Queueing Analysis of Scheduling Policies in Copy Networks of Space Based Multicast Packet Switches

Biplab Sikdar, *Student Member, IEEE* and D. Manjunath, *Member, IEEE*

***Abstract*— Space based multicast switches use copy networks to generate the copies requested by the input packets. In this paper our interest is in the multicast switch proposed by Lee [12]. The order in which the copy requests of the input ports are served is determined by the copy scheduling policy and this plays a major part in defining the performance characteristics of a multicast switch. In any slot, the sum of the number of copies requested by the active inputs of the copy network may exceed the number of output ports and some of the copy requests may need to be dropped or buffered. We first propose an exact model to calculate the overflow probabilities in an unbuffered Lee's copy network. Our exact results improve upon the Chernoff bounds on the overflow probability given by Lee by a factor of more than 10. Next, we consider buffered inputs and propose queueing models for the copy network for three scheduling policies – cyclic service of the input ports with and without fanout splitting of copy requests and acyclic service without fanout splitting. These queueing models obtain the average delay experienced by the copy requests. We also obtain the *sustainable throughput* of a copy network, the maximum load that can be applied to all the input ports without causing an unstable queue at any of the inputs, for the scheduling policies mentioned above.**

***Keywords*—Multicast Switches, Copy Networks, Queueing Analysis, Scheduling Algorithms**

I Introduction

MANY new network applications like teleconferencing, video on demand, distributed computing, etc., require that multipoint communications capability be available from the network. This means that the network should inherently support multicasting, which, in turn, means that the network should have packet switches that support multicasting in addition to unicasting. A multicast packet switch is capable of making multiple copies of an incoming packet and route them to the desired outputs.

Various space division switch architectures which support multicasting have been proposed in literature (see for example, [11, 16]). Depending on how the copies are made, two basic design paradigms have been suggested for supporting multicasting in space division switches - *space based* and *time based* [11, 17]. The general structure of a space-based space division multicast switch is that of a copy network followed by a routing stage. Various buffer placement and addressing table management policies defined on this general structure differentiate various multicast packet switch proposals [4, 5, 20]. Our interest in this paper is in modeling and analysis of space based copy networks for copy scheduling algorithms that are simple and easy to implement in hardware.

In a copy network, in any slot, the sum of the number of copies requested by the active inputs may exceed its capacity and some requests may need to be queued. Thus, in a multicast switch, the copy network introduces a delay stage, in addition to the delay in the switching stage, and we are interested in modeling this delay. The copy scheduling policy determines the order in which the input ports are served in each slot and contributes significantly to the overall performance characteristics of the multicast switch.

Performance analysis of multicast switches has been addressed in literature but very little analysis has been done for copy networks. Studies in [1, 3, 7, 8] are aimed at modeling the performance of generic multicast switches. The analyses in [12] and [13] study the copy generation process in terms of the overflow and loss probabilities but provide no insight into the queueing processes in the copy networks. In [12], Lee studies the performance his copy network by using Chernoff bounds to calculate the overflow probabilities at each input port of the copy network. In this analysis, it is assumed that the input ports are unbuffered and comparison with simulation results shows that these bounds are very loose. An approximate analysis for calculating the loss probability of packet copies in the shuffle exchange copy network has been given by Liew in [13]. Many scheduling policies with reference to multicast switches are discussed in [6, 9].

In this paper, in Section II, we first propose an exact solution to calculate the overflow probabilities in Lee's copy network for the case when no buffers are present at its inputs. The same analysis may be used for the shuffle exchange based copy network with deadlock resolution. Following this, in Section III, we consider buffered inputs and propose queueing models for the delay analysis for three scheduling policies for the service of the queues at the inputs of the copy network. The scheduling policies that we consider are cyclic service of the input ports with and without fanout splitting and acyclic service without fanout splitting. We also introduce a performance measure called the sustainable throughput for a copy network, (defined as the maximum load that can be applied to all the input ports without resulting in an unstable queue at any of the inputs) and evaluate this parameter for the above

B. Sikdar is with the Dept. of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, 12180 USA (email: bsikdar@networks.ecse.rpi.edu).

D. Manjunath is with the Dept. of Electrical Engineering, Indian Institute of Technology, Bombay, Mumbai 400076 India (email: dmanju@ee.iitb.ernet.in).

Some of this work was done while the authors were with the Dept. of Electrical Engineering , Indian Institute of Technology, Kanpur, India. Some of the results were presented at CISS'99, Baltimore, MD, USA and IEEE BSS'99, Kingston, ON, Canada.

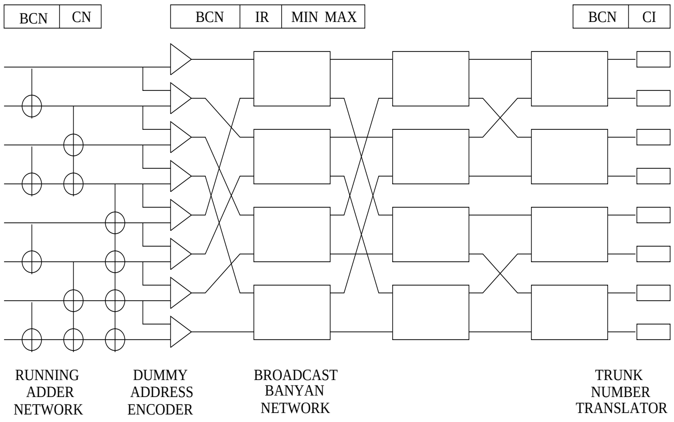


Figure 1: Lee's Copy Network for a Multicast Packet Switch [10]

mentioned scheduling policies. In Section IV we present the numerical results from our analytical models and also from simulation models. Finally, in Section V we discuss the results and provide concluding remarks on extending our results to more general situations.

II Exact Loss Analysis in Lee's Copy Network

Consider the copy network based on a broadcast banyan network proposed by Lee [12] that we will call the Lee Copy Network (LCN). The details of the LCN are shown in Fig. 1. Time is slotted and all the inputs are synchronised such that packet arrivals occur at the beginning of a slot. The copy network is a cascaded combination of a packet header encoder and a decoder and the basic structure consists of the following components - Running Adder Network (RAN), Dummy Address Encoders (DAE), Broadcast Banyan Network (BBN) and Trunk Number Translator (TNT). The header of each input packet contains the copy request, the number of copies of this packet that needs to be made by the copy network. The copy request is input to the running adder network that produces a running sum of the copy requests at each input port. These running sums are used by the dummy address encoder to form a new packet header consisting of two fields - a *dummy address interval* and an *index reference*. The running sum at the port along with that of the preceding port, represented by two binary numbers, form the address interval. This interval corresponds to the port numbers at which the copies of the input packet will be available at the end of the copying process. The index reference is used at a later stage by the trunk number translators to determine the destination port of the switch for the copies. The switching elements in the broadcast banyan network are capable of routing an input packet to either or both its outputs and the routing algorithm used is the boolean interval splitting algorithm described in [12].

To explain the working of the LCN, consider a $M \times N$ copy network, inputs numbered from 1 to M, at the beginning of a slot. Let the number of copies requested by port i be c_i. The running adder network determines the order in which the copy requests at the input ports are served in each slot. In the simplest case, an acyclic service discipline can be used in which the running adder begins at the top of the network and obtains the running sum starting from port 1 in every slot. In this service policy port i is serviced (all the copies requested by the input packet are made) if $\sum_{j=1}^{i} c_j \leq N$ in that slot. Alternatively, a cyclic service scheme that works as follows could be used. If port i is the last port served in slot n, then in slot $n+1$ ports $i+1, i+2, \cdots, i+k$ will be served. (All additions are modulo $M+1$ and $1 \leq k \leq M$.) In this policy, port m is served if $\sum_{j=i+1}^{m} c_j \leq N$. Another variation would be to introduce fanout splitting [7] in which a part of a copy request will be served whenever possible and the rest of the request will be served in subsequent slots. For example, if a packet at port i requests five copies in a slot, and $\sum_{j=1}^{i-1} c_j = N-3$, then in this slot, three copies of the packet at port i will be made and a request for two copies will be retained for service in a subsequent slot. Fanout splitting can be used with cyclic or acyclic service.

In an $M \times N$ copy network, the number of copies that can be generated in a slot is limited to N. In each slot the running adder starts summing the copy requests of the packets at the head of the input queues sequentially, beginning with port 1 (acyclic service without fanout splitting). Overflow occurs when the sum of the copy requests of the packets at the head of the input queues is greater than N. In [12] Lee obtains the Chernoff bound for this overflow probability in a copy network with no queueing at the input. The Chernoff bound is obviously a very loose bound and we now give an exact model for calculating the overflow probability.

