On per-session end-to-end delay distributions and the call admission problem for real-time applications with QOS requirements

D. Yates, J. Kurose
D. Towsley and M.G. Hluchyj

CMPSCI Technical Report 93-20

December, 1993

On per-session end-to-end delay distributions and the call admission problem for real-time applications with QOS requirements

David Yates[*], James Kurose[†] and Don Towsley[†]
Department of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
{yates,kurose,towsley}@cs.umass.edu

Michael G. Hluchyj
Motorola Codex
20 Cabot Boulevard
Mansfield, MA 02048, USA
hluchyj@merlin.dev.cdx.mot.com

CMPSCI Technical Report 93-20
December 10, 1993

Abstract

A crucial problem facing the designers and deployers of future high-speed networks is providing applications with quality of service (QOS) guarantees. For soft real-time applications, which are delay sensitive but loss tolerant, delay distribution is an important QOS measure of interest. In this paper we study (through simulation) the end-to-end delay distribution seen by individual sessions under simple first-come first-served (FCFS) multiplexing in a network model with two significant features: (1) all traffic is connection-oriented, (2) cross traffic along routes is representative of that seen by calls in a moderately sized wide area network (i.e., less than 100 switches). We compare these delay distributions with the worst case point-valued analytic delay bounds predicted by three different techniques for providing such bounds (two of which require a more sophisticated link-level scheduling policy). We also consider the per-hop delay distributions seen as a session progresses "deeper" into the network and determine the sensitivity of these delay distributions to the manner in which the interfering traffic is modeled. Finally, we use our delay distribution results to examine the tradeoff between the QOS requested by a call, the manner in which the QOS guarantee is provided, and the number of calls that are admitted at the requested QOS.

[*] The work of this author was supported by a Motorola Codex University Partnership in Research Grant.

[†] The work of this author was supported in part by the National Science Foundation under grant NCR-911618, and the Advanced Research Projects Agency under contract NAG2-578.

1 Introduction

High speed integrated networks are becoming a more important part of our national and global infrastructure. Because these networks carry a complex mixture of traffic types, technology to support packet switching is needed. A crucial problem that needs to be solved in packet-switched networks is that of providing real-time applications with quality of service (QOS) guarantees. Certain "soft" real-time applications, such as interactive packetized voice and video, are delay sensitive but loss tolerant, and therefore packet loss at the receiver due to excessive end-to-end delay is a QOS measure of interest. Since such losses occur when packets are excessively delayed, the ***delay distribution*** seen by packets in a session becomes a critical performance measure.

In this paper we study (through simulation) the end-to-end and per-hop queueing delay distribution seen by individual sessions under simple FCFS multiplexing. We consider a connection-oriented network model with traditional voice (ON/OFF) source models, as well as a variant of this model. Of particular concern to us is the accurate modeling of the interfering traffic a session sees as it progresses "deeper" into the network. We also examine the sensitivity of these delay distributions to the manner in which the interfering traffic is modeled. Propagation delays and per-packet processing delays are not considered.

Recently, considerable research has been devoted towards the development of techniques for providing a provable analytic bound on the delay experienced by an individual session. Such bounds may potentially be useful in providing a delay-based QOS guarantee to a session having real-time constraints. Techniques have been proposed both for providing a worst case, point-valued delay bound [1, 2, 3, 4, 5] as well as a bound on the delay distribution [6, 7, 8, 9]. In the former case, a specific link scheduling discipline is sometimes needed. However, the worst case delay bounds are readily computable and insure that *all* packets will experience a delay below this bound. As we will see, such a worst case delay bound can thus be used as a "cautious" upper bound on the tail of the delay distribution.

In this paper we also compare our delay distributions obtained via simulation with the point-valued worst case bound on the delay distribution predicted by three methods: *(i)* Cruz's method [1, 2] for computing delay bounds under FCFS multiplexing, *(ii)* Golestani's delay bounds using stop-and-go queueing [10, 11] and *(iii)* Parekh and Gallager's method [3, 4, 5] for computing delay bounds under weighted fair queueing (WFQ) [12, 13]. Our results show that the point at which the tail of delay distributions becomes very small (i.e., on the order of 10^{-5}) is often quite far from the worst case delay either guaranteed (under FCFS or WFQ) or enforced/created (under stop-and-go queueing).

The computation of a worst case delay bound is one way to provide a QOS guarantee to accepted calls – the bound is used by a call admission procedure to determine whether an arriving call can be admitted to the network at its requested QOS without violating existing QOS guarantees. An alternate call admission procedure might be to monitor and measure the delays experienced and make call acceptance decisions based on the observed delays. Such an approach, termed an "observation-based" approach towards call admission [14], was recently proposed in [15, 16]. Another call admission procedure might provide statistical QOS guarantees by approximating the aggregate bit rate of sessions, using their so-called "equivalent capacity"[17, 18]. The final topic addressed in this paper is thus a comparison between a call admission procedure which uses worst case bounds in making call admission decisions for arriving real-time sessions, one which uses equivalent capacity, and an idealized one which has knowledge of the actual delay distributions experienced by already-admitted sessions. As we will see, since the point-valued bound is often far greater than the point at which the delay distribution becomes "small" (i.e., on the order of 10^{-5}) the number of calls that the network can provably support (admit) at a guaranteed QOS is significantly smaller than can be supported if the true, resulting delay distributions under FCFS are known or can be accurately estimated.

The remainder of this paper is structured as follows. In section 2 we describe the network model which was simulated. We also discuss the three bounding techniques described above. In section 3 we present the numerical results (which have been highlighted above). Section 4 discusses the ramifications of these results. Section 5 summarizes the paper.

2 Network Model

In this section we detail the routing, topology, and traffic sources in our network model. We also describe the link-level multiplexing mechanisms required by two of the analytic delay-bounded techniques.

2.1 Network Topology and Routing

We analyze the following network model to determine the quality of service provided to sessions that traverse a fixed route, denoted by R, which is H hops in length. Choosing H defines the network topology, which we call M_H, and R. R traverses specific switches, which we label $S_1, S_2, \ldots, S_H$.

As we will see, for network M_H of reasonable size (e.g., $H = 5$), cross traffic along R is in some sense representative of cross traffic seen by sessions in a moderately sized wide area network. In particular, we will see that the cross traffic encountered on route R varies such that, at the beginning and end of R, cross traffic comes primarily from other sessions near the beginning or end of their routes; near the middle of R the cross traffic is primarily from other sessions near the middle of their routes.

We now describe the topology and routing strategy of M_H which together determine the cross traffic along R. We begin with a discussion of the model of a switch. We then describe an M_1 network, and then larger networks (i.e., M_2, M_3, etc.) which themselves are composed of several such smaller networks.

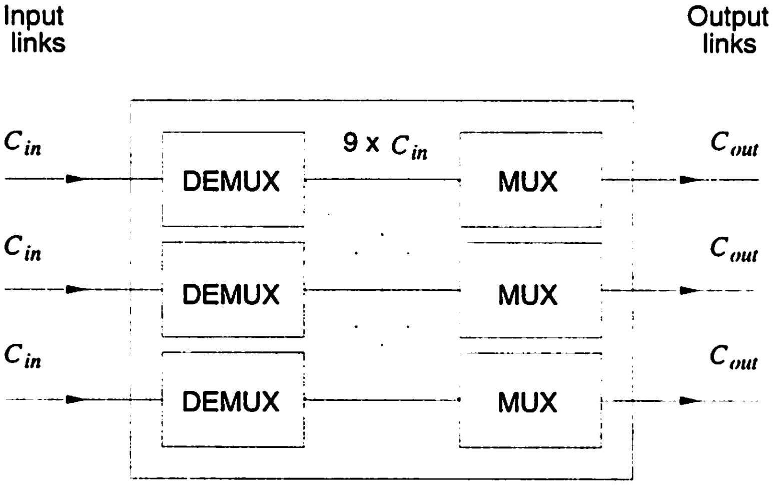

Figure 1: Switch architecture.

An M_1 network consists of a single switch. Our simulated network is built up from the interconnection of these switches, as discussed below. Each switch has three input and three output links, as shown in figure 1. The switch fabric is fully connected with demultiplexers (DEMUX's) at the front end which feed into multiplexers (MUX's) that implement output queueing. Routing in the network will be fixed and connection-oriented. Each link into a switch demultiplexes 1/3 of its sessions to each of the three output links.

Let us now focus on a switch at the input "edge" of the network (e.g., the only switch in M_1, or switch S_1 in figure 2). The particular route of interest, R, enters the network at some switch, which we will label S_1. Each of the three incoming links to such a switch is assumed to multiplex N sessions, each of which has the input traffic characteristics described in the following subsection. For such a switch at the "edge" of the network, the input links are assumed to be of infinite capacity. For all other links in the network we study, we will assume a common link speed of C_l.

In M_H (where $H \geq 2$), traffic leaving smaller M_j networks ($1 \leq j \leq H-1$) forms the cross traffic along route R. Route R traverses switches $S_1, S_2, \ldots, S_H$ and H links leaving these switches. For example, figure 2 shows M_3 with cross traffic along R from M_1 and M_2 subnetworks. Within these M_2 subnetworks, the circled M_1's and the other switches are also used to generate cross traffic. As another example, figure 3 shows M_4 with cross traffic along R from M_1, M_2 and M_3 subnetworks (subnetworks within M_2 and M_3 are not circled in this figure). As in M_1, S_1 is a switch with N active sessions on each of the three input links. The cross traffic at each switch S_j, where $2 \leq j \leq H$, in M_H has three components (one from each input link):

1. 1/3 from an M_{j-1} network,
2. 1/3 from an M_{H-j+1} network,
3. and the remainder from S_{j-1} (the previous node in the route R under study).

For example, in figure 3 the cross traffic at S_4 is from M_3, M_1, and S_3.

We now complete our description of figures 2 and 3, which illustrate M_3 and M_4, respectively. Each input link at the "edge" of the network (i.e., a line which is incident to the left side of a switch and does not connect to an upstream switch) brings N sessions into the network. Route R begins at the lowest switch in the figures and passes through the rightmost switch in the figures. The numerous output links shown which do not directly connect to another switch

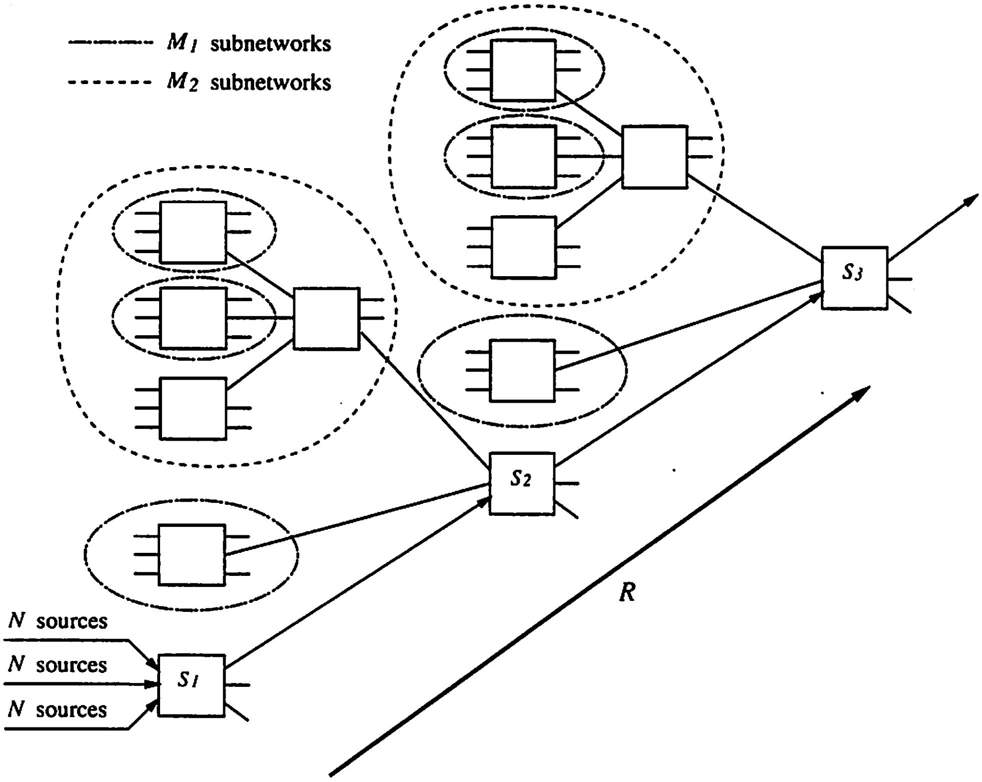

Figure 2: M_3 network.

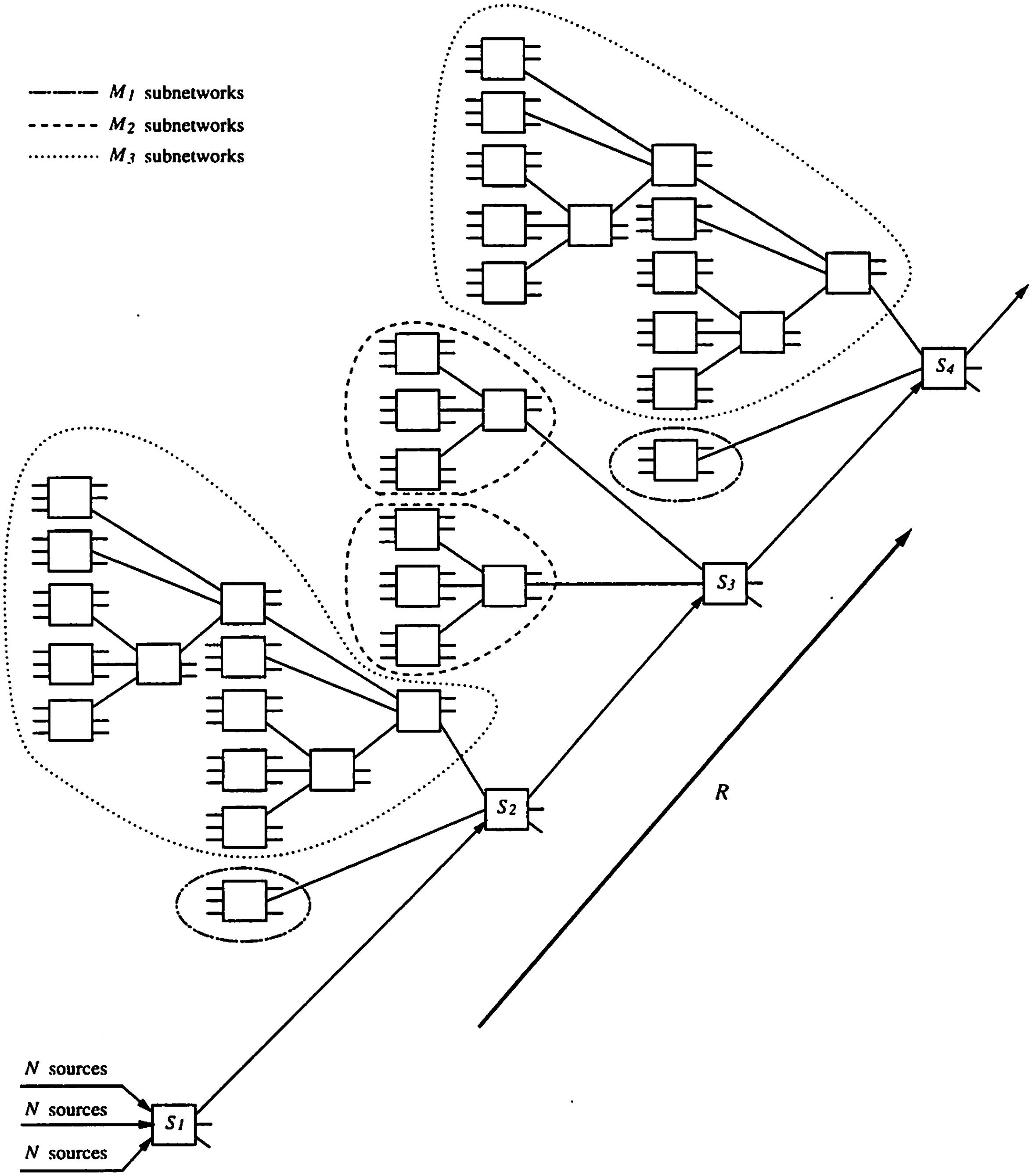

Figure 3: M_4 network.

are assumed to carry traffic to other switches (not shown) which no longer interfere with R. Figure 4 shows M_5, the network we analyze, with the subnetworks which introduce cross traffic condensed.

