijm. Analysis of the symmetric shortest queue problem. *Stochastic Models*, 6:691–713, 1990.

[3] I. Adan, J. Wessels, and W. Zijm. Analysis of the asymmetric shortest queue problem. *Queuing sytems*, 8:1–5, 1991.

[4] F. Baskett, K. Mani Chandy, R. Muntz, and F. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22(2):248–260, April 1975.

[5] G. Bolch, S. Greiner, H. Meer, and K. Trivedi. *Queueing Networks and Markov Chains-Modeling and Performance Evaluation with Computer Science Applications*. John Wiley and Sons, 1998.

[6] K. Mani Chandy and J. H. Howard Jr. and D. F. Towsley. Product form and local balance in queueing networks. *Journal of the ACM*, 24(2):250–263, 1977.

[7] K. Mani Chandy, U. Herzog, and L. S. Woo. Approximate analysis of general queuing networks. *IBM Journal of Research and Development*, 19(1):43–49, 1975.

[8] A. Ephremides, P. Varaiya, and J. Walrand. A simple dynamic routing problem. *IEEE Transactions on Automatic Control*, 25:690–693, August 1980.

[9] W. J. Gordon and G. F. Newel. Closed queueing networks with exponential servers. *Operations Research*, 15(2):252–267, 1967.

[10] J. R. Jackson. Jobshop-like queueing systems. *Management Science*, 10:131–142, October 1963.

[11] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido. A nonstationary poisson view of internet traffic. In *Proceedings of IEEE INFOCOM*, 2004.

[12] L. Kleinrock. *Communication Nets: Stochastic Message Flow and Delay*. McGraw-Hill, 1964.

[13] G. Koole. A simple proof of the optimality of a threshold policy in a two-server queueing system. *Systems and Control Letters*, 2002.

[14] A. Kumar, D. Manjunath, and J. Kuri. *Communication Networking: An Analytical Approach*. Morgan Kaufman Publishers, San Francisco CA, 2004.

[15] W. Lin and P. R. Kumar. Optimal control of a queueing system with two heterogeneous servers. *IEEE Transactions on Automatic Control*, 29, 1984.

[16] R. R. Muntz. Poisson departure processes and queueing networks. Technical report, IBM Thomas J. Watson Research Center, 1972.

[17] R. D. Nelson. The mathematics of product form queueing networks. *ACM computing Surveys*, 25(3), September 1993.

[18] M. F. Neuts. *Matrix Geometric Solutions in Stochastic Models*. John Hopkins University Press, Baltimore, 1980.

[19] R. Tandra, N. Hemachandra, and D. Manjunath. Join minimum cost queue for multiclass customers: Stability and performance bounds. In *Probability in the Engineering and Informational Sciences*, 2002.

[20] D. Towsley. Queueing network models with state dependent routing. *Journal of the ACM*, 27(2):323–337, April 1980.

[21] G. J. van Houtoum, I. J. B. F. Adan, J. Wessesl, and W. H. M. Zijm. Performance analysis of parallel identical machines with a generalized shortest queue arrival mechanism. *OR Specktrum*, 23:411–428, 2000.

[22] I. Viniotis and A. Ephremides. Extension of the optimality of the threshold policy in heterogeneous multiserver queueing systems. *IEEE Transactions on Automatic Control*, 33:104–109, 1988.

[23] J. Walrand. A note on optimal control of a queueing system with two heterogeneous servers. *Systems and Control Letters*, 4:131–134, 1984.

[24] J. Walrand. *An introduction to Queueing Networks*. Prentice Hall, Inc, 1988.

[25] J. Walrand. *Queueing Networks*. Prentice Hall, 1988.

[26] J. Winston. Optimality of the shortest line discipline. *Journal of Applied Probability*, 14:181–198, 1997.