Let the packet arrivals to input port i form a Bernoulli process with rate ρ_i. A packet in the i^{th} input port requests k copies with probability $q_i(k)$. Thus,

$$q_i(k) \equiv \text{Prob}\{c_i = k\}$$

Let X_i be the random variable for the number of copies requested by the i^{th} input port (regardless of it being active or idle). Then,

$$f_i(x_i) \equiv \text{Prob}\left\{X_i = x_i\right\} = \begin{cases} 1-\rho_i, & x_i = 0 \\ \rho_i q_i(x_i), & x_i = 1, 2, \cdots N \end{cases}$$

The probability generating function of X_i is then given by

$$\mathcal{F}_i(z) \equiv \sum_{x_i} f_i(x_i) z^{x_i} = (1-\rho_i) + \rho_i \mathcal{Q}_i(z)$$

where $\mathcal{Q}_i(z)$ is the probability generating function of $q_i(x_i)$. The copy request of port i is served if $X_1 + X_2 + \cdots + X_i \leq N$. The probability of loss at port i, $P_{loss}(i)$, is given by

$$P_{loss}(i) = 1 - \sum_{\sum_{j=1}^{i} x_j \leq N} \prod_{j=1}^{i} f_j(x_j) \qquad (1)$$

The summation on the RHS of the above equation is carried out over all possible combinations of copy requests from ports 1 to i that sum to less than or equal to N. This summation is similar to obtaining the normalisation constant in a product form queueing network with an inequality constraint on the state space. Therefore, following [14], we can obtain the $P_{loss}(i)$s as follows. Define

$$\delta(k) \equiv \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{if } k \neq 0 \end{cases} \qquad \Phi_N(k) \equiv \begin{cases} 1 & \text{if } k \leq N \\ 0 & \text{if } k > N \end{cases} \tag{2}$$

$\delta(k)$ and $\Phi_N(k)$ can be represented as contour integrals in the complex plane as follows.

$$\begin{aligned} \delta(k) &= \oint z^{(k-1)} dz \\ \Phi_N(k) &= \sum_{i=0}^{N} \delta(k-i) \;=\; \sum_{i=0}^{N} \oint z^{(k-i-1)} dz \\ &= \oint \left[\frac{z^{(N+1)}-1}{z-1}\right]\left[\frac{z^k}{z^{(N+1)}}\right] dz \end{aligned}$$

Here the contour of integration is the unit circle in the complex plane. We can use $\Phi_N(k)$ to represent the summation in Eqn. 1. This representation and the attendant simplifications are derived below.

$$\begin{aligned} 1 - P_{loss}(i) &= \sum_{x_1=0}^{N} \cdots \sum_{x_i=0}^{N} \prod_{k=1}^{i} f_k(x_k) \Phi_N(x_1 + \cdots + x_i) \\ &= \sum_{x_1=0}^{N} \cdots \sum_{x_i=0}^{N} \prod_{k=1}^{i} f_k(x_k) \oint z^{(x_1+x_2+\cdots+x_i)} \\ &\qquad \left[\frac{z^{(N+1)}-1}{z-1}\right]\left[\frac{1}{z^{(N+1)}}\right] dz \\ &= \oint \sum_{x_1=0}^{N} f_1(x_1) z^{x_1} \cdots \sum_{x_i=0}^{N} f_i(x_i) z^{x_i} \\ &\qquad \left[\frac{z^{(N+1)}-1}{z-1}\right]\left[\frac{1}{z^{(N+1)}}\right] dz \\ &= \oint \left[\frac{z^{(N+1)}-1}{z-1}\right]\left[\frac{1}{z^{(N+1)}}\right] \prod_{k=1}^{i} \mathcal{F}_i(z) dz \end{aligned} \tag{3}$$

The Chernoff bounds on the overflow probabilities can be represented as,

$$\begin{aligned} P_{Chernoff}(i) &= \text{Prob}\{X_1 + \cdots + X_i > N\} \\ &\leq e^{-sN} \prod_{k=1}^{i} \mathcal{F}_i(e^s) \end{aligned} \tag{4}$$

RHS of Eqn. 3 is evaluated using the residue theorem. Note that our contour of integration is the unit circle and the only poles of the integrand in Eqn. 3 inside the unit circle is at $z = 0$. Hence, the integral may be evaluated by calculating the residue of the integrand at $z = 0$.

To compare the improvement of the exact result over the bound we consider an unbuffered 64×64 copy network. Each input port is assumed to have an iid Bernoulli arrival process with an effective load of 0.6 in all cases. Eqns. 3 and 4 give the exact value and the Chernoff bounds for

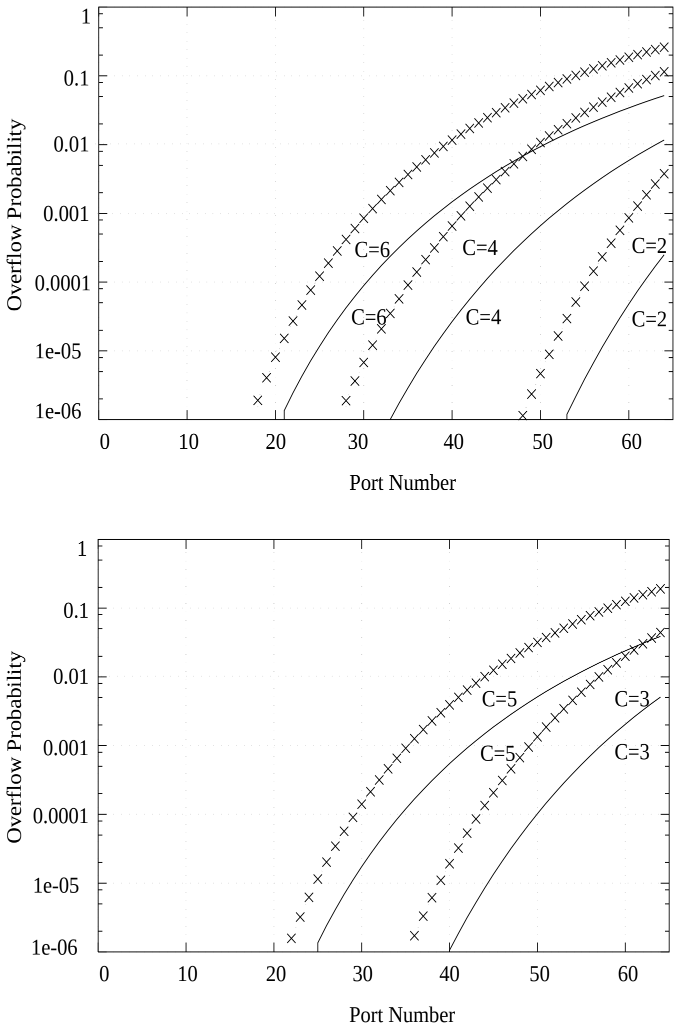
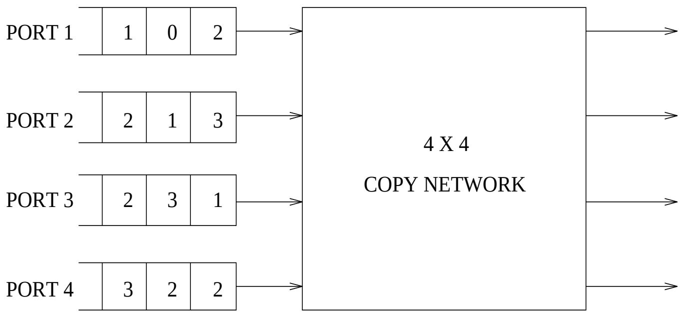


Figure 2: Overflow probabilities in a 64×64 copy network with deterministic copy requests. The broken lines denote the Chernoff bounds while the smooth lines represent the exact results. The overflow probabilities have been evaluated for $\bar{Q} = 2, 3, 4, 5$ and 6.

the overflow probabilities at the inputs of the copy network, respectively. For deterministic copy requests of size $\bar{Q}$, Fig. 2 shows the Chernoff bound and the exact result for $\bar{Q} = 2$, 3, 4, 5 and 6. As can be seen, the values of the overflow probabilities obtained using Chernoff bounds are of the order of 10 times higher than the actual values.