Given the construction of an M_H network, traffic at the beginning and end of route R will be interfered with primarily by other traffic which is near the beginning or end of its route (i.e., with traffic output from M_j subnetworks where j is either close to 1 or close to H). In this sense, we can consider our simulated network to be representative of a network with an "edge" where traffic originates and terminates, and a "center". Take for example, M_4 (see figure 3). The cross traffic arriving on the upper incoming two links at S_3 has traversed the same number of hops as traffic on R (i.e., two). However, at S_2 and S_4 some traffic (i.e., on the upper link) has traversed many hops (up to four - representing sessions near the end of their route) and some traffic (i.e., on the middle link) has traversed only one hop (representing sessions near the beginning of their route).

There are several properties of our network model worth pointing out. Recall that every switch in M_H routes 1/3 of the sessions arriving at each input link to each of the output links. If we assume that links in the network carry an infinite number of sessions, the network has the following properties:

1. all sessions which depart S_H have traversed at least 2 and at most $\frac{1}{2}H^2 - \frac{1}{2}H + 1$ hops, and

2. for networks of reasonable size ($H < 11$)[1], if $F(H, n)$ denotes the fraction of sessions which depart S_H after traversing n or more hops, then

$$F(H,n) \geq F(H-1,n) \quad \text{for } 2 < H < 11 \text{ and } n > 1. \tag{1}$$

Because of these properties, if QOS guarantees made for R depend on the length of cross traffic routes before they intersect with R, our network model should reveal this. As we will see later, the cross traffic along R results in an upper bound on queueing delay for FCFS which grows exponentially in H.

To illustrate these properties by example, we examine M_4 (see figure 3). The shortest sessions which depart S_4 arrive from M_1 on its middle input link. The longest sessions which depart

[1] An M_{11} network has 88573 nodes.

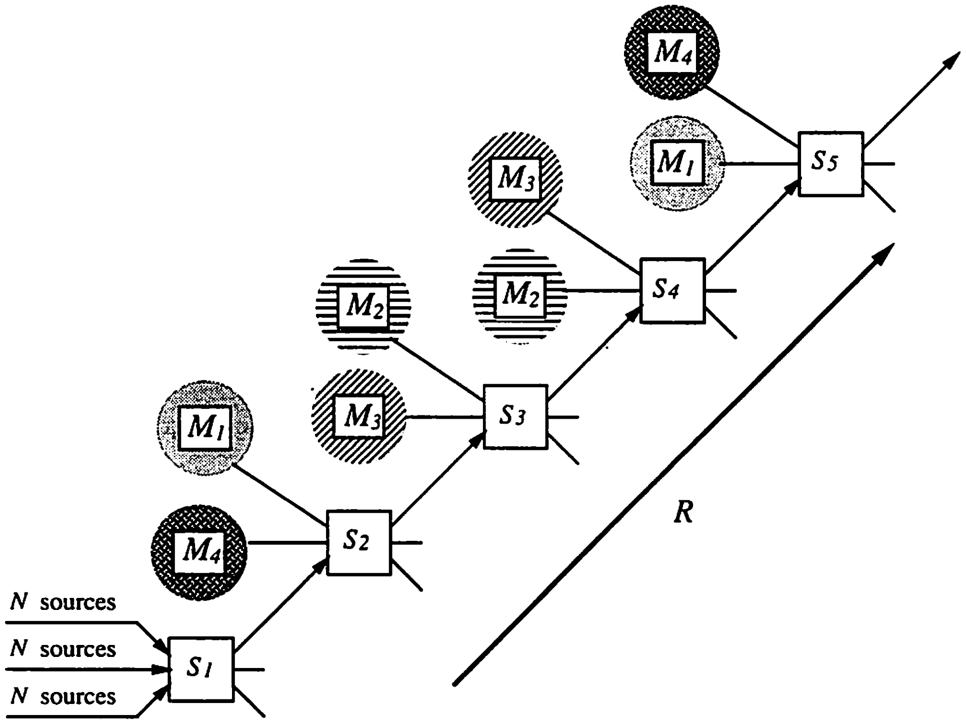

Figure 4: M_5 network.

S_4 (after seven hops) originate from the lower three leftmost switches in the M_3 subnetwork which generates cross traffic at S_2. Table 1 shows the range of $F(H, n)$ for M_4.

n	$F(H,n)$
2	1.0000
3	0.6667
4	0.5555
5	0.0741
6	0.0247
7	0.0041

Table 1: $F(H, n)$ for M_4 ($H = 4$).

In the M_5 network we analyze (see figure 4), the links in the network carry a finite number of sessions (both N and C_l are finite). In this case, the particular fixed routing strategy chosen dictates the length of sessions as they enter and depart the switches along R. However, this does not change the upper bound on queueing delay for the three queueing disciplines we study. We describe the routing strategy used in M_5 and its impact on the length of cross traffic sessions in section A in the appendix.

Finally, we note that because every switch in M_H routes 1/3 of the sessions arriving at each input link to each of its output links, all links in the network carry N calls and have the same utilization.

2.2 Traffic Source Model

Each session source in our network is modeled as having a packet arrival rate which is determined by the state of a Markov chain. The packet stream for the session consists of arrivals at fixed intervals of T ms when the process is in the ON state and no arrivals when it is in the OFF state. The times that a source spends in the ON and OFF states are exponentially distributed with means α^{-1} and β^{-1}, respectively. We approximate the time in the ON state by a geometric distribution where the mean number of packets generated is $\lceil(\alpha T)^{-1}\rceil$. We choose $T = 16$ ms and assume a fixed packet size of $L = 512$ bits. We approximate the time in the OFF state by rounding to the nearest time divisible by the packet transmission time (L/C_l). Figure 5 shows the traffic source model.

We vary the link utilization, denoted by γ, using two methods. The first is to set $\alpha^{-1} = 352$ ms and $\beta^{-1} = 650$ ms, and increase N, the number of calls carried by each link in the network. Note that choosing $\alpha^{-1} = 352$ ms and $\beta^{-1} = 650$ ms gives a model used by other researchers for packetized voice encoded using ADPCM at 32 Kbps [19, 20, 21]. Hence, we refer to this model as the standard voice source model. The second way we vary link utilization is to fix α^{-1} at 352 ms, and adjust the value of β^{-1}. For values of β^{-1} much smaller than 352 ms, the model resembles a continuous packet rate source typical of real-time devices such as remote sensors. We refer to this model as the reduced OFF period model.

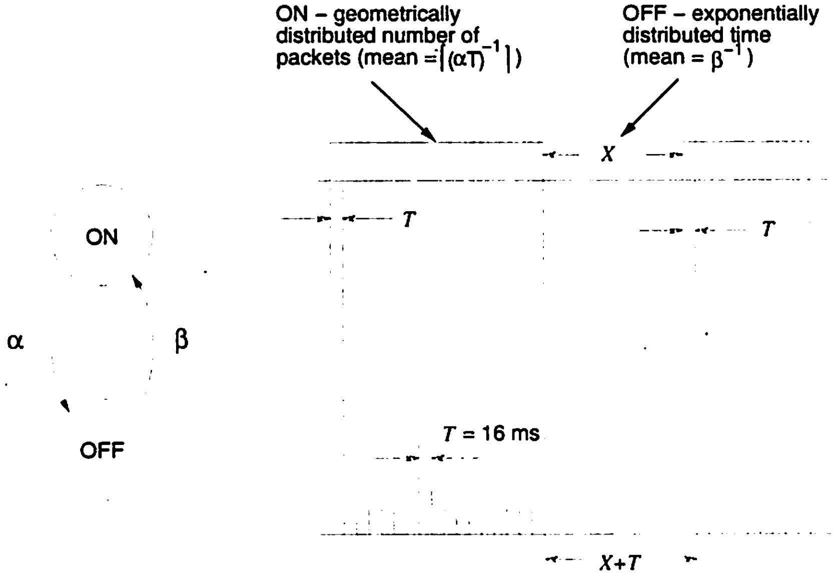
ON - geometrically distributed number of packets (mean = $\lceil (\alpha T)^{-1} \rceil$)
OFF - exponentially distributed time (mean = β^{-1})
X
T
T
ON
α
β
OFF
T = 16 ms
X+T

Figure 5: Traffic source model.

2.3 Multiplexing Disciplines and Delay Bounds for M_H

Subject to the constraint that peak link utilization (i.e., link utilization when all traffic sources are transmitting at their peak rate) not exceed link capacity anywhere in the network, FCFS, stop-and-go queueing, and WFQ all support deterministic upper bounds on delay. The general expression for the delay bound along R in M_H under FCFS is presented in section B in the appendix.

We now briefly describe the delay bounds along R for WFQ and stop-and-go. Each session source can be represented as a *linear bounded arrival process* (LBAP). A source is a LBAP with parameters σ and ρ if the number of bits that arrive within a time interval of length t is bounded by $\sigma + \rho t$, [1]. Given the source characterization in figure 5,

$$\begin{aligned} \sigma &= L \\ &= 512\,\text{bits, and} \\ \rho &= L/T \\ &= 32\,\text{bits/ms.} \end{aligned}$$

Observe that this corresponds to the output of a leaky bucket controlled source with a token generation rate of L/T and a token buffer size of L. However, in our network model, no leaky buckets are actually present to shape the traffic at the edges of the network.

Work conserving queueing disciplines, such as WFQ, which can provide calculable, point-valued, worst case upper bounds on queueing delay are of interest for real-time applications since they impose no lower bound on delay. In the following, we will consider WFQ, where weights are assigned to sessions in proportion to their rate. The upper bound on queueing delay for a session that traverses route R in network M_H is denoted by $\overline{Q_{M_H}}$. From equation (23) in [5],

$$\overline{Q_{M_H}} = \frac{\sigma + (H-1)L}{\rho} + (H-1)\frac{L}{C_l}.$$

In M_5, $\overline{Q_{M_H}} = 81.3$ ms.

Because of the mechanism used by stop-and-go queueing to support an upper bound on delay, a *lower* bound is also imposed on the delay. This forces the delay distribution to lie between the bounds, independent of the link utilization, γ. For stop-and-go queueing, the delay along route R for any packet in an M_H network, which we denote Q_{M_H}, satisfies

$$2HT + (H-1)\frac{L}{C_l} - T < Q_{M_H} < 2HT + (H-1)\frac{L}{C_l} + T,$$

where $2HT$ is the maximum route-dependent queueing delay [10, 11] along R. For M_5,

$$145.3 \text{ ms} < Q_{M_H} < 177.3 \text{ ms}.$$

Note that under stop-and-go, if R had the minimum route-dependent queueing delay (HT) instead of the maximum, the upper and lower delay bounds would be $81.3 + T$ ms and $81.3 - T$ ms. We present a more detailed derivation of the delay bounds for stop-and-go in section D in the appendix.

The queueing disciplines discussed so far are only three of several which provide deterministic upper bounds on queueing delay, subject to the constraint that the sum of the sessions' peak rates does not exceed link capacity. Table 2 shows some of these other queueing disciplines, and their upper bound on delay along R in M_5. With the exception of FCFS, all of these queueing disciplines exhibit a delay bound which grows linearly in the number of hops in our network model. Note also that stop-and-go and WFQ represent the extremes in terms of delay bounds, within this class of queueing disciplines. Zhang and Keshav present a comparison of some of these queueing disciplines in [22].

Queueing discipline	Upper bound on queueing delay (ms)
Weighted Fair Queueing (WFQ) [3, 4, 5, 12, 13]	81.3
Earliest Due Date (EDD) [23][a]	81.3
Hierarchical Round Robin (HRR) [24][b]	161.3
Jitter Earliest Due Date (J-EDD) [25, 26][c]	161.3
Stop-and-go queueing [10, 11]	177.3
First-come first-served (FCFS) [1, 2]	622.5

[a] For EDD, we specify a delay bound of T ms for all sessions at each switch.
[b] For HRR, we assume a frame size of T ms.
[c] For J-EDD, the delay and jitter bounds are $2T$ ms for all sessions at each switch.

Table 2: Upper bounds on queueing delay along R in M_5.

3 End-to-end Delay Distributions

In this section we present and discuss delay distributions under FCFS in an M_5 network (see figure 4) in which all internal links have T1 capacity (C_l = 1536 bits/ms); we also compare the delay distributions obtained with the point-valued worst-case upper bound provided by the three delay-bounding techniques. In the next two subsections we present results for the reduced

OFF period traffic source, and the standard voice source. We follow that in subsection 3.3 with a discussion of the delay distribution seen under Poisson cross traffic.