III Queueing Models for Lee's Copy Network

When the number of copies requested in a slot exceeds N, some requests need to be buffered. A scheduling policy will decide which of the input queues are to be served. Within each queue the packets are served on a FCFS basis and only the HOL packet is considered for service in a slot. Thus, head of the line blocking limits the maximum achievable throughput from each queue. Fig. 3 gives an example of this phenomenon. Yet another kind of blocking that we call "predecessor port blocking" is experienced by an input packet in a copy network. This type of blocking



In a slot, assume that service begins with port 1. The head of the line packets at the other ports is not served because the running adder sum exceeds 4 at port 2. The second packet at port 2 could have been served but was blocked due to the packet at the head of the line. Also, the packet at port 3 could have been served along with the packet at port 1 if the running adder had skipped port 2 but is blocked because of "predecessor port blocking".

Figure 3: Blocking in a Copy Network

is best explained by an example. Consider ports $i-1$, i and $i+1$ of the copy network requesting c_{i-1}, c_i and c_{i+1} copies respectively. Ports i and $i+1$ will not be served if $\sum_{j=1}^{i} c_j > N$ even if $\sum_{j=1}^{i-1} c_j + c_{i+1} \leq N$, i.e., port $i+1$ is blocked because the running sum exceeds N at i although $i+1$ could have been served by not serving i. In unicast switches, head of the line blocking limits the maximum input load that can be supported to 0.586 [10]. To obtain a similar characterisation, we can define the *sustainable throughput* of a copy network as the maximum of the load, which when applied to all the input ports, does not lead to instability in any of the M input queues. In this section, we describe methods to find the sustainable throughput for the scheduling policies that we consider.

To maintain analytical tractability, we assume that the packet arrivals are independent and identical Bernoulli processes at each input. Each input has infinite buffers. We also assume that the copy request of the packets at the head of the line at the inputs in a given slot are independent of those from the previous slot. This is a strong assumption but necessary to keep the analysis simple and manageable. Our analysis method is as follows. For the scehuling without fanout splitting (Sections A and B, we model each input queue as discrete time M/M/1 queue [19] in which the arrival process is Bernoulli and the service times are geometrically distributed. Under each of the above policies, we calculate the service rate at each input port, i.e., the probability that an active input (copy request) is served in a slot. We then use the results of discrete time M/M/1 queue analysis to obtain the waiting time and queue length distributions. A similar analysis is used for the case of finite buffers at the inputs. In modeling cyclic service with fanout splitting, the entire switch is modeled as a single GI/D/1 queue.

A Acyclic Service without Fanout Splitting

This is probably the most obvious scheduling policy in a copy network. In each slot, this policy serves the input ports serially, starting from port 1. The running adder sums up the copy requests of the head of the line packet at each port. All the ports for which the running adder sum is less than or equal to N are served in the slot. Since service always starts from port 1, the policy is unfair to the ports with higher addresses. Consequently, the service rate varies with the port number and decreases as the port address increases. In addition, since the policy does not employ fanout splitting, in cases where a part of a copy request could be served, resources remain unutilised. Thus the sustainable throughput in this case will depend on the service rate of the last input port.

The effective service rate at an input port, the rate at which the copy requests can be actually served, depends on the arrival processes and effective service rates at the preceding ports. We model each input port as a discrete time M/M/1 queue except for the first port which always gets to be served in every slot. At port i, let the effective service rate, the mean waiting time and the probability of its input queue being empty be denoted by $\mu(i)$, $W(i)$ and $P_{0,i}$ respectively. Let ρ be the rate of the Bernoulli arrival process at each of the ports. A packet in an active input requests k copies with probability $q(k)$. As in the previous section, let X_i denote the number of copies requested by port i. The number of ports that can be served in a slot depends on the copy requests of the packets at the head of the queues. Let $f_{H,i}(k)$ be the probability mass function (pmf) of the number of copies requested by the packet at the head of input queue i. From our definition above, $1-P_{0,i}$ is the probability that the head of queue i is non empty. Hence the probability mass function and the probability generating function of the copies requested by a packet at the head of the queue i will be,

$$\begin{aligned} f_{H,i}(x_i) &\equiv \text{Prob}\{X_i = x_i\} \\ &= \begin{cases} P_{0,i}, & x_i = 0 \\ (1-P_{0,i})\, q(x_i), & x_i = 1, 2, \cdots \end{cases} \end{aligned} \tag{5}$$

$$\mathcal{F}_{H,i}(z) \equiv \sum_{x_i} f_{H,i}(x_i) z^{x_i} = P_{0,i} + (1-P_{0,i})\, \mathcal{Q}(z)$$

where $\mathcal{Q}(z)$ is the probability generating function of $q(k)$.

Since port 1 is always served in each slot irrespective of the copy requests of the other ports,

$$\mu(1) = 1.0 \qquad W(1) = 1.0 \qquad P_{0,1} = 1 - \rho$$

To obtain $\mu(i)$, $W(i)$ and $P_{0,i}$ for $i = 2, \cdots M$, consider a slot in which it is given that port i has a packet and is requesting k copies. This copy request will be served in the slot if the sum of the copies requested by ports 1 to $i-1$ is less than or equal to $N-k$. By unconditioning this probability on k, $\mu(i)$, is evaluated as follows

$$\begin{aligned} \mu(i+1) &\equiv \text{service rate at port } i+1 \\ &= \text{Prob}\{\text{packet at port } i+1 \text{ is served in a slot}\} \\ &= \sum_{k=1}^{N} q(k)\ \text{Prob}\left\{ \sum_{j=1}^{i} X_j \leq N-k \right\} \end{aligned}$$

$$= \sum_{k=1}^{N} q(k) \left[\sum_{\sum_{j=1}^{i} x_j \le (N-k)} \prod_{j=1}^{i} f_{H,j}(x_j) \right] \quad (6)$$

Using $\Phi(N-k)$ to simplify the above equation and proceeding as for Eqn. 3, we have,

$$\begin{aligned}\mu(i+1) &= \sum_{k=1}^{N} q(k) \sum_{x_1=0}^{N} \cdots \sum_{x_i=0}^{N} \prod_{j=1}^{i} f_{H,j}(x_j) \times \\ &\quad \times \Phi_{N-k}(x_1 + \cdots + x_i) \\ &= \sum_{k=1}^{N} q(k) \oint \left[\frac{z^{(N-k+1)} - 1}{z-1} \right] \left[\frac{1}{z^{(N-k+1)}} \right] \prod_{j=1}^{i} \mathcal{F}_{H,j}(z) dz \end{aligned} \quad (7)$$

Using the results for discrete time M/M/1 queues from [19], $W(i)$ and $P_{0,i}$ are

$$\begin{aligned} W(i) &= \begin{cases} 1.0 & \text{for } i = 1 \\ \frac{1-\rho}{\mu(i)-\rho} & \text{for } i = 2, \cdots M \end{cases} \\ P_{0,i} &= \begin{cases} 1-\rho & \text{for } i = 1 \\ \frac{(\mu(i)-\rho)}{\mu(i)} & \text{for } i = 2, \cdots M \end{cases} \end{aligned} \quad (8)$$

It is easy to see that $W(i)$ above can also be obtained from Little's theorem. Eqns. 7 and 8 can be used to recursively obtain the performance statistics at each port.

As has been mentioned earlier, the effective service rate decreases as the port number increases. Thus, if we allow the same arrival rate to all the input ports, the effective service rate of port M upper bounds the per port input traffic rate that can be supported by the copy network. Consequently, the sustainable throughput of the copy network under the acyclic scheduling policy without fanout splitting, $\lambda_{A:NFS}$, is determined by the effective service rate at port M. Thus

$$\lambda_{A:NFS} = Q'(1) \max_{\rho} \{\mu(M)\} \quad (9)$$

where $Q'(1)$ is the average number of copies requested by a packet and

$$\mu(M) = \sum_{k=1}^{N} q(k) \left[\oint \left[\frac{z^{(N-k+1)} - 1}{z-1} \right] \left[\frac{1}{z^{(N-k+1)}} \right] \prod_{j=1}^{M-1} \mathcal{F}_{H,j}(z) dz \right]$$

Using results from [2] we can show that the variance of the waiting time for the packets at input i, $\mathrm{var}(W(i))$ is

$$\mathrm{var}(W(i)) = \begin{cases} 0.0 & \text{for } i = 1 \\ \frac{1-\rho-\mu(i)+\rho\mu(i)}{(\mu(i)-\rho)^2} & \text{for } i = 2, \cdots M \end{cases} \quad (10)$$

A.1 The Finite Buffer Case

When there are finite buffers at each input, say K, the analysis is similar to the previous case and we model each input port as a discrete time M/M/1/K queue. The queue at port 1 always gets a chance to be served in each slot and the queue contents never exceed the buffer capacity. For ports $i = 2, \cdots, M$, $P_{0,i}$ in the expression for $\mathcal{F}_{H,i}(z)$ will change. Other than that, $\mu(i)$ is calculated as in the infinite buffer case. Using the results from [19], $W(i)$ and $P_{0,i}$ are given by

$$\begin{aligned} W(i) &= \frac{1 - \gamma(i)^K [1 + K(1-\gamma(i))]}{(1-\gamma(i))(1-\gamma(i)^K)\mu(i)} \\ P_{0,i} &= \frac{(\mu(i)-\rho)}{\mu(i) - \left[\frac{\rho(1-\mu(i))}{\mu(i)(1-\rho)}\right]^K \rho} \end{aligned} \quad (11)$$

where

$$\gamma(i) \equiv \frac{\rho(1-\mu(i))}{\mu(i)(1-\rho)} \quad (12)$$

The buffer overflow probability at ports $2 \cdots M$ is [19],

$$P_{overflow}(i) = \frac{(1-\gamma(i))\gamma(i)^K}{1 - \gamma(i)^{1+K} - \mu(i)(1-\gamma(i))} \quad (13)$$

B *Cyclic Service without Fanout Splitting*

In this scheduling policy, the inputs are served cyclically. In each slot, the running adder sums the copy requests of packets at the head of the queues beginning at the first input port that was not served in the previous slot and continues sequentially till it sums the requests from M ports or the sum exceeds N. The ports where the sum exceeds N, are not served in the slot. Fanout splitting is not allowed. This policy is fair to all the inputs. It is easy to see that if we assume the traffic to be independent and identically distributed at all the ports, the performance metrics will be identical at all the inputs.

As in the previous subsection, we assume independent, identical Bernoulli arrivals at each of the input ports. In a slot, consider the probability $\mu(k)$, that a tagged input port having a packet requesting k copies gets served. This probability depends on how many ports are to be served before the tagged port in the slot and the copy request distribution of those ports. In the slot, the number of ports to be served before the tagged port is uniform in $[0, M-1]$. Thus the probability that the tagged port is served after i ports is $1/M$ for $0 \le i \le M-1$. The probability of service for the packet under consideration can then be obtained by summing the probability that the sum of the copies requested by the head of the line packets of the preceding i ports is at most $N-k$, for all values of i. For the case when $i = 0$, i.e. the tagged port is the first one to be served, the probability of service of the packet is 1. Thus,

$$\begin{aligned}\mu(k) &\equiv \text{Prob}\{\text{tagged port is served in the slot} \mid \text{HOL} \\ &\qquad \text{packet requests } k \text{ copies}\} \\ &= \frac{1}{M} \left[1 + \sum_{i=1}^{M-1} \left[\sum_{\sum_{j=1}^{i} x_j \le (N-k)} \prod_{j=1}^{i} f_{H,j}(x_j) \right] \right] \end{aligned} \quad (14)$$

The probability of service for an arbitrary copy request can be obtained by unconditioning Eqn. 14 on k. The probability of a copy request at the tagged port being served, μ, in a slot is then given by,

$$\mu = \frac{1}{M}\sum_{k=1}^{N} q(k)\left[1+\sum_{i=1}^{M-1}\sum_{\sum_{j=1}^{i} x_j \le (N-k)}\prod_{j=1}^{i} f_{H,j}(x_j)\right] \quad (15)$$

Using the simplifications of Eqn. 3 and 7, we have,

$$\begin{aligned}\mu &= \frac{1}{M}\sum_{k=1}^{N} q(k)\left[1+\sum_{i=1}^{M-1}\sum_{x_1=0}^{N}\cdots\sum_{x_i=0}^{N}\right.\\ &\qquad\left.\prod_{j=1}^{i} f_{H,j}(x_j)\Phi_{N-k}(x_1+\cdots+x_i)\right]\\ &= \frac{1}{M}\sum_{k=1}^{N} q(k)\left[1+\sum_{i=1}^{M-1}\oint\left[\frac{z^{(N-k+1)}-1}{z-1}\right]\right.\\ &\qquad\left.\left[\frac{1}{z^{(N-k+1)}}\right]\prod_{j=1}^{i}\mathcal{F}_{H,j}(z)dz\right] \quad (16)\end{aligned}$$

This μ is used as the service rate in the discrete time M/M/1 queue model for each input. As before, known results from [19] are used to obtain W, P_0 and var(W), the mean waiting time, the probability of the queue being empty and the variance of the waiting time, at each port.

$$W=\frac{1-\rho}{\mu-\rho},\ P_0=\frac{(\mu-\rho)}{\mu},\ \mathrm{var}(W)=\frac{1-\rho-\mu+\rho\mu}{(\mu-\rho)^2} \quad (17)$$

Once again, as in the previous section, note that W above can be obtained from Little's theorem. From Eqns. 16 and 17, μ is evaluated iteratively. We have not investigated the proof of convergence but for all the examples considered, encompassing diverse types of input traffic, convergence in the fifth decimal place occurs in less than fourteen iterations.

The effective service rate is the same at all the ports and the sustainable throughput for this scheduling policy, $\lambda_{C:NFS}$, is the maximum load that can be supported by any input port. Thus,

$$\lambda_{C:NFS} = Q'(1)\max_{\rho}\{\mu)\} \quad (18)$$

where $Q'(1)$ is the average number of copies requested by a packet and μ is obtained from an iterative solution of Eqn. 16.

To analyse the finite buffer case, as with the previous policy, we model each finite buffered input port as a discrete time M/M/1/K queue and μ is obtained from Eqn. 16, but with a different P_0 in the $\mathcal{F}_{H,j}(z)$ because the queue has finite buffers. As in the analysis of the acyclic policy, using $\gamma=\rho(1-\mu)/\mu(1-\rho)$, P_0, W and $P_{overflow}$ are given by [19],

$$\begin{aligned}W &= \frac{1-\gamma^K\left[1+K(1-\gamma)\right]}{(1-\gamma)(1-\gamma^K)\mu}\\ P_0 &= \frac{(1-\mu(1-\gamma)}{1-\gamma^{1+K}-\mu(1-\gamma)}\\ P_{overflow} &= \frac{(1-\gamma)\gamma^K}{1-\gamma^{1+K}-\mu(1-\gamma)} \quad (19)\end{aligned}$$

C Cyclic Service with Fanout Splitting

This scheduling policy is like the previous one except that we allow fanout (or group splitting). Fanout splitting of packets whose copy requests can be partially served in the slot increases the throughput of the copy network and as before the performance statistics will be identical at all the ports.

Under this policy, N copies are generated in a slot whenever the sum of the copy requests at the heads of the queues is greater than or equal to N. If this sum is less than N, all of these requests are served. Thus the copy network can be approximated as a single, discrete time N-server queue with deterministic service times. Since the arrival process to the copy network is the sum of the iid processes of each of the inputs, the copy network can be modeled as a GI/D/N queue. Cyclic service results in the service rate at each input and hence the delay statistics for all the copies being identical. This insensitivity of the service rate on the port number along with the fact that if there are N or more requests at the HOL of the inputs, N will be served, and if there are less than N, all will be served, allows the entire copy network to be treated as a single queue. We now obtain the arrival process to the queue and use the results for the GI/D/N queue to obtain the delay statistics.

Let $A(k)$ denote the number of copies requested by the packet arrivals during the k^{th} slot. The distribution of the total number of copy request arrivals to the copy network in a slot is the M-fold convolution of the copy request distributions at the M input ports. The probability generating function of $A(k)$, $\mathcal{A}(z)$, is given by,

$$\mathcal{A}(z)=\prod_{i=1}^{M}\mathcal{F}_i(z)=((1-\rho)+\rho\mathcal{Q}(z))^M \quad (20)$$

In a slot, the system state refers to the total number of copy requests present in the queue. Let the number of copy requests in the system at the beginning of the k^{th} slot be denoted by S_k and its moment generating function be given by $\mathcal{S}_k(z)$. Let the steady state versions of S_k and $\mathcal{S}_k(z)$ be denoted by S and $\mathcal{S}(z)$ respectively. From [2], $\mathcal{S}(z)$ is given by

$$\mathcal{S}(z)=(N-A'(1))\frac{(z-1)\mathcal{A}(z)}{z^N-\mathcal{A}(z)}\prod_{j=1}^{N-1}\frac{(z-z_j)}{(1-z_j)} \quad (21)$$

where the quantities z_j, $1\le j\le N-1$, (and also $z=1$) are the complex zeroes of $z^N-\mathcal{A}(z)$ inside the unit disk of the complex z-plane. The mean number of copy requests in the system, $\bar{S}$, and the mean waiting time, W, are given by [2]

$$\bar{S} = A'(1)+\frac{1}{2}\prod_{j=1}^{N-1}\frac{(1+z_j)}{(1-z_j)}+\frac{A''(1)-(N-1)A'(1)}{2\,(N-A'(1))}$$

$$W = \frac{1}{A'(1)} \left[\frac{1}{2} \sum_{j=1}^{N-1} \frac{(1+z_j)}{(1-z_j)} + A'(1) + \frac{A''(1) - (N-1)\,A'(1)}{2\,(N - A'(1))} \right] \quad (22)$$

The sustainable throughput for this policy is obviously 1.0.