3.1 Delay Distributions for Reduced OFF Period Source

Figure 6 compares the analytic worst case delay bounds and the simulation results for delay distributions along R in an M_5 network with links carrying 48 sessions ($N = 48$). The curves in this figure (and all other figures in this paper) plot simulation results whose point value exceed twice their 90 % confidence interval halfwidth. Note the dramatic difference in figure 6 between the point at which the actual delay distributions under FCFS become "small" (e.g., smaller than 10^{-5}) and the point at which the various bounding techniques (and their associated multiplexing mechanism) indicate that the delay distribution must be zero. For example, at utilizations up to 98.2 % the 10^{-5} value of the delay distribution obtained via simulation is 12 ms or less, whereas the upper bounds on delay vary between 81.3 ms for WFQ and a 622.5 ms bound for FCFS.

We were also interested in studying how the delay distribution seen by a session changes as it goes "deeper" into the network, as it (and other sessions') traffic characteristics have been altered as a result of multiplexing at upstream switches. Figure 7 shows the end-to-end delay distribution (labeled S_1 thru S_5 in our figures) and the hop-by-hop delay distributions for the reduced OFF period source model at 90 % link utilization. The results are typical for all utilizations we examined in that

1. the distribution at S_1 has greater mass at the tail when compared to the other single hop distributions, and
2. the distributions at $S_2, \ldots, S_5$ are all similar.

The first point can be explained by the infinite capacity of the access links and the traffic smoothing that occurs at the first multiplexing point in the network. The second point is surprising since our network explicitly varies the cross traffic at $S_2, \ldots, S_5$. We had conjectured that these delay distributions would have been more different, since the sessions routed on R were encountering different cross traffic at each hop (note for example, that in figure 4, the upstream subnetworks producing the interfering traffic at each S_j are considerably different).

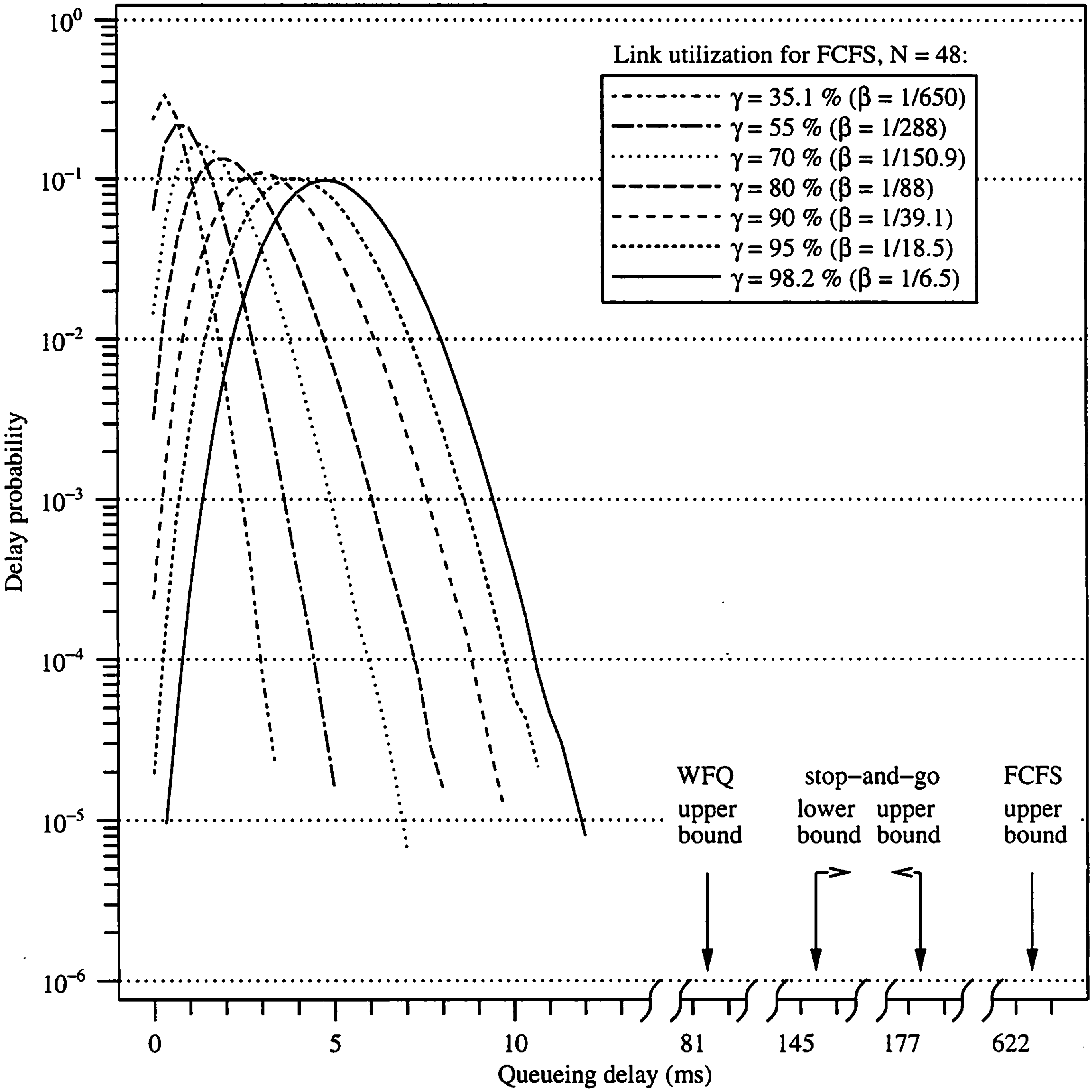

Figure 6: End-to-end queueing delay distributions and bounds for reduced OFF period source in M_5.

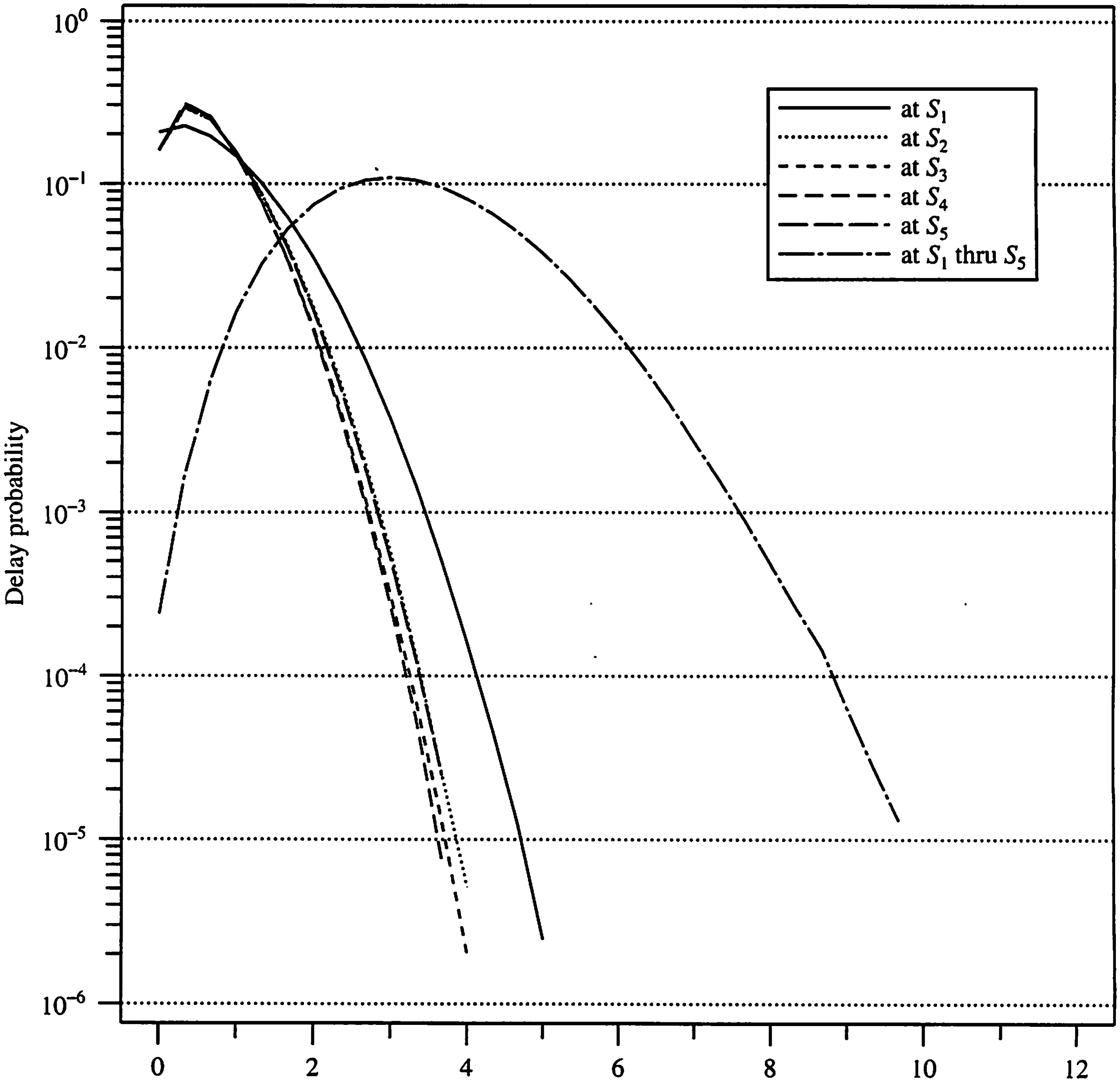

Figure 7: End-to-end and single hop queueing delay distributions for γ = 90 % using reduced OFF period source in M_5.

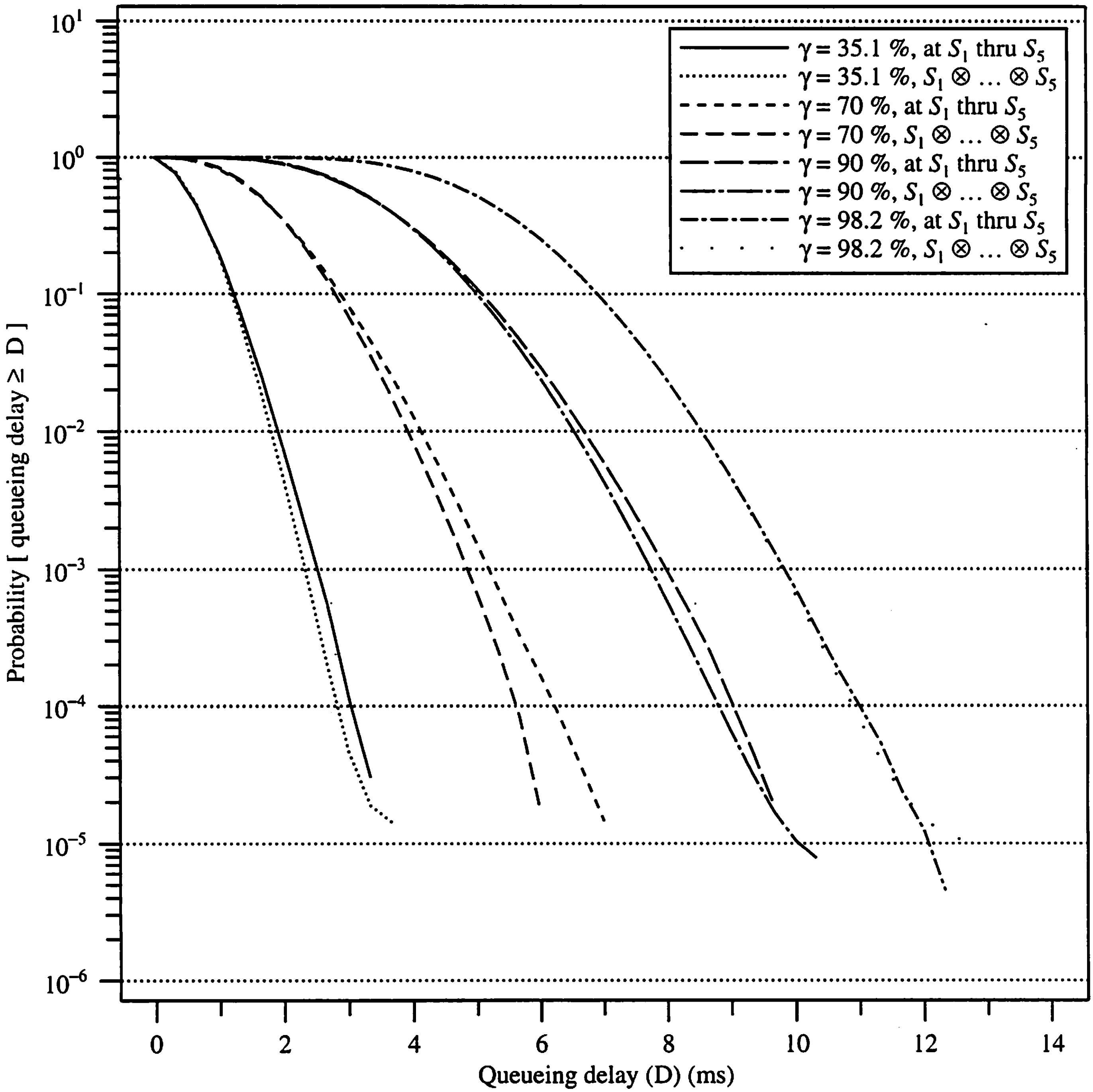

Figure 8: Queueing delay tail distributions – measured end-to-end and computed by convolution for reduced OFF period source.

We suspect that the traffic smoothing that occurs at all initial multiplexing points is so dramatic that the cross traffic at these switches is almost homogeneous.

Because of the similarity of the single hop distributions at $S_2, \ldots, S_5$ for the reduced OFF period source model, we compared an estimate of the end-to-end delay distribution obtained by convolving the five single hop distributions with the end-to-end delay distributions measured (via simulation). Figure 8 shows this comparison for several values of link utilization (the convolutions are labeled $S_1 \otimes \ldots \otimes S_5$). We observe that the distributions obtained by convolving the single hop distributions are close to the observed end-to-end delay distributions. This observation is consistent with the hypothesis that the single hop delays incurred by a packet are independent random variables, particularly when the link utilizations are high. The differences are only up to a factor of six for a given value of end-to-end delay at low and moderate link utilizations. From a practical standpoint, this suggests that a call admission procedure which admits calls based on observed delays (e.g., [15, 16]) need only monitor the individual single-hop (local) delays in order to estimate the end-to-end delay distribution.

3.2 Delay Distributions for Standard Voice Source

Recall that in the previous subsection, traffic intensity (equivalently, link utilization) was varied by decreasing the expected length of the sources' silence period. This resulted in a situation in which the sum of the peak rates (calculated over a 16 ms interval) of all sources being multiplexed over any link in the network was less than that link's capacity. Given this condition, it was possible to analytically compute worst case, point-valued delay bounds using the methods of [1, 2, 3, 4, 5, 10, 11].