It is easy to see that the copy network is not exactly a GI/D/N queue. When the head of the line copy requests sum to less than N, the copy network serves the requests of only the head of the line packets even though there may be packets in the inputs which could have been served. Thus work conservation is not preserved in the copy network whereas under the GI/D/N queue model these packets would have been served and work conservation would have been preserved. This approximation leads to a slight underestimation of the waiting time for the copies. Also, unlike the GI/D/N queue, the service discipline of the copy network is not first come first served. Thus, only the first moment results of the delay analysis of our model are valid.

D Acyclic Service with Fanout Splitting

This policy will have a higher throughput because of fanout splitting. The acyclic service discipline makes the scheme unfair although the sustainable throughput will be higher than in the case of acyclic service without fanout splitting. In each slot, the copy requests are serviced starting with port 1. Consequently, the probability of service at port i, $i > 1$, depends on the copy request distributions of the head of the line packets of the $i-1$ ports above it. Fanout splitting of contending packets changes the original distribution for the copy requests at the head of the input queues, $F_{H,i}(x_i)$, given in Eqn. 5. This distribution must now be modified to account for the residue of an original copy request that has been partially served. We have not been able to develop a suitable technique to model this distribution. If this distribution is known, the techniques developed in Section A may be applied to develop a queueing model for this service discipline.

IV Numerical Results

In this section we present the numerical results of the delay analysis and sustainable throughputs for the three scheduling policies modeled in the previous section. We compare these results with those from simulation models to test the goodness of our approximations. The simulation results are obtained by running the simulator till the statistics converge to within a few decimal places.

The copy requests in each slot are assumed to be either deterministic (C) or random (with mean $\bar{Q}$). For random copy requests we assume a truncated geometric distribution. To compare results for different copy request distributions we use $\lambda_{eff}(i)$, the effective load at port i, defined as

$$\lambda_{eff}(i) \equiv \rho_i \bar{Q}_i \quad (23)$$

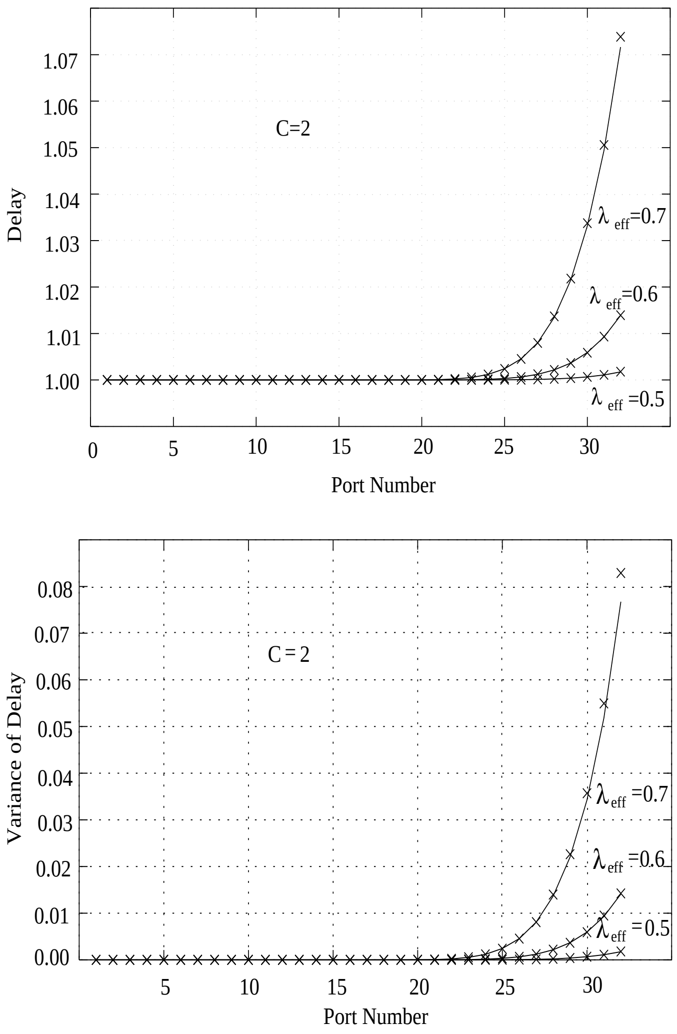


Figure 4: Mean delay and variance in a 32×32 copy network under acyclic service without fanout splitting scheduling policy. Deterministic copy requests.

A Acyclic Service without Fanout Splitting

With the service beginning with port 1 in each slot, the scheduling policy is unfair to the higher numbered ports. A lower service rate and consequently, a higher delay is thus expected at ports with higher addresses. Figs. 4 and 5 show the mean delay and the variance against the port number for various values of the effective load for a 32×32 copy network. Recall that the delay at port i, $W(i)$, is given by Eqn. 8. Ports 1-20 experience approximately equal mean delay at almost all loads. Further, the delay variance is also very low for these ports low. It is interesting to note that the unfairness of this algorithm is significant for ports greater than about 25 in terms of both the mean and the variance of the delay and increases very fast with the port number. Also observe that as expected, the variance is higher for random copy requests than that for deterministic copy requests. We also note that the results for deterministic copy requests are more accurate than those for the random requests. This is because the copy requests of the head of the line packets in successive slots are indeed

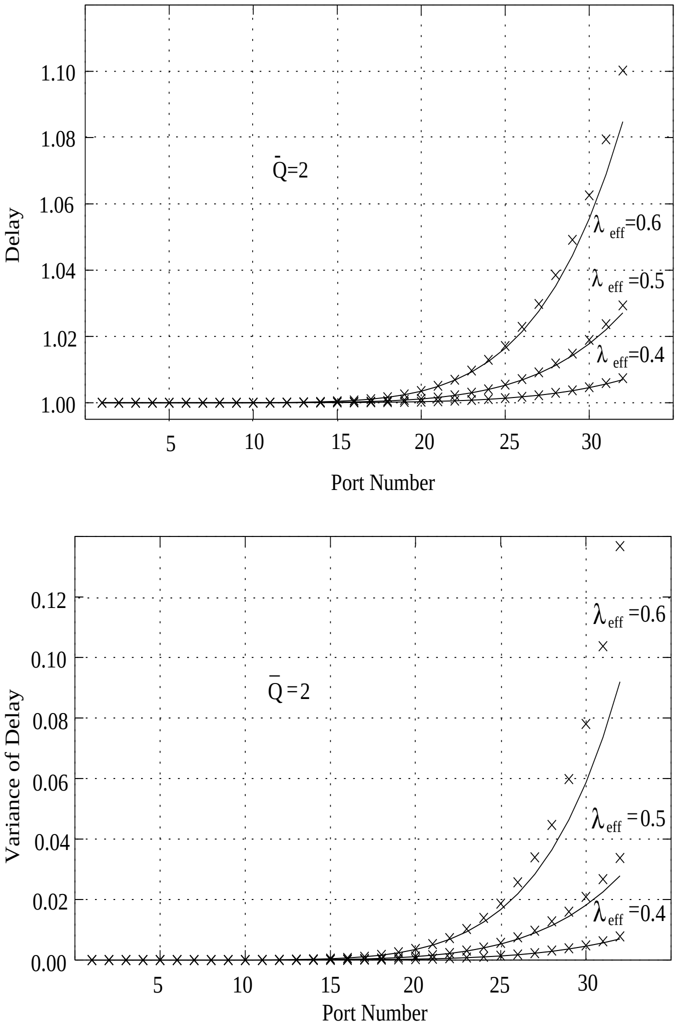


Figure 5: Mean delay and variance in a 32×32 copy network under acyclic service without fanout splitting scheduling policy. Random copy requests with truncated geometric distribution.

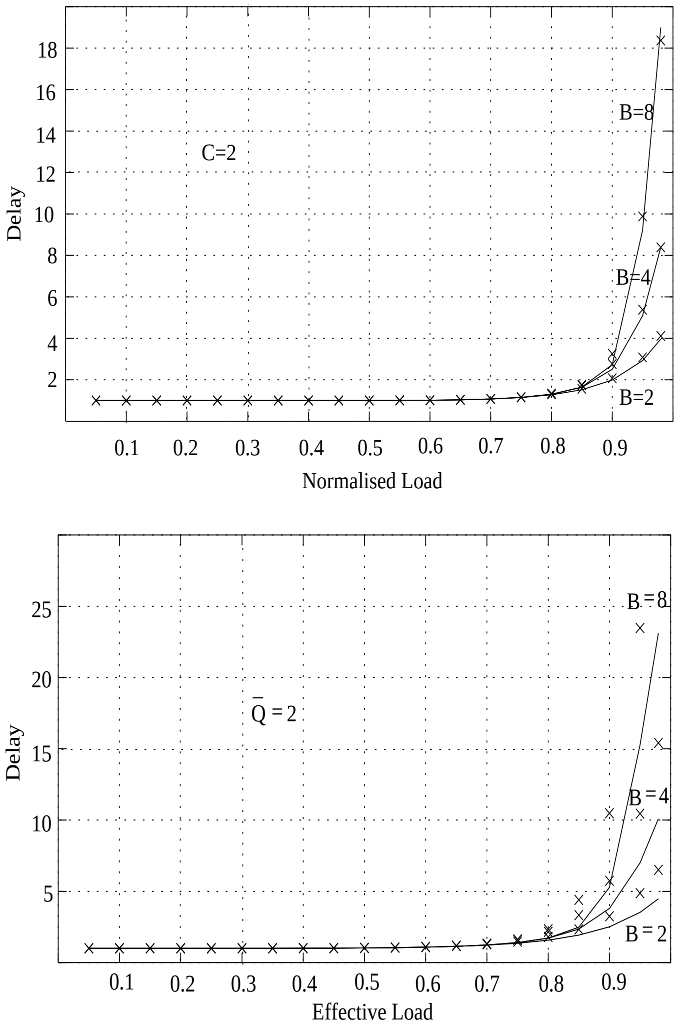


Figure 6: Mean delay in a 32×32 copy network with finite buffers under acyclic service without fanout splitting scheduling policy. Deterministic and random copy requests.

"independent" in the case of deterministic copy requests. In cases where the copy request is random, packets with larger fanout would be more likely to be blocked and stay at the head of the queue. The independence assumption fails to take this into account thereby resulting in inaccuracies in the results. Nevertheless, the isolated, discrete time M/M/1 queue model of each input port can be seen to be a very good approximation as the worst case difference between the simulation and analytical results in the delay is about 12%.

In Fig 6, we show the waiting times for port 32 in a 32×32 copy network with finite buffers. We show the mean delay for both deterministic and random copy requests. Table 1 shows the overflow probability for deterministic and random copy requests. Observe that for random copy requests the blocking probability can be nearly 10 times higher than when the copy requests are deterministic. Note that although the delay increases slowly for large effective loads, the overflow probability is considerably high even with 8 buffers for effective loads of about 0.8. This probability is significantly larger for random copy requests than for deterministic requests.

Eff Load	Deterministic, C=2		Random, $\bar{Q} = 2$	
	Buffer Size		Buffer Size	
	4	8	4	8
0.80	0.000221	0.000000	0.011688	0.001048
0.85	0.002556	0.000020	0.050244	0.012765
0.90	0.024025	0.002488	0.169924	0.112832

Table 1: Overflow Probabilities at input port 32 for acyclic service without fanout splitting scheduling policy

B *Cyclic Service without Fanout Splitting*

The cyclic scheduling policies are fair and provide the same service rate at all the input ports. Thus it is sufficient to characterize the performance metrics of a port.

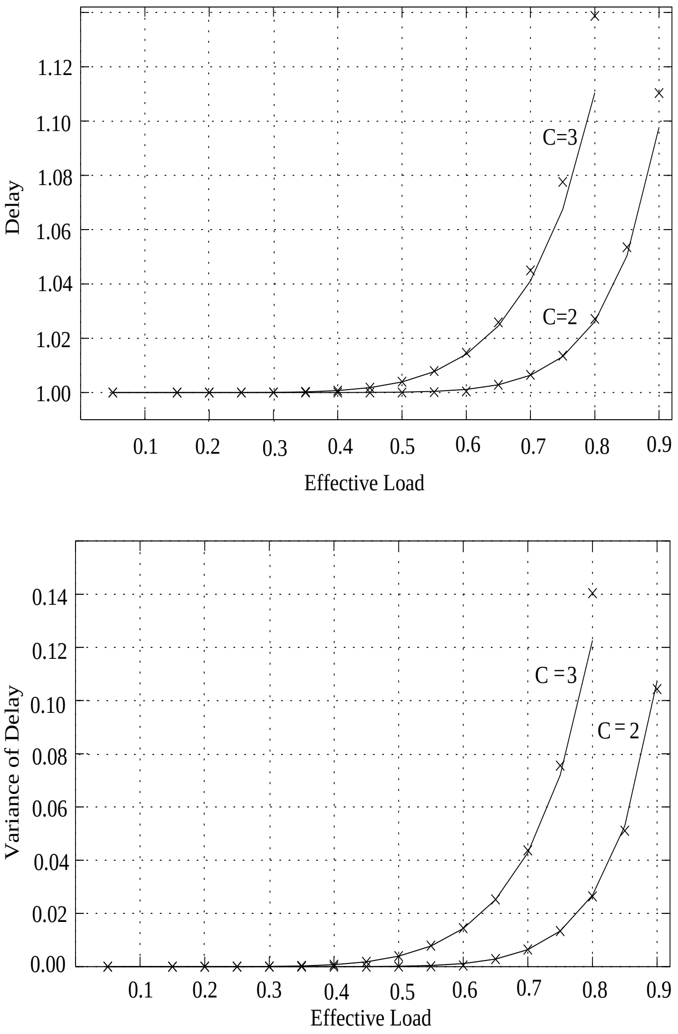


Figure 7: Mean delay and variance in a 32×32 copy network under cyclic service without fanout splitting scheduling policy. Deterministic copy requests.

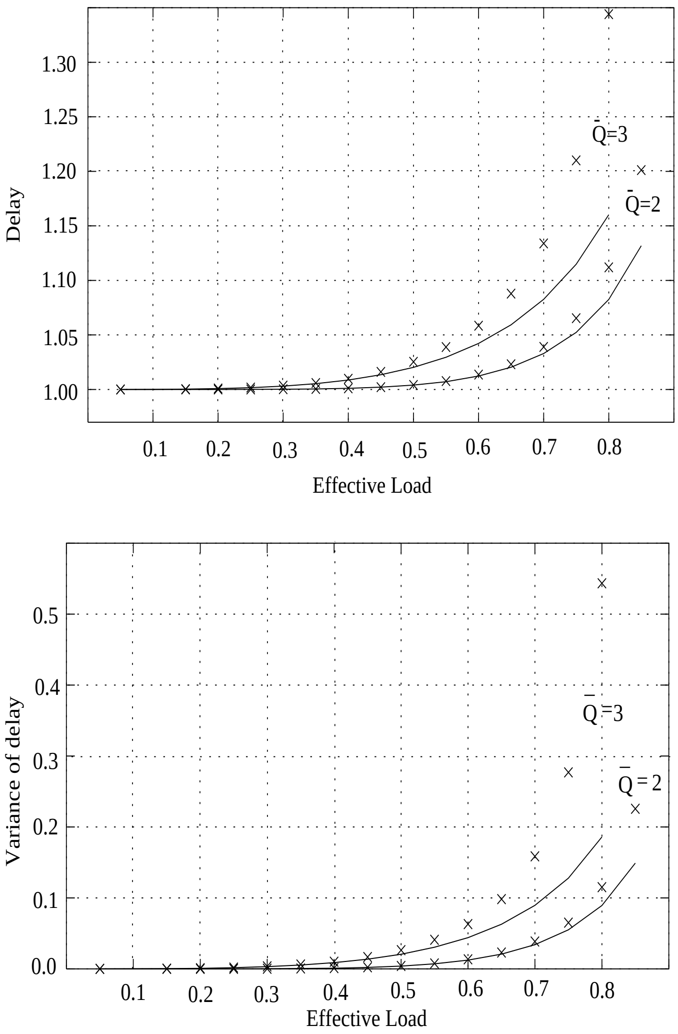


Figure 8: Mean delay and variance in a 32×32 copy network under cyclic service without fanout splitting scheduling policy. Random copy requests with truncated geometric distributions.