In this section, we vary the link utilization by increasing the number of active sources. Since this results in a situation in which the sum of the peak rates (calculated over a 16 ms interval) of all sources being multiplexed over any link in the network may now *exceed* that link's capacity, *no* bounds can be computed.

Figure 9 shows delay distributions along R under FCFS with links carrying 48 or more active sessions ($N \geq 48$). Buffers in the network are sized so that the delay at each hop along R is less than 234 ms. We chose such a large buffer size so that are delay distributions at moderate utilizations were not biased by packet loss. For the delay distributions shown, no loss

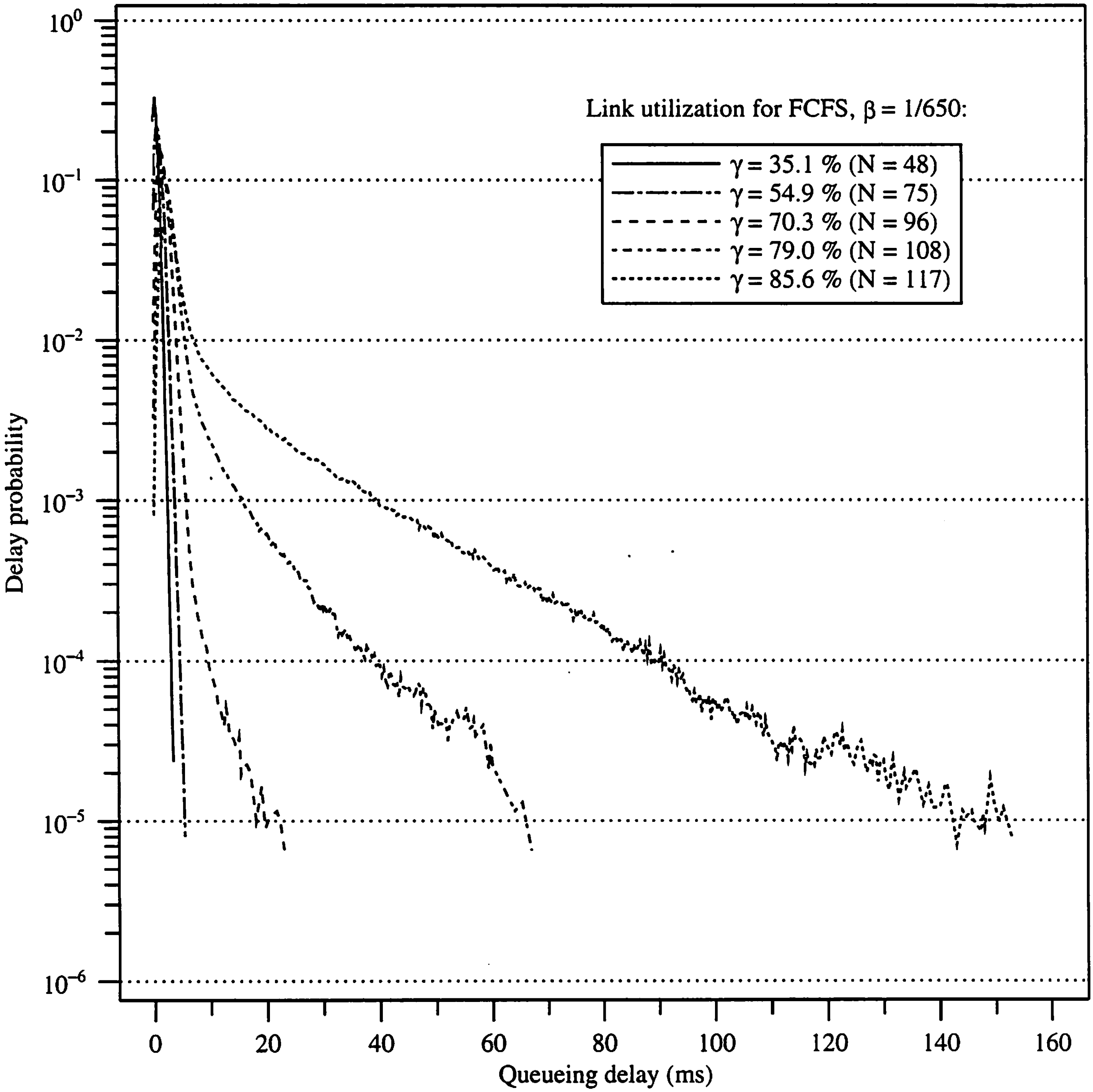

Figure 9: End-to-end queueing delay distributions for standard voice source in M_5.

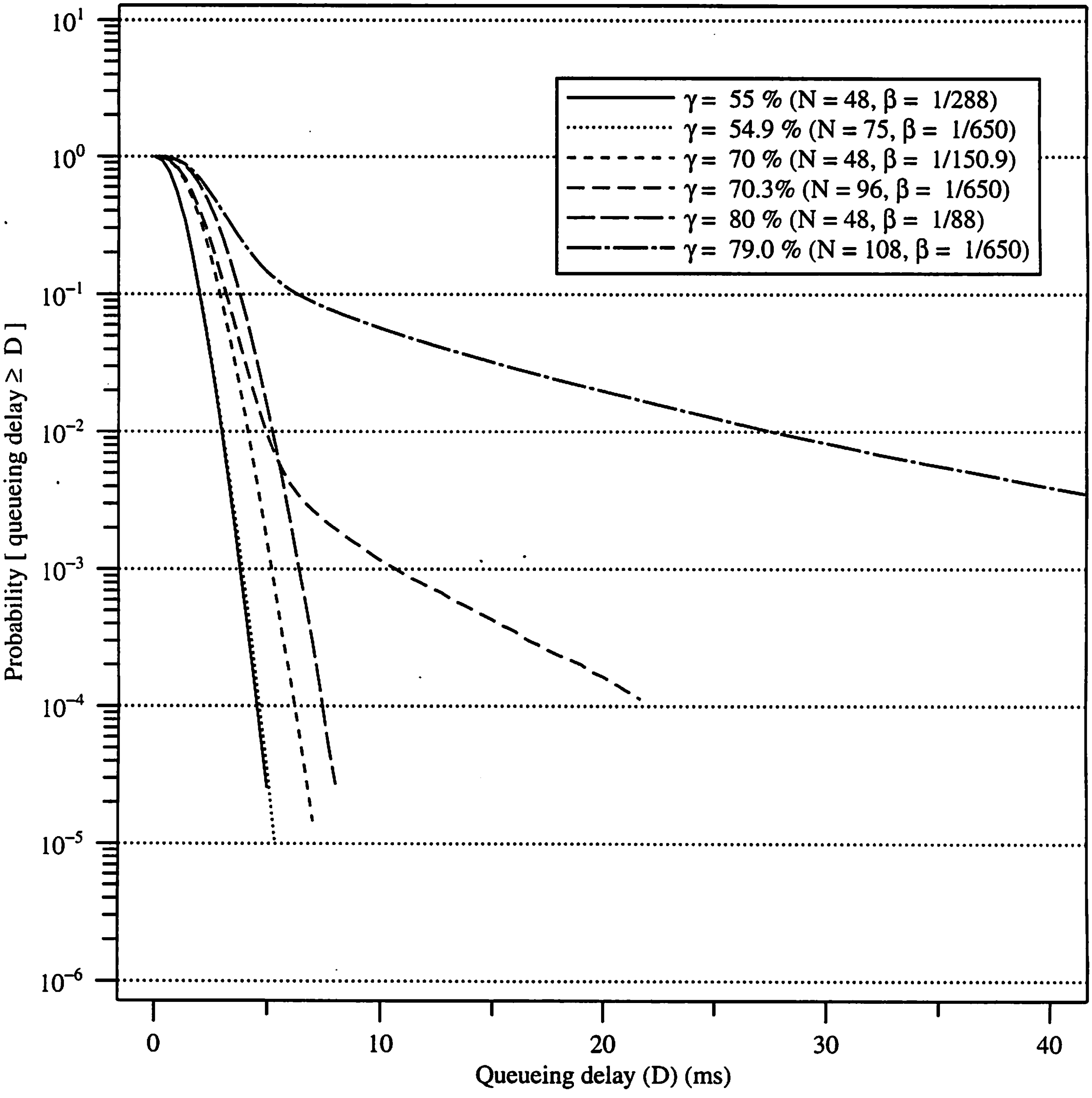

Figure 10: Comparison of end-to-end queueing delay tail distributions for reduced OFF period and standard voice source models.

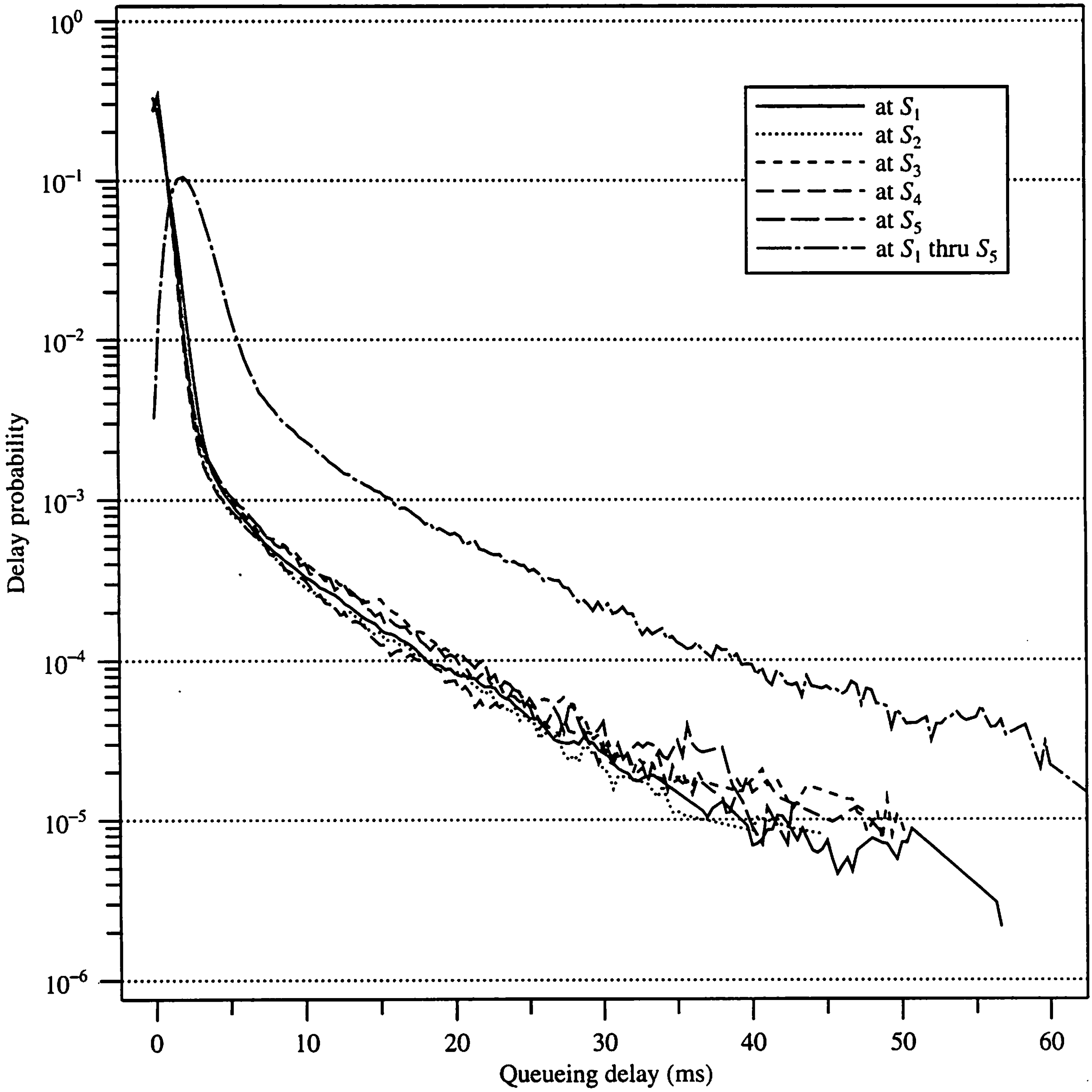

Figure 11: End-to-end and single hop queueing delay distributions for γ = 79.0 % using standard voice source in M_5.

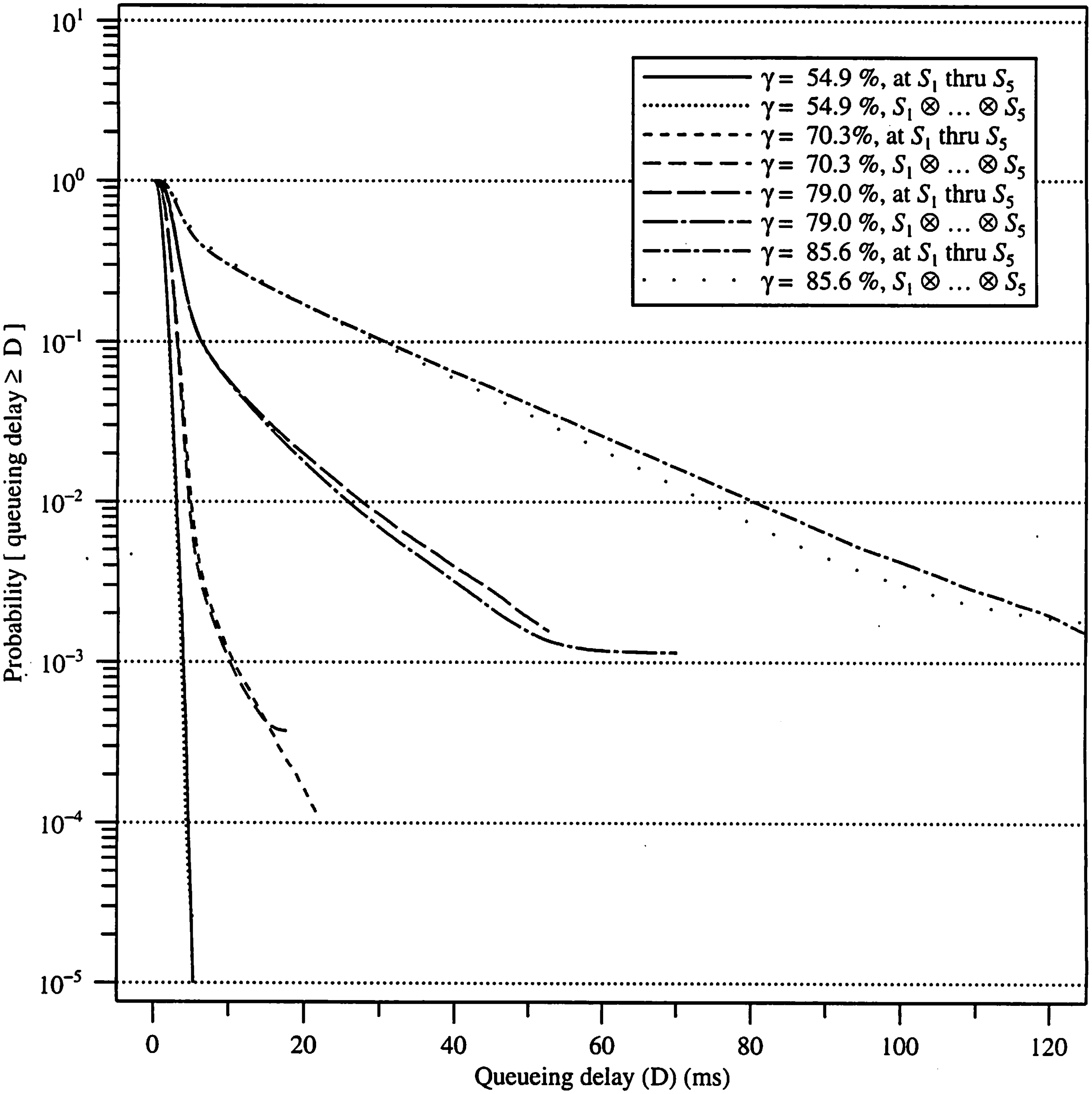

Figure 12: Queueing delay tail distributions – measured end-to-end and computed by convolution for standard voice source.

occurred, however, at 90.0 % utilization a loss probability of 2.4×10^{-7} was observed over 10 independent simulation runs. Note that the criteria for plotting points obtained by simulation described in subsection 3.1 reveals noise at the tail of the distributions shown. Rather than clip our data, we felt it more important to show the magnitude of low probability delays in this figure.

Note that the FCFS delay distributions along route R in M_5 have changed dramatically from the case of the reduced OFF period model. At moderate and high utilizations the delay distributions in figure 9 extend beyond those in figure 6 by more than an order of magnitude. Figure 10 directly compares the delay distributions along R at similar utilizations for the reduced OFF period and standard voice source models. At moderate utilizations the tails of the distributions are comparable within a factor of two for a given value of end-to-end delay. However, at high utilizations (70 % and greater) the distributions differ by up to almost four orders of magnitude.

We also examined how the delay distribution seen by a session changes as it gets "deeper" into the network for the standard voice source model. Figure 11 shows the hop-by-hop delay distributions and the end-to-end delay distribution at 79 % utilization. In contrast to the results for the reduced OFF period source (see figure 7), the single hop distributions at S_1 through S_5 are similar at utilizations of 70.3 % and above. At a utilization of 54.9 %, the S_1 distribution has slightly greater mass at the tail when compared to the other single hop distributions.

We also compared delay distributions measured end-to-end and computed by the convolution of single hop distributions for several values of γ using the standard voice source model. These results, shown in figure 12, are also consistent with the hypothesis that single hop delays incurred by a packet are independent of each other in that the measured distribution and the convolution are always within a factor of two.

3.3 Discussion of Cross Traffic

In this paper we have directly simulated smaller networks, using their output packets as interfering traffic to the route R under study. Others have modeled cross traffic by a Poisson process [27], an MMPP [21], a Bernoulli process with batch arrivals [28], and the superposition of several interrupted Poisson processes [29]. As pointed out in [20] and [29], using a Poisson process to model cross traffic is inappropriate for future integrated services networks. This is because the traffic exhibits long term correlation, especially at high loads.

To see how our cross traffic model compared with Poisson cross traffic, we used the five hop tandem queue network shown in figure 13. The leftmost queue in this figure models the output MUX in the first switch (S_1) along route R'. The remaining queues model the output MUX's in switches S_2 through S_5, where one input is from the previous switch along R' and two inputs are from Poisson processes with parameter $\lambda = \gamma C_l/(3L)$ packets/ms. As in our previous model, only a third of the sessions departing S_i on R' are routed to the next output MUX shown in S_{i+1}. Each Poisson process in this network emits cross traffic on behalf of $N/3$ sessions on "access links" of infinite capacity.

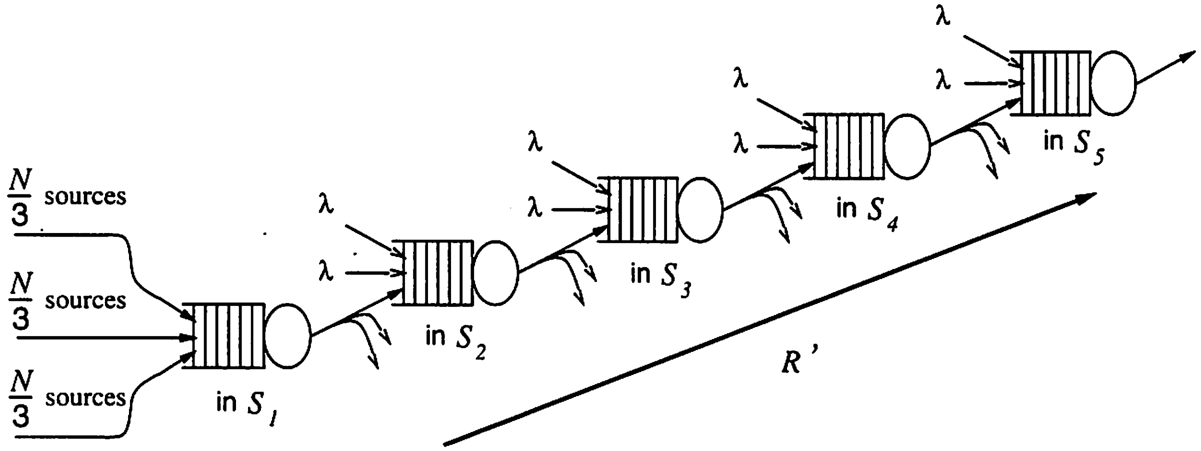

Figure 13: Five hop tandem queue network with Poisson cross traffic.

Figure 14 compares delay distributions under FCFS along R in M_5 (with $M_1, \ldots, M_4$ cross traffic) and along R' with Poisson cross traffic in networks with links carrying 48 sessions fed by reduced OFF period sources. The Poisson model works well at light link utilizations, but gives tail probabilities that are off by over three orders of magnitude for a given value of end-to-end delay at 90 % utilization. Note that the delay distribution for Poisson cross traffic always has greater mass at the tail than for $M_1, \ldots, M_4$ cross traffic.

Figure 15 compares delay distributions under FCFS along R in M_5 and along R' with Poisson cross traffic in networks with sessions fed by standard voice sources. The Poisson model gives

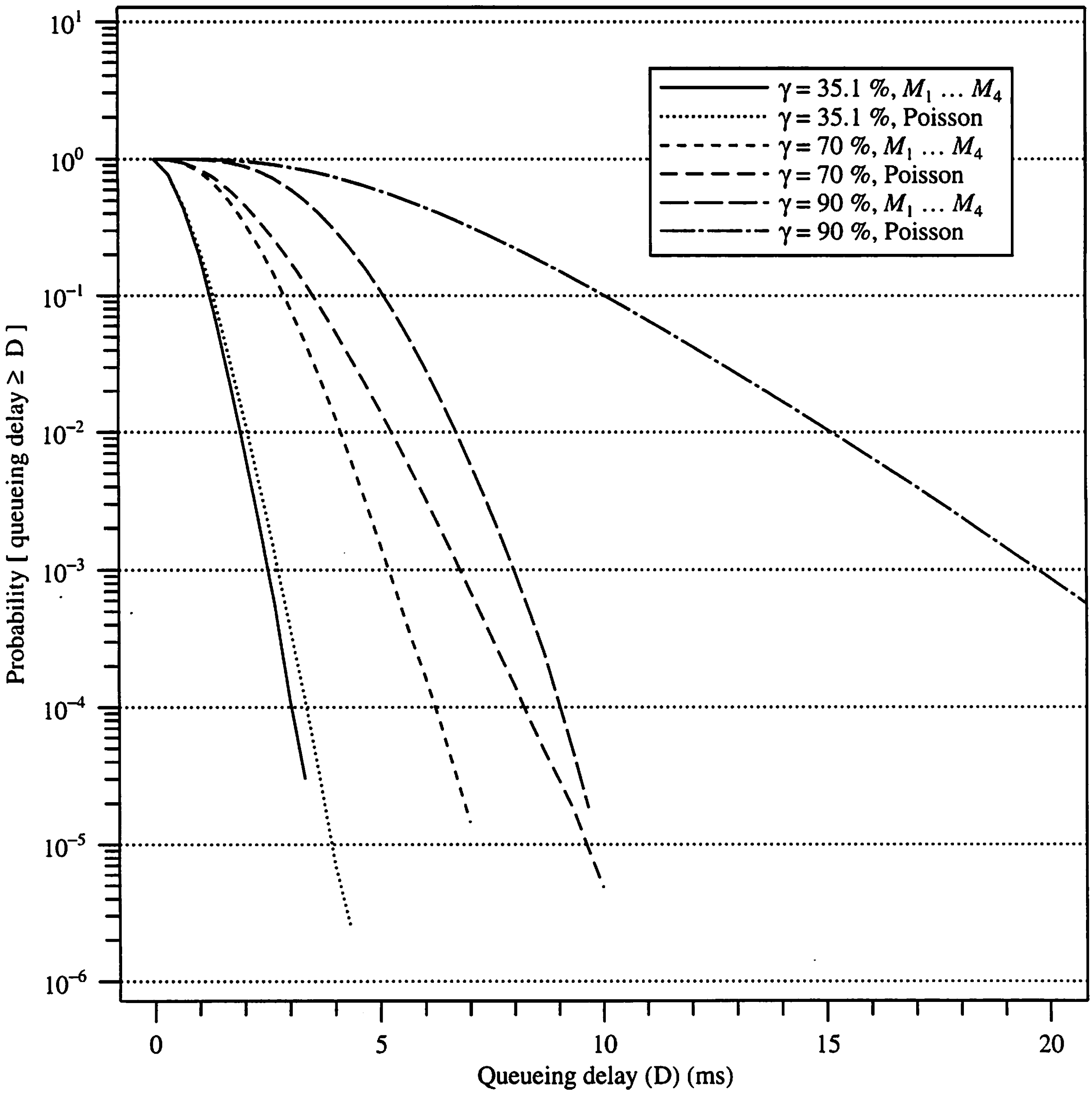

Figure 14: End-to-end queueing delay tail distributions for $M_1, \ldots, M_4$ cross traffic and Poisson cross traffic using reduced OFF period source.

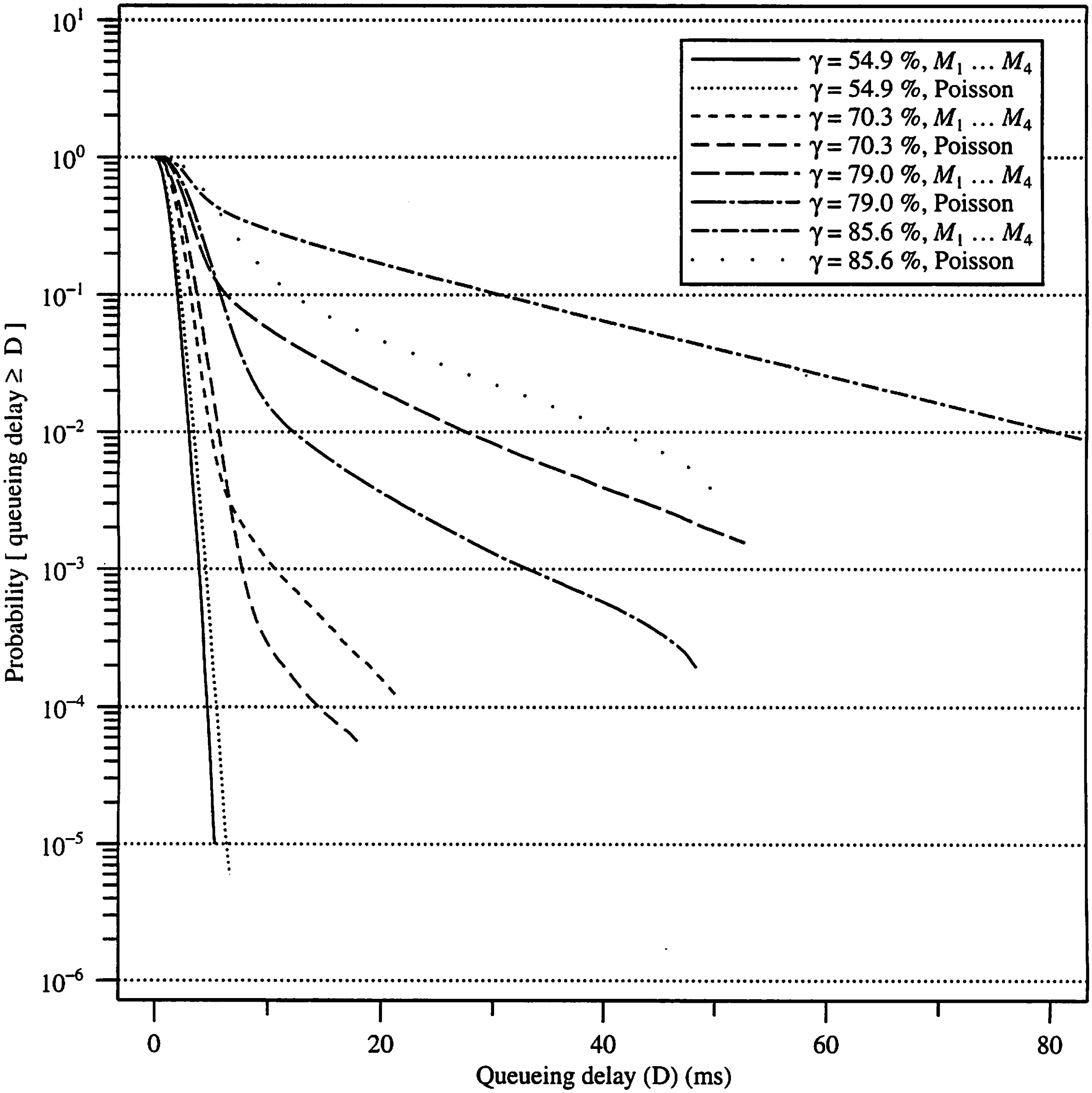

Figure 15: End-to-end queueing delay tail distributions for $M_1, \ldots, M_4$ cross traffic and Poisson cross traffic using standard voice source.

tail probabilities that are off by up to an order of magnitude, depending on the utilization. In contrast with figure 14, at moderate and high utilizations the delay distribution for Poisson cross traffic has smaller mass at the tail than for $M_1, \ldots, M_4$ cross traffic.