Eqn. 17 gives the expression for the mean waiting time W. Figs. 7 and 8 show the graph of mean delay and variance versus the effective load for a 32×32 copy network under this scheduling policy for deterministic and random copy requests. Qualitatively, the results are similar to those of acyclic service without fanout splitting, in that deterministic copy requests result in lower delays. However, the delays are much lower than in the case of acyclic service, especially at the higher numbered input ports. As before, the analysis for deterministic copy requests does better than that for random copy requests but we see that the worst case error in the delay is about 20% for the case of random copy requests. Fig. 9 shows the mean waiting times for a finite buffer copy network for both deterministic and random copy requests.

In Table 2 we show the overflow probabilities for various buffer sizes. Observe the considerable improvement in the overflow probability with this policy as compared to the acyclic policy. For example, with random requests, 90% load, and 8 buffer spaces at the inputs, the acyclic policy offers a blocking probability of 0.11 at port 32 (the worst affected port) whereas with the cyclic policy without fanout splitting, the probability is 4×10^{-6}.

C Cyclic Service with Fanout Splitting

The delay characteristics for a 32×32 copy network under cyclic service with fanout splitting scheduling policy as given by the approximate discrete time GI/D/N model is

Eff Load	Deterministic, C=2		Random, $\bar{Q} = 2$	
	Buffer Size		Buffer Size	
	4	8	4	8
0.80	0.000000	0.000000	0.000002	0.000000
0.85	0.000000	0.000000	0.000030	0.000000
0.90	0.000001	0.000000	0.000399	0.000004

Table 2: Overflow Probabilities for cyclic service without fanout splitting scheduling policy

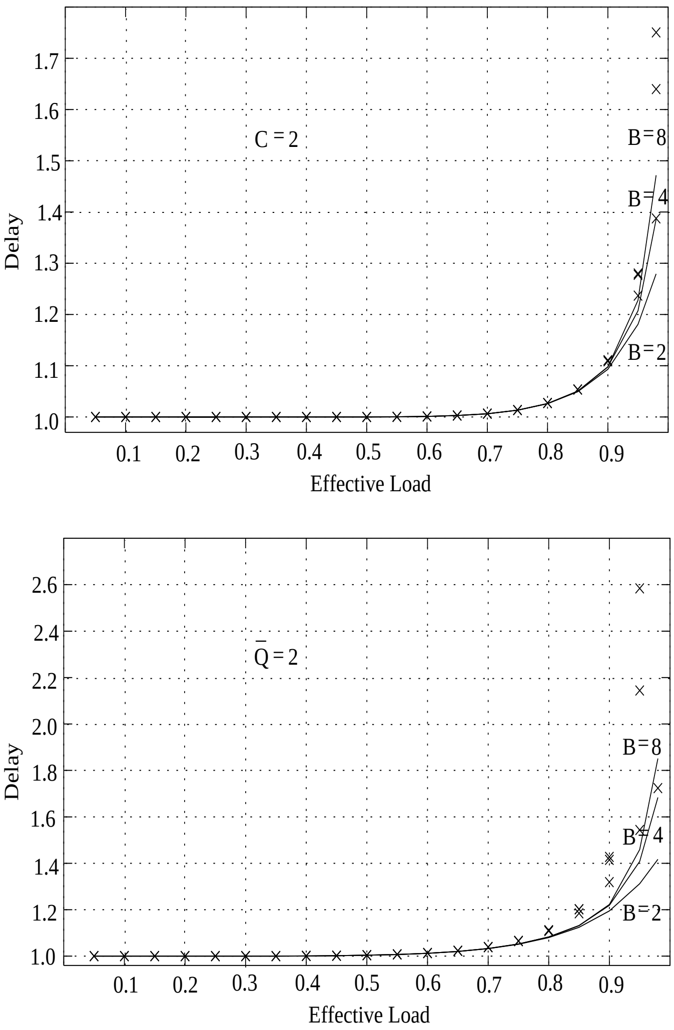


Figure 9: Mean delay and variance in a 32×32 copy network with finite buffers under cyclic service without fanout splitting scheduling policy. Deterministic and random copy requests.

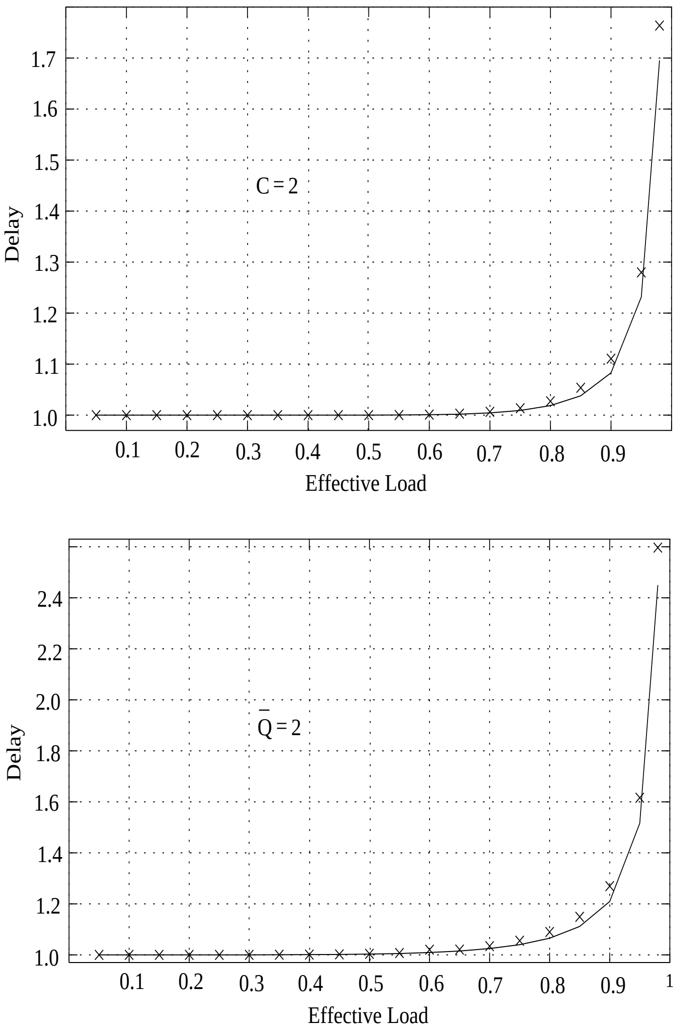


Figure 10: Mean delay in a 32×32 copy network under cyclic service with fanout splitting scheduling policy. Deterministic and random copy requests.

shown in Fig. 10. Recall that the expression for the delay, derived in Section C is given by Eqn. 22. As is evident from the figures, the model is a very good approximation at all loads and the worst case error in the delay is less than 3%. The sustainable throughput of the copy network under this scheduling policy is 1. Hence, this scheduling policy has the best performance characteristics amongst the scheduling policies we have discussed. Table 3 shows the simulation results for the overflow probabilities at the inputs of the copy network for various sizes. It is easy to see that this policy should do better than the other two policies. However, observe that there is substantial improvement in the overflow probability, as compared to the cyclic policy without fanout splitting only when the load is 90%.

V Discussions and Conclusion

In Section II we presented the overflow probabilities at the inputs of the copy network when there is no queueing at the input. Our model gives exact results which are a considerable improvement over the bounds given by Lee [12]. The Chernoff bounds are of the order of 10 times higher than the actual values. For a more realistic analysis, buffers at the inputs and the delay that they introduce need to be modeled. This delay depends on the copy scheduling policy. For the various copy scheduling policies that we believe are easy to implement in hardware, our interest was primarily in obtaining the delay in copy generation and the

Eff Load	Deterministic, C=2		Random, $\bar{Q}=2$	
	Buffer Size		Buffer Size	
	4	8	4	8
0.80	0.000000	0.000000	0.000000	0.000000
0.85	0.000000	0.000000	0.000004	0.000000
0.90	0.000001	0.000000	0.000062	0.000000

Table 3: Overflow Probabilities for cyclic service with fanout splitting scheduling policy

Num of Ports	Deterministic		Random	
	Acyclic	Cyclic	Acyclic	Cyclic
4	0.69	0.75	0.61	0.72
8	0.74	0.86	0.67	0.80
16	0.78	0.90	0.73	0.86
32	0.82	0.93	0.77	0.90

Table 4: Sustainable throughput for cyclic and acyclic service without fanout splitting scheduling policy for $\bar{Q} = 2$

sustainable throughputs for each of the scheduling policies.

The fanout splitting scheduling policies have a considerably lower delay than the non fanout splitting policy for the same input load. This is because the fanout splitting policies have a better system usage. It can also be seen that there is an improvement in the sustainable throughput for cyclic service with no fanout splitting compared to acyclic with no fanout splitting policy. The increase is as much as 17% for a 32×32 copy network with random copy requests with a mean request of 2 copies. Random copy requests experience larger variations in the delay than deterministic copy requests. The variance increases exponentially as the load increases and this may lead to unacceptable jitter for delay sensitive applications.

An attractive feature of cyclic service policies is their inherent fairness to all ports, although it adds to the implementation complexity. The analytical results for the average delay in cyclic service without fanout splitting have a worst case error of around 20% in comparison with the simulation results. The worst case error in the sustainable throughput with cyclic service without fanout splitting is for a 4×4 network and is about 9%.

The copy network with cyclic service with fanout splitting scheduling policy has been modeled as a discrete time GI/D/N queue. In Section D it was argued that this is a good approximation although this is valid only for the first moment of the delay. Comparison with simulation results validate the approximation with the worst case error in the mean delay of less than 3% at extremely high loads of 0.98. The sustainable throughput in this copy scheduling policy is 1, the best amongst all the scheduling policies considered. The work conserving nature of this policy leads to the lowest overflow probabilities amongst the scheduling policies considered with small buffer spaces sufficing to accommodate heavy loads.

Table 4 lists the sustainable throughput of the copy network under both the cyclic and the acyclic scheduling policies without fanout splitting for various switch sizes. Observe that under random copy requests with a mean of 2, a 32×32 switch has a 90% sustainable throughput under this policy as compared to the 77% from the acyclic policy.

We wish to mention here that the traffic into a switch will most likely be a combination of unicast and multicast packets. In such a case, we could either separate the two kinds of packets and send only the multicast packets through the copy networks or we could send both of them into the copy networks with the unicast packets requesting only one "copy". In the former case, our analysis follows. To account for the mix of unicast and multicast traffic, we can use a copy request distribution that is derived as follows. Let the fraction of the total arrivals at port i which is unicast be given by ϵ_i. Also, let the probability mass function of the number of copies requested by a multicast packet at port i be denoted by $g_i(k)$. Then, $q_i(k)$, the probability that a packet in input port i requests k copies is given by,

$$q_i(k) \equiv \text{Prob}\{c_i = k\} = \begin{cases} \epsilon_i, & k = 1 \\ (1-\epsilon_i)g_i(k), & k = 2, 3, \cdots N \end{cases}$$

As before, if X_i denotes the random variable for the number of copies requested by the i^{th} input port regardless of it being active or idle. Then,

$$f_i(x_i) \equiv \text{Prob}\{X_i = x_i\} = \begin{cases} 1-\rho_i, & x_i = 0 \\ \rho_i q_i(x_i), & x_i = 1, 2, \cdots N \end{cases}$$

and

$$\mathcal{F}_i(z) \equiv \sum_{x_i} f_i(x_i) z^{x_i} = (1-\rho_i) + \rho_i \mathcal{Q}_i(z) \qquad (24)$$

We can then use Eqn. 24 in the performance models of the previous sections.

Acknowledgments

The authors would like to thank Prof. H. Bruneel, Prof. B. Steyaert and Prof. S. Wittevrongel of the University of Ghent, Belgium for their useful comments and help. We would also like to thank the reviewers for their suggestions and comments.

References

[1] R. Ahuja, B. Prabhakar and N. McKeown, "Multicast Scheduling Algorithms for Input-Queued Switches," *IEEE Jour on Sel Areas of Commun,* vol. 15, no. 5, pp. 855-866, Jun 1997.

[2] H. Bruneel and B. G. Kim, "Discrete-Time Models for Communication Systems Including ATM," Kluwer Academic Publishers, Boston, 1993.

[3] R. G. Bubenik and J. S. Turner, "Performance of a Broadcast Packet Switch," *IEEE Trans on Commun,* vol. 37, no. 1, pp. 60-69, Jan 1989.

[4] J. W. Byun and T. T. Lee, "The Design and Analysis of an ATM Multicast Switch with Adaptive Traffic Controller," *IEEE/ACM Trans on Networking,* vol. 2, no. 3, pp. 288-298, Jun 1994.

[5] H. J. Chao, "Design and Analysis of a Large-Scale Multicast Output Buffered ATM Switch," *IEEE/ACM Trans on Networking,* vol. 3, no. 2, pp. 126-138, Apr 1995.

[6] X. Chen and J. F. Hayes, "Access Control in Multicast Switching", *IEEE/ACM Trans on Networking,* vol. 1, no. 6, pp 638-649, Dec 1993.

[7] J. F. Hayes, R. Breault and M. K. Mehmet-Ali, "Performance Analysis of a Multicast Switch," *IEEE Trans on Commun,* vol. 39, no. 4, pp. 581-587, Apr 1991.

[8] J. F. Hui and T. Renner, "Queueing Analysis for Multicast Packet Switching," *IEEE Trans on Commun,* vol. 42, pp. 723-731, 1994.

[9] J. F. Hui and T. Renner, "Queueing Strategies for Multicast Packet Switching," *Proc of IEEE Globecom '90,* 1990, pp 1431–1437.

[10] M. J. Karol, M. G. Hluchyj and S. P. Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch," *IEEE Trans on Commun,* vol. COM-35, no. 12, pp. 1347-1356, Dec 1987.

[11] Chin-Tau Lea, "A Multicast Broadband Packet Switch," *IEEE Trans on Commun,* vol. 41, no. 4, pp. 621-630, Apr 1993.

[12] T. T. Lee, "Nonblocking Copy Networks for Multicast Packet Switching," *IEEE Jour on Sel Areas of Commun,* vol. 6, no. 9, pp. 1455-1467, Dec 1988.

[13] S. C. Liew, "A General Packet Replication Scheme for Multicasting With Application to Shuffle-Exchange Networks," *IEEE Trans on Commun,* vol. 44, no. 8, pp. 1021-1033, Aug 1996.

[14] D. Manjunath and B. Sikdar, "Integral Expressions for the Numerical Evaluation of Product Form Expressions Over Irregular Multidimensional State Spaces," *Proc of SPECTS'99,* pp. 326-333, 1999.

[15] F. Sestini, "Recursive Copy Generation for Multicast ATM Switching," *IEEE/ACM Trans on Networking,* vol. 5, no. 3, pp. 329-335, Jun 1997.

[16] J. S. Turner, "Design of a Broadcast Packet Switching Network," *IEEE Trans on Commun,* vol. 36, no. 6, pp.734-743, Jun 1988.

[17] J. S. Turner, "An Optimal Nonblocking Multicast Virtual Circuit Switch," *Proc of IEEE INFOCOM,* pp. 298-305, 1994.

[18] B. Vinck and H. Bruneel, "Delay analysis for single server queues," *Electronics Letters,* vol. 32, no. 9, pp. 802-803, Feb 1996.

[19] M. E. Woodward, "Communication and Computer Networks : Modeling with Discrete-Time Queues," Pentech Press, London,1993.

[20] W. De Zhong, Y. Onozato and J. Kaniyil, "A Copy Network with Shared Buffers for Large-Scale Multicast ATM Switching," *IEEE/ACM Trans on Networking,* vol. 1, no. 2, pp. 157-165, Apr 1993.

Biplab Sikdar (S'98) received the B. Tech degree in electronics and communication engineering from North Eastern Hill University, Shillong, India and the M. Tech in electrical engineering from Indian Institute of Technology, Kanpur, India in 1996 and 1998 respectively. He is currently working towards his Ph.D in the department of electrical, computer and systems engineering of Rensselaer Polytechnic Institute, Troy, NY, USA. His research interests include traffic modeling and the implications of long-range dependence, queueing theory and switch performance evaluation and TCP modeling.

Mr. Sikdar is a student member of IEEE and Eta Kappa Nu.

D. Manjunath (S'86-M'93) received his BE from Mysore University, MS from Indian Institute of Technology, Madras and PhD from Rensselaer Polytechnic Inst, Troy NY in 1986, 1989 and 1993 respectively. He was a Visiting Faculty in the Deptt of Computer and Information Sciences, University of Delaware and a Post Doctoral Fellow in the Deptt of Computer Science, University of Toronto. He was with the Deptt of Electrical