4 QOS and Call Admission Control

As mentioned earlier, if true delay distributions under FCFS were known (or could be accurately estimated), a call admission procedure could support more calls requiring soft real-time service (e.g., packetized voice) than if the application had "hard" real-time requirements (e.g., all packets having a delay less than a deterministic bound). So far, our results have given end-to-end queueing delay distributions under FCFS multiplexing in M_5 for two source models. We have also derived worst case analytic queueing delay bounds for FCFS, stop-and-go, and WFQ along a fixed route in M_5. We now revisit these results from the perspective of performing call admission.

Figures 16 and 17 show previously discussed results from a different perspective in order to illustrate the effect of the level of QOS required (and the manner in which it is guaranteed) on the number of calls that can be accepted into the M_5 network (equivalently, link utilization by accepted calls).

Recall that soft real-time applications are "loss tolerant" in the sense that a certain amount of packet loss due to excessive delay is considered acceptable. Each of the six plotted curves in figure 16 is for a constant fixed "acceptable" packet loss probability (at the receiver) due to excessive end-to-end delay (referred to as y in the figure legend) and a given method for computing the delay distribution guarantee (three methods which provide an upper bound on the distribution and three curves derived from our simulation results). Each curve plots the link utilization at which the end-to-end constraint along route R

$$\text{Probability[queueing delay} \geq \text{D]} = y,$$

is satisfied.

First consider figure 16. Our results indicate, for example, that if a call arrives for route R with the requirement that *no* packet be delayed by queueing more than 200 ms, it can be accepted under stop-and-go and WFQ, but not under FCFS, providing that the resulting

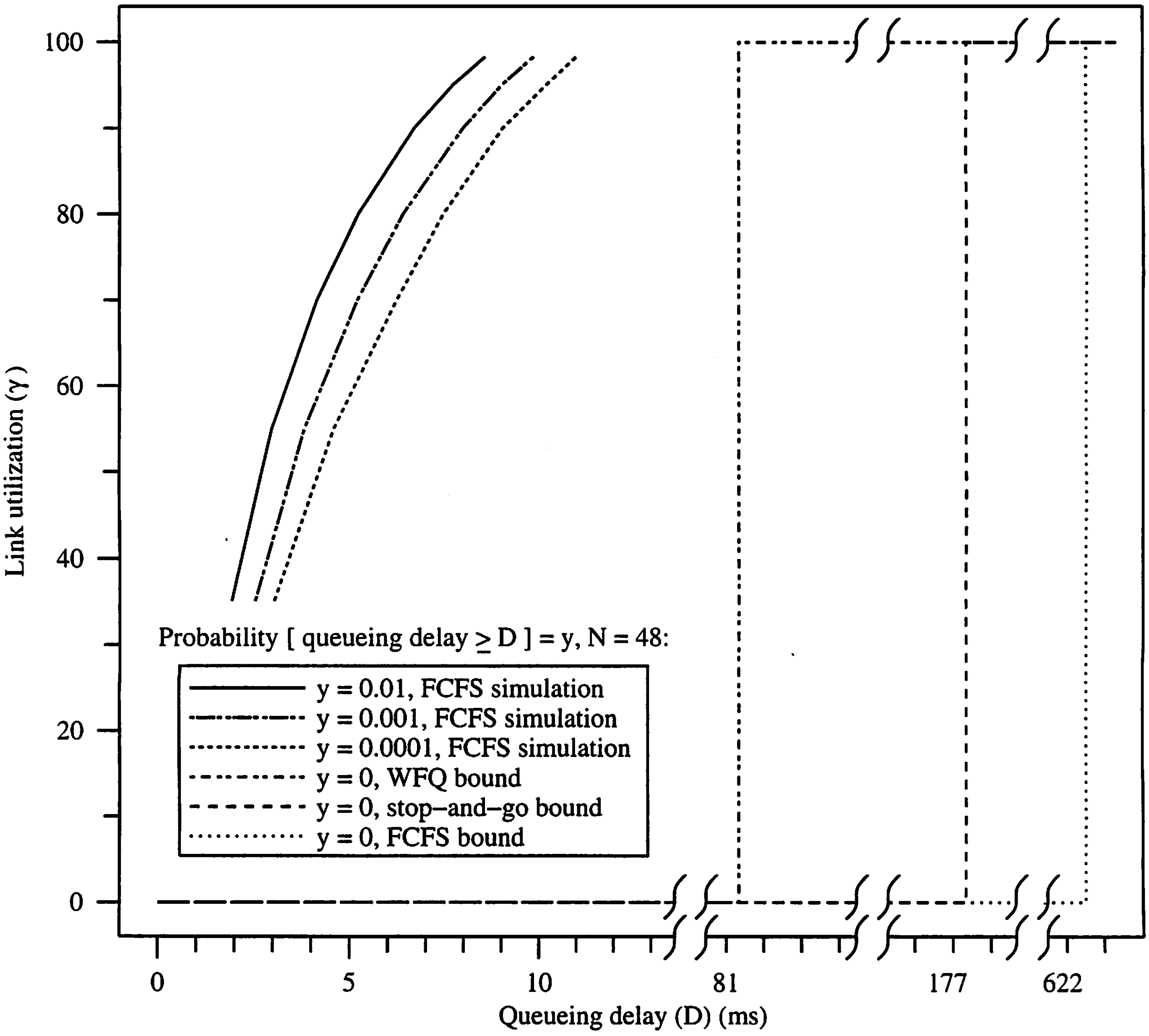

Figure 16: Link utilization supported versus delay/loss constraint (specified by D and y) for reduced OFF period source in M_5.

link utilization would be less than 100 %. If an arriving call requires that 99 % of packets be delayed less than 6 ms, then a call admission procedure which had knowledge of (or could accurately estimate) the actual delay distribution would accept such a call provided that the link utilization would be less than 85 %. A call admission procedure which used point-valued, worst-case bounds could not accept such a call.

Next, consider figure 17. A call requiring a deterministic 200 ms delay bound could be supported by stop-and-go and WFQ if there are less than 48 calls on all links along R. This corresponds to a link utilization of less than 35 %. If we were willing to relax this constraint to be that 99.9 % of the packets have a delay bound less than 200 ms, both queueing disciplines could still only accept up to 48 calls/link. Since the point-valued FCFS delay bound is 622.5 ms (not shown in the figure), no calls can be accepted if they have a delay requirement less than this value. However, once the delay bound is above this value, FCFS, like stop-and-go and WFQ can support 48 calls.

Simulation results are also shown in figure 17 for FCFS, and indicate that more than 115 calls could in fact be supported with this requirement that only 0.1 % of the packets have a delay in excess of 200 ms. If an arriving call requires that 99.9 % of packets must be delayed less than 40 ms, our simulation results indicate that FCFS can support the call as long as links along R are currently supporting less than 100 calls; a call admission policy based on worst case bounds would admit no calls at this guaranteed quality of service.

Figure 17 also plots the number of supportable calls under an admission control procedure based on the equivalent capacity formulation of [17, 18]. In this approach, the traffic characteristics of a source are assumed to remain unchanged as it passes through the network. Hence the calculated bandwidth required to meet a given QOS measure (in our case, the requirement that no more than a certain fraction of the packets have a delay exceeding a given value), are, in some sense, "approximate" performance guarantees.

The equivalent capacity approach itself provides two methods for computing the number of supportable calls. For the scenario considered here, the so-called stationary approximation (labeled FCFS EC Gaussian in figure 17) indicates that up to 87 calls can be supported with less that 0.1 % of the packets having non-zero end-to-end queueing delay. This gives rise to the flat equivalent capacity curve for 0.1 % packet loss. Beyond 87 calls, the so-called fluid flow approximation (labeled FCFS EC fluid in the figure) gives rise to the slightly increasing

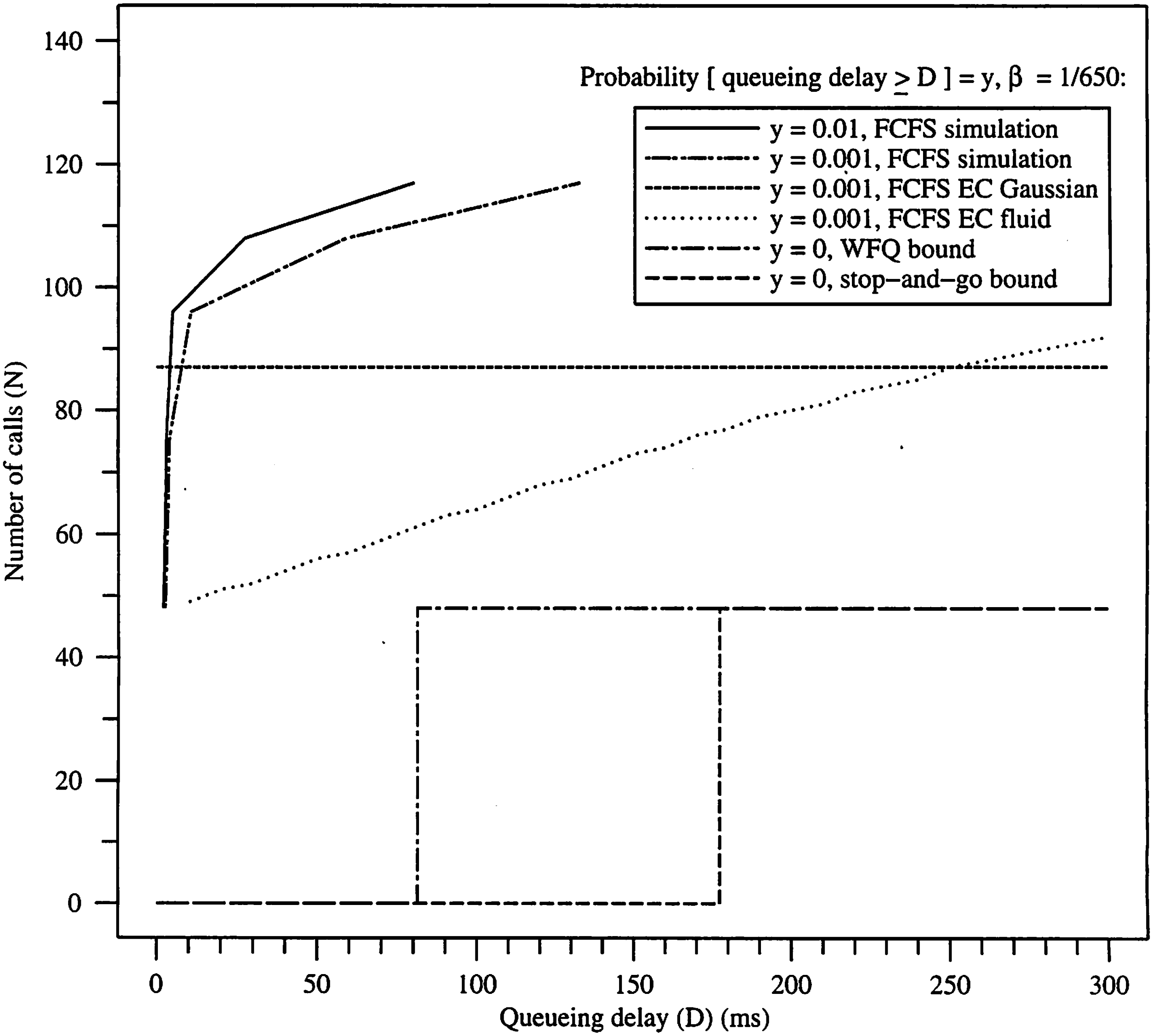

Figure 17: Number of calls supported versus delay/loss constraint (specified by D and y) for standard voice source in M_5.

equivalent capacity beyond 250 ms. The equivalent capacity curves for a loss of 1 % were not found to be significantly different from the 0.1 % curves shown in the figure. Details of the equivalent capacity calculations can be found in section E in the appendix.

We note that the equivalent capacity call admission scheme allows a much larger number of calls to be supported than under an admission scheme based on using point-valued bounds, as one would expect. The equivalent capacity curves can also be seen to be generally conservative estimates of the number of supportable calls indicated by our simulation results. However, this is not always the case – for example, when the delay bounds are extremely tight.

For soft real-time applications, the number of admissible calls even when the loss constraint is fairly stringent (e.g., 10^{-4}) is significantly larger than the number of calls admissible when point-valued worst case bounds on the distribution are used. This is perhaps not surprising since the worst case bounds are indeed quite stringent – they require that *all* packets have a delay less than the given bound. This difference suggests that a so-called observation-based approach to providing QOS guarantees might be effectively used to provide QOS guarantees for soft real-time traffic. In [15, 30] the authors describe measuring maximum delays over fixed intervals in time, and filtering these measurements using an exponential average. Figure 8 suggests that measuring single hop delay distributions, and convolving them to obtain end-to-end distributions might be a promising approach to providing approximations for different grades of QOS (in terms of delay).

We also note that recent research has addressed the possibility of providing *distributional bounds* on per-session delay in a network setting [6, 7, 8, 9]. So far this work has relied on fairly complex source characterizations (as opposed to the much simpler traffic characteristics which we have used to compute point-valued worst case delay bounds). Also it appears difficult at this point to apply these techniques to the call admission problem when any set of routes in a network form a cycle.

5 Conclusion

In this paper we have studied (through simulation) the end-to-end delay distribution seen by individual sessions under simple FCFS multiplexing in a connection-oriented network model. We compared these delay distributions with the worst case point-valued analytic delay bounds predicted by three different techniques for providing such bounds (two of which require a more sophisticated link-level scheduling policy). We also considered the per-hop delay distributions seen as a session progresses "deeper" into the network and determined the sensitivity of these delay distributions to the manner in which the interfering traffic is modeled. Finally, we used our delay distribution results to examine the tradeoff between the QOS requested by a call, the manner in which the QOS guarantee is provided, and the number of calls that are admitted at the requested QOS.

The drawback with using point-valued bounds to guide a call admission policy for soft real-time traffic is that the worst case bounds must, by definition, consider delays resulting from scenarios which may be extremely unlikely (e.g., all network sources being simultaneously active in a given packet-transmission time). For our network model, we conjecture that the FCFS worst case bound is not achievable. However the bounds for stop-and-go and WFQ are provably achievable to within an arbitrarily small tolerance [5, 11]. When using our reduced OFF period source, at high link utilizations the delay distributions for FCFS fall substantially below all bounds on delay. Thus, the delay distribution observed under FCFS is better than the delay guaranteed by stop-and-go and WFQ, *and* the best possible delay under stop-and-go.

A second potential problem with deterministic guarantees is that the necessary conditions to provide the guarantees (e.g., peak link utilization not exceeding link capacity) may be unrealistic in a real network. For instance, using the standard voice source in our network model, delay bounds can not be computed for link utilizations greater than 35.1 %. Leaky bucket control of the sources can ameliorate this situation but themselves add an additional component into the end-to-end delay.

Statistical QOS guarantees allow a user to select between several grades of QOS, each with an associated probability that the QOS will be met during the call. We have focused on statistical and deterministic QOS in terms of delay in this paper, as has work presented in [23, 31]. Other researchers have examined loss as a performance metric [17, 32, 33, 34, 35]. Users may also find it useful to adjust other QOS parameters such as throughput [36] and delay jitter [11, 25]. Providing such flexibility poses a considerable challenge in developing queueing disciplines and call admission procedures which together provide for statistical QOS guarantees.

Acknowledgments

The authors would like to thank Ken Vastola, Sugih Jamin, and the members of the Computer Networks and Performance Evaluation Laboratory at the University of Massachusetts, and especially Ramesh Nagarajan, for improving the ideas presented in this paper.

References

[1] R. L. Cruz, "A calculus for network delay, part I: Network elements in isolation," *IEEE Transactions on Information Theory*, vol. 37, pp. 114–131, Jan. 1991.

[2] R. L. Cruz, "A calculus for network delay, part II: Network analysis," *IEEE Transactions on Information Theory*, vol. 37, pp. 132–141, Jan. 1991.

[3] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks — the single node case," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, vol. 2, (Florence, Italy), pp. 915–924 (7A.3), IEEE, May 1992.

[4] A. K. J. Parekh, *A generalized processor sharing approach to flow control in integrated services networks.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, Feb. 1992.

[5] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks — the multiple node case," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, vol. 2, (San Francisco, CA), pp. 521–530 (5A.1), IEEE, Mar. 1993.

[6] J. Kurose, "On computing per-session performance bounds in high-speed multi-hop computer networks," in *Sigmetrics 1992*, (New Port, Rhode Island), pp. 128–139, ACM, June 1992.

[7] C.-S. Chang, "Stability, queue length, and delay, part I: Deterministic queueing networks," Research Report RC 17708, IBM T. J. Watson Research Center, Yorktown Heights, NY, Feb. 1992.

[8] C.-S. Chang, "Stability, queue length, and delay, part II: Stochastic queueing networks," Research Report RC 17709, IBM T. J. Watson Research Center, Yorktown Heights, NY, Feb. 1992.

[9] O. Yaron and M. Sidi, "Calculating performance bounds in communication networks," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, vol. 2, (San Francisco, CA), pp. 539–546 (5A.3), IEEE, Mar. 1993.

[10] S. J. Golestani, "A framing strategy for congestion management," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 1064–1077, Sept. 1991.

[11] S. J. Golestani, "Congestion-free communication in high-speed packet networks," *IEEE Transactions on Communications*, vol. 39, pp. 1802–1812, Dec. 1991.

[12] A. G. Greenberg and N. Madras, "Comparison of a fair queueing discipline to processor sharing," in *Performance '90: 14th International Symposium on Computer Performance Modelling, Measurement and Evaluation* (P. J. B. King, I. Mitrani, and R. J. Pooley, eds.), (Edinburgh, Scotland), pp. 193–207, IFIP WG7.3, North-Holland, Amsterdam, Holland, Sept. 1990.

[13] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," *Internetworking: Research and Experience*, vol. 1, pp. 3–26, Jan. 1990.

[14] J. Kurose, "Open issues and challenges in providing quality of service guarantees in high-speed networks," *ACM Computer Communication Review*, vol. 23, pp. 6–15, Jan. 1993.

[15] S. Jamin, S. Shenker, L. Zhang, and D. D. Clark, "An admission control algorithm for predictive real-time service (extended abstract)," in *Proceedings of the Third International Workshop on Network and Operating System Support for Digital Audio and Video*, (San Diego, CA), pp. 308–314, IEEE, Nov. 1992.

[16] T. E. Tedijanto and L. Gün, "Effectiveness of dynamic bandwidth management mechanisms in ATM networks," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (San Francisco, CA), pp. 358–367 (3d.2), IEEE, Mar. 1993.

[17] R. Guérin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 968–981, Sept. 1991.

[18] R. Guérin and L. Gün, "A unified approach to bandwidth allocation and access control in fast packet-switched networks," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, vol. 1, (Florence, Italy), pp. 1–12 (1A.1), IEEE, May 1992.

[19] P. T. Brady, "A statistical analysis of on-off patterns in 16 conversations," *Bell System Technical Journal*, vol. 47, pp. 73–91, Jan. 1968.

[20] K. Sriram and W. Whitt, "Characterizing superposition arrival processes in packet multiplexers for voice and data," *IEEE Journal on Selected Areas in Communications*, vol. SAC-4, pp. 833–846, Sept. 1986.

[21] H. Heffes and D. M. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE Journal on Selected Areas in Communications*, vol. SAC-4, pp. 856–868, Sept. 1986.

[22] H. Zhang and S. Keshav, "Comparison of rate-based service disciplines," in *Sigcomm '91 Symposium – Communications Architectures and Protocols*, (Zurich, Switzerland), pp. 113–121, ACM, Sept. 1991.

[23] D. Ferrari and D. C. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 368–379, Apr. 1990.

[24] C. R. Kalmanek, H. Kanakia, and S. Keshav, "Rate controlled servers for very high-speed networks," in *Proceedings of the Conference on Global Communications (GLOBECOM)*, (San Diego, CA), pp. 12–20 (300.3), IEEE, Dec. 1990.

[25] D. C. Verma, H. Zhang, and D. Ferrari, "Delay jitter control for real-time communication in a packet switching network," in *Proceedings of Tricomm '91*, (Chapel Hill, NC), pp. 35–43, IEEE, Apr. 1991.

[26] D. Ferrari, "Distributed delay jitter control in packet-switching internetworks," *Internetworking: Research and Experience*, vol. 4, pp. 1–20, Jan. 1993.

[27] D. Mitra, "Optimal design of windows for high speed data networks," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (San Francisco, CA), pp. 1156–1163, IEEE, June 1990.

[28] M. Murata, Y. Oie, T. Suda, and H. Miyahara, "Analysis of a discrete-time single-server queue with bursty inputs for traffic control in ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 447–458, Apr. 1990.

[29] Y. Ohba, M. Murata, and H. Miyahara, "Analysis of interdeparture processes for bursty traffic in ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 468–476, Apr. 1991.

[30] D. D. Clark, S. Shenker, and L. Zhang, "Supporting real-time applications in an integrated services packet network: architecture and mechanism," in *SIGCOMM Symposium on Communications Architectures and Protocols*, (Baltimore, MD), pp. 14–26, ACM, Aug. 1992.

[31] G. M. Woodruff and R. Kositpaiboon, "Multimedia traffic management principles for guaranteed ATM network performance," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 437–446, Apr. 1990.

[32] T. Kamitake and T. Suda, "Evaluation of an admission control scheme for an ATM network considering fluctuations in cell loss rate," in *Proceedings of the Conference on Global Communications (GLOBECOM)*, (Dallas, Texas), pp. 1774–1780, IEEE, Nov. 1989.

[33] C. Rasmussen and J. Sorensen, "A simple call acceptance procedure in an ATM network," *Computer Networks and ISDN Systems*, vol. 20, pp. 197–202, Dec. 1990. ITC Specialist Seminar, 25–29 September 1989, Adelaide, Australia.

[34] H. Saito and K. Shiomoto, "Dynamic call admission control in ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 982–989, Sept. 1991.

[35] A. I. Elwalid and D. Mitra, "Effective bandwidth of general markovian traffic sources and admission control of high-speed networks," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 329–343, June 1993.

[36] L. Zhang, "VirtualClock: A new traffic control algorithm for packet-switched networks," *ACM Transactions on Computer Systems*, vol. 9, pp. 101–124, May 1991.

[37] H. Schulzrinne, J. F. Kurose, and D. Towsley, "Congestion control for real-time traffic in high-speed networks," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (San Francisco, CA), pp. 543–550, June 1990.

[38] R. Nagarajan, J. F. Kurose, and D. Towsley, "Allocation of local quality of service constraints to meet end-to-end requirements," *IFIP Transactions C-15, Modeling and Performance Evaluation of ATM Technology*, pp. 98–118, 1993.

Appendix

A Routing Strategy for M_5 Network

As discussed in subsection 2.1, the routing strategy used in an M_H network when the number of sessions carried by each link (N) is finite impacts the length of sessions which introduce cross traffic along R. This section describes the fixed routing strategy used in the M_5 network for which we present results (see figure 4).

In the entire M_5 network, we use only six different routing tables (which contain mappings from input link and session number to output link and session number). Both input links and output links are numbered from one to three. For simplicity, we only describe routing assuming $N = 48$. For $N > 48$, fixed routes are reused arbitrarily. For the remainder of this section, we refer to M_1 and M_2 as "small" (sub)networks, and M_3, M_4 and M_5 as "big" (sub)networks.

In small M_H (sub)networks, we employ only half of the six routing tables used in M_5. Each of these routing tables corresponds to an output link and has a regular structure. The routing table for output link one maps the $1st, 4th, 7th, \ldots, 46th$ session from each input link to sessions 1 through 48 in increasing order. The table for output link two maps the $2nd, 5th, 8th, \ldots, 47th$ sessions in a similar manner. Output link three carries the remaining input sessions, also mapped in order. As an example, table 3 shows the routing table for output link two.

In big M_H (sub)networks, we employ the remaining three routing tables at the output links of the intermediate switches along R (in S_j, where $1 < j < H$). These routing tables have an irregular structure, which is chosen to maximize the number of sessions which traverse H hops along R in M_H (four in M_5). For convenience, switches that are present in more than one big (sub)network (e.g., S_2 appears in M_3, M_4, and M_5) use the same routing table in all big (sub)networks. For the other switches along R (S_1 and S_H), we use the same routing tables as used in the small (sub)networks. Tables 4 through 6 shows the three routing tables that complete our description of the routing in big (sub)networks, including M_5 (S_2 in M_3, M_4 and M_5; S_3 in M_4 and M_5; and S_4 in M_5).

To illustrate how the particular routing strategy described above impacts the length of sessions which intersect with R, we focus on the last switch along R in M_5 (S_5). Table 7 shows a comparison of the fraction of sessions which depart S_5 in an M_5 network where $N = \infty$ with an M_5 network where $N = 48$.

Output Link Session	(Input link, Session)
1	(1,2)
2	(2,2)
3	(3,2)
4	(1,5)
5	(2,5)
6	(3,5)
7	(1,8)
8	(2,8)
9	(3,8)
⋮	⋮
46	(1,47)
47	(2,47)
48	(3,47)

Table 3: Routing table for output link two in "small" (sub)networks.

Output Link Session	(Input link, Session)	Output Link Session	(Input link, Session)	Output Link Session	(Input link, Session)
1	(1,1)	17	(1,16)	33	(2,31)
2	(2,1)	18	(3,16)	34	(2,34)
3	(3,1)	19	(1,19)	35	(1,34)
4	(2,4)	20	(2,19)	36	(3,34)
5	(3,4)	21	(3,19)	37	(1,37)
6	(1,4)	22	(2,22)	38	(2,37)
7	(3,7)	23	(3,22)	39	(3,37)
8	(1,7)	24	(1,22)	40	(2,40)
9	(2,7)	25	(3,25)	41	(3,40)
10	(3,10)	26	(1,25)	42	(1,40)
11	(2,10)	27	(2,25)	43	(3,43)
12	(1,10)	28	(3,28)	44	(1,43)
13	(1,13)	29	(2,28)	45	(2,43)
14	(3,13)	30	(1,28)	46	(3,46)
15	(2,13)	31	(1,31)	47	(2,46)
16	(2,16)	32	(3,31)	48	(1,46)

Table 4: Routing table for S_2 in "big" (sub)networks.

Output Link	(Input link,	Output Link	(Input link,	Output Link	(Input link,
Session	Session)	Session	Session)	Session	Session)
1	(1,1)	17	(1,16)	33	(2,31)
2	(2,1)	18	(3,16)	34	(2,34)
3	(3,1)	19	(1,19)	35	(1,34)
4	(2,4)	20	(2,19)	36	(3,34)
5	(3,4)	21	(3,19)	37	(1,37)
6	(1,4)	22	(2,22)	38	(2,37)
7	(3,7)	23	(3,22)	39	(3,37)
8	(1,7)	24	(1,22)	40	(2,40)
9	(2,7)	25	(3,25)	41	(3,40)
10	(3,10)	26	(1,25)	42	(1,40)
11	(2,10)	27	(2,25)	43	(1,43)
12	(1,10)	28	(3,28)	44	(3,43)
13	(1,13)	29	(2,28)	45	(2,43)
14	(3,13)	30	(1,28)	46	(2,46)
15	(2,13)	31	(1,31)	47	(1,46)
16	(2,16)	32	(3,31)	48	(3,46)

Table 5: Routing table for S_3 in M_4 and M_5.

Output Link	(Input link,	Output Link	(Input link,	Output Link	(Input link,
Session	Session)	Session	Session)	Session	Session)
1	(1,1)	17	(1,16)	33	(2,31)
2	(2,1)	18	(3,16)	34	(2,34)
3	(3,1)	19	(1,19)	35	(1,34)
4	(2,4)	20	(2,19)	36	(3,34)
5	(3,4)	21	(3,19)	37	(1,37)
6	(1,4)	22	(2,22)	38	(2,37)
7	(3,7)	23	(3,22)	39	(3,37)
8	(1,7)	24	(1,22)	40	(2,40)
9	(2,7)	25	(1,25)	41	(3,40)
10	(3,10)	26	(3,25)	42	(1,40)
11	(2,10)	27	(2,25)	43	(3,43)
12	(1,10)	28	(2,28)	44	(1,43)
13	(1,13)	29	(1,28)	45	(2,43)
14	(3,13)	30	(3,28)	46	(3,46)
15	(2,13)	31	(1,31)	47	(2,46)
16	(2,16)	32	(3,31)	48	(1,46)

Table 6: Routing table for S_4 in M_5.

n	$F(H,n)$, $N=\infty$	$F(H,n)$, $N=48$
2	1.000000	1.000000
3	0.666667	0.666667
4	0.555556	0.555556
5	0.370370	0.444444
6	0.074074	0.104167
7	0.020576	0.020833
8	0.008230	0.000000
9	0.000914	0.000000
10	0.000305	0.000000
11	0.000051	0.000000

Table 7: $F(H,n)$ values for M_5 ($H = 5$).

B Bounds on Delay for FCFS queueing

In [1, 2] Cruz presents a method for obtaining bounds on delay in a packet switched network operating under a fixed routing strategy. This method presents several network elements that can be used as building blocks to model a wide variety of networks. We use three of these in our network model: the constant delay line to model the links, the demultiplexer (DEMUX) for the switch inputs, and the first-come first-served multiplexer (FCFS MUX) for the switch outputs. Cruz depends on traffic sources (with instantaneous rate $R(t)$) conforming to "burstiness constraints" defined as follows: Given $\sigma \geq 0$ and $\rho \geq 0$, we write $R(t) \sim (\sigma, \rho)$ if and only if for all x, y satisfying $y \geq x$ there holds

$$\int_x^y R(t)\,dt \leq \sigma + \rho(y - x). \tag{2}$$

Thus, if $R(t) \sim (\sigma, \rho)$, there is an upper bound to the amount of traffic contained in any interval that is equal to a constant σ plus a quantity proportional to the length of the interval [1]. We refer to such a source as a linear bounded arrival process in subsection 2.3. If we choose the minimum σ and ρ for our ON/OFF source transmitting over a link with capacity $C = \infty$ which satisfy (2), we get

$$\begin{aligned} \sigma &= L(1 - \rho/C) \\ &= 512\,\text{bits, and} \\ \rho &= L/T \\ &= 32\,\text{bits/ms.} \end{aligned}$$

We first consider a session along R in an M_1 network in which internal links have capacity $C_l = 1536$ bits/ms and all input links carry $N = 48$ active sessions. We assume that the delay through the switch is merely the output queueing delay experienced at the switch (i.e., the delay through the DEMUX and switching fabric is 0). Any session through this switch has the same upper bound on its delay which we denote $\overline{Q_{M_1}}$.

$$\overline{Q_{M_1}} = \frac{1}{C_l} \max_{u \geq 0} \left[\sigma + \rho u + \left(\sum_{j=1}^{3} \sigma_j + \rho_j (u + L/C_{in}) \right) - C_l u \right]$$

where $R(t) \sim (\sigma, \rho)$ characterizes the session traffic and $R(t)_j \sim (\sigma_j, \rho_j)$ characterizes the cross traffic carried by each of the three input links. Since $C_{in} = \infty$ and $\rho + \sum_{j=1}^{3} \rho_j - C_l = 0$,

$$\begin{aligned} \overline{Q_{M_1}} &= \frac{1}{C_l} \left(\sigma + \sum_{j=1}^{3} \sigma_j \right) \\ &= \frac{1}{C_l} 48\sigma \end{aligned} \tag{3}$$

because the aggregate cross traffic is composed of 47 sessions with rate $R(t) \sim (\sigma, \rho)$. Substituting values into the right hand side of (3), we get $\overline{Q_{M_1}} = 16$ ms.

We now consider an H hop session along R in an M_H network. To continue the analysis used to compute the bound for M_1, we need to characterize the output traffic of a single session from S_1 and M_1, which are same $(R(t)_{S_1}^{out} \sim (\sigma_S(H,1), \rho) = R(t)_{M_1}^{out} \sim (\sigma_M(1), \rho))$. From equation 4.28 in [1],

$$R(t)_{M_1}^{out} = \min\{ \, g_{M_1}(u), \, C_l u \, \}$$

where

$$\begin{aligned} g_{M_1}(u) &= \max_{\Delta \geq 0, \, D \geq 0} \Big[\min \{ \sigma + \rho (u + D), \\ &\qquad \sigma + \rho (u + D + \Delta) + \left(\sum_{j=1}^{3} \sigma_j + \rho_j (\Delta + L/C_{in}) \right) \\ &\qquad - C_l (\Delta + D) \} \Big] \\ &= \max_{D \geq 0} \Big[\min \{ \sigma + \rho (u + D), \\ &\qquad \sigma + \rho (u + D) + \left(\sum_{j=1}^{3} \sigma_j \right) - C_l D \} \Big]. \end{aligned} \tag{4}$$

The maximum on the right hand side of (4) occurs at $D = 47\sigma/C_l$, so

$$\begin{aligned} R(t)_{M_1}^{out} &= \min\{ \, \sigma + \rho (u + 47\sigma/C_l), \, C_l u \, \} \\ &\sim \left((1 + 47\rho/C_l)\sigma, \, \rho \, \right). \end{aligned} \tag{5}$$

Substituting values into the right hand side of (5), we get $R(t)_{M_1}^{out} \sim (\frac{95}{48}\sigma, \rho)$, meaning

$$\sigma_S(H,1) = \sigma_M(1) = \frac{95}{48}\sigma. \tag{6}$$

Finally, the upper bound on queueing delay for a session that traverses R in an M_H network we denote $\overline{Q_{M_H}}$.

$$\begin{aligned}\overline{Q_{M_H}} &= \overline{Q_{M_1}} + \\ &\frac{1}{C_l}\sum_{j=2}^{H} \sigma_S(H, j-1) + \sigma_{cross,j} + \rho_{cross}\frac{L}{C_l},\end{aligned} \tag{7}$$

where $\sigma_{cross,j} = 15\,\sigma_S(H, j-1) + 16\,\sigma_M(j-1) + 16\,\sigma_M(H-j+1)$ and $\rho_{cross} = 47\rho$. We prove that this bound grows exponentially as a function of H in section C. Note that (7) reflects the fact that $\overline{Q_{M_1}}$ holds as a delay bound through S_1 in all M_H networks. Like $\sigma_{cross,j}$, $\sigma_S(H, k)$ and $\sigma_M(H)$ contain terms for each of the three components of cross traffic at a switch:

$$\begin{aligned}\sigma_S(H,k) &= \sigma_S(H, k-1) + \frac{\rho}{C_l}(15\,\sigma_S(H, k-1) + \\ &16\,\sigma_M(k-1) + 16\,\sigma_M(H-k+1) + \sigma)\end{aligned} \tag{8}$$

and

$$\begin{aligned}\sigma_M(H) &= \sigma_S(H, H-1) + \frac{\rho}{C_l}(15\,\sigma_S(H, H-1) + \\ &16\,\sigma_M(H-1) + 16\,\sigma_M(1) + \sigma),\end{aligned} \tag{9}$$

where $\sigma_S(H, 1)$ for $H = 1, 2, \ldots$ and $\sigma_M(1)$ are given in (6).

Table 8 shows the output characterization of sessions which traverse $S_1, S_2, \ldots, S_H$ for several M_H networks, as well as upper bounds on queueing delay through the switches.

Network	Bound on queueing delay in switches (ms)					Bound on queueing delay	Output Characteriza-
(M_H)	S_1	S_2	S_3	S_4	S_5	($\overline{Q_{M_H}}$ in ms)	tion $(R(t)^{out}_{M_H})$
M_1	16.00	-	-	-	-	16.00	$\sim (1.98\sigma, \rho)$
M_2	16.00	31.99	-	-	-	47.99	$\sim (3.94\sigma, \rho)$
M_3	16.00	42.44	56.37	-	-	114.81	$\sim (8.02\sigma, \rho)$
M_4	16.00	64.20	74.07	109.42	-	263.69	$\sim (17.08\sigma, \rho)$
M_5	16.00	112.52	111.94	148.26	233.80	622.52	$\sim (38.82\sigma, \rho)$

Table 8: Upper bounds on queueing delay for FCFS in M_1 through M_5.

C Proof for Asymptotic Growth of $\overline{Q_{M_H}}$ for FCFS

For FCFS we prove that the upper bound on queueing delay, $\overline{Q_{M_H}}$ in (7), grows exponentially in the number of hops along a route in our network model. Specifically, we show that $\overline{Q_{M_H}} \in \Omega\left(\left(\frac{79}{48}\right)^H\right)$.

We can bound equations (7) through (9) from below by

$$\overline{Q_{M_H}} \geq \frac{16}{C_l}\sum_{j=1}^{H}\sigma_S(H,j-1)+\sigma_M(j-1) \tag{10}$$

where

$$\begin{aligned}\sigma_S(H,k) &= \left(\frac{C_l+15\rho}{C_l}\right)\sigma_S(H,k-1)+\\ &\quad \left(\frac{16\rho}{C_l}\right)\sigma_M(k-1) \qquad (11)\\ \text{and } \sigma_M(H) &= \left(\frac{C_l+15\rho}{C_l}\right)\sigma_S(H,H-1)+\\ &\quad \left(\frac{16\rho}{C_l}\right)\sigma_M(H-1). \qquad (12)\end{aligned}$$

We define $\sigma_S(H,0) = \sigma_M(0) = \sigma$, to match the burstiness of a single traffic source.

To complete the proof, we derive a closed form expression for the right hand side of (10). This requires two steps. In the first step we solve for $\sigma_S(H,k)$ and $\sigma_M(k)$ in terms of H and k. In the second step we substitute these expressions into (10) and finally solve a geometric series.

Step 1: We show

$$\sigma_S(H,k) = \left(\frac{C_l+31\rho}{C_l}\right)^k\sigma \quad \text{for } 0 \leq k \leq H \tag{13}$$

by induction. This is true for $k = 0$ by definition. We assume (13) is true for $k = p$ and show it holds for $k = p+1$:

$$\begin{aligned}\sigma_S(H,p+1) &= \left(\frac{C_l+15\rho}{C_l}\right)\sigma_S(H,p)+\left(\frac{16\rho}{C_l}\right)\sigma_M(p)\\ &= \left(\frac{C_l+15\rho}{C_l}\right)\sigma_S(H,p)+\left(\frac{16\rho}{C_l}\right)\sigma_S(p,p).\end{aligned}$$

By the inductive hypothesis,

$$\begin{aligned}\sigma_S(H,p+1) &= \left(\frac{C_l+15\rho}{C_l}\right)\left(\frac{C_l+31\rho}{C_l}\right)^p\sigma+\\ &\quad \left(\frac{16\rho}{C_l}\right)\left(\frac{C_l+31\rho}{C_l}\right)^p\sigma\\ &= \left(\frac{C_l+31\rho}{C_l}\right)^{p+1}\sigma.\end{aligned}$$

Since $\sigma_M(k) = \sigma_S(k,k)$,

$$\sigma_M(k) = \left(\frac{C_l+31\rho}{C_l}\right)^k\sigma.$$

Step 2:

$$\begin{aligned}
\overline{Q_{M_H}} &\geq \frac{16}{C_l}\sum_{j=1}^{H}\sigma_S(H,j-1)+\sigma_M(j-1) \\
&= \frac{32\sigma}{C_l}\sum_{j=0}^{H-1}\left(\frac{C_l+31\rho}{C_l}\right)^j \\
&= \frac{32\sigma}{31\rho}\left[\left(\frac{C_l+31\rho}{C_l}\right)^H-1\right] \\
&\in \Omega\left(\left(\frac{79}{48}\right)^H\right)
\end{aligned}$$

since $C_l = 48\rho$.

D Bounds on Delay for Stop-and-Go Queueing

In [10, 11] Golestani presents a queueing discipline (stop-and-go queueing) which supports an upper and lower bound on delay in a connection-oriented network. Stop-and-go queueing requires that traffic sources to the network be characterized in the following manner: A source is defined to be smooth with regard to the ordered pair (r, T), or (r, T)-*smooth*, if during any time frame of size T, the packets generated collectively have no more than rT bits. Since our traffic source generates at most one L bit packet every $T = 16$ ms, the smallest value of r which describes the source is

$$\begin{aligned}
r &= L/T \\
&= 32\,\text{bits/ms}.
\end{aligned}$$

Note that we may choose a smaller value for T (or larger value for r), to characterize our source, but this reduces the number of calls that can be carried by the links in our network.

For stop-and-go queueing the end-to-end queueing delay for a session is equal to a fixed value plus a term to account for delay jitter. As in section B, we assume that the delay through a switch is merely the output queueing delay experienced at the switch. Thus, the queueing delay for an H hop path is

$$\sum_{h=1}^{H}(Q_h+p_h) + d_p,$$

where Q_h is the delay, not including the packet reception time (p_h), at switch h on the path; and d_p is the delay jitter term for the entire path. Because of stop-and-go $T \leq Q_h < 2T$, depending on the route through the network, and $-T < d_p < T$. This makes $T(2H+1)+(H-1)\frac{L}{C_l}$ the upper bound on queueing delay for a session that traverses H hops. If this is the upper bound on delay for a session that traverses R in an M_H network, then the delay for any packet, which we denote Q_{M_H}, satisfies

$$2HT + (H-1)\frac{L}{C_l} - T < Q_{M_H} < 2HT + (H-1)\frac{L}{C_l} + T,$$

where $2HT$ is the maximum route-dependent queueing delay [10, 11] along R. For M_5,

$$145.3 \text{ ms} < Q_{M_H} < 177.3 \text{ ms}.$$

E Equivalent Capacity Calculations for FCFS

The equivalent capacity results in figure 17 were calculated using equations (1) and (2) in [18]. Since that formulation of equivalent capacity was directed primarily at packet loss, rather than packet delay (as in our case), we provide additional details here. We first note that we are interested in packet delay and loss on an ***end-to-end*** basis. In our study, these end-to-end performance requirements were translated into local, per-hop requirements by dividing the end-to-end value by the number of hops along R. A discussion of the issue of determining local performance constraints given end-to-end constraints can be found in [37, 38].

The theory of equivalent capacity is based on modeling session traffic and its interaction with other sessions at a multiplexer using one of two techniques: a stationary approximation, or a fluid flow approximation. The stationary approximation component of equivalent capacity models the traffic rate using a Gaussian distribution, whose mean and variance are computed from the mean and variance of the bit rate of the individual sessions being multiplexed. These calculations are as detailed in [18]. Loss is assumed to occur whenever the aggregate bit rate exceeds link capacity. Consequently, there is no sense of buffering (or delay) in the stationary approximation. This can be thought of as a multiplexer with no buffer capacity – all traffic that is not lost passes through the multiplexer with zero delay. This gives rise to the flat stationary approximation curve in figure 17. The curve indicates that under the stationary approximation, the probability that the aggregate bit rate of 87 calls (or less) exceeds the link capacity is less than 0.001. It is interesting to consider the stationary approximation in light of figure 16, in which there are a fixed number of calls per link ($N = 48$), but with a varying link utilization on the y-axis. In this case, the sum of the peak rates never exceeds the link capacity, and hence there is a zero probability of packet loss, even at 100 % utilization.

The fluid-flow component of equivalent capacity is based on modeling session traffic as a fluid flow. Equations (1) and (2) in [18] indicate how to compute the amount of capacity needed to support N sessions with a given buffer size (which is equivalent to a given delay bound in our case, given fixed length packets and FCFS service). The number of supportable sessions, plotted in figure 17 was found by finding the largest value of N such that the required capacity was less than the link capacity (C_l), for the given delay value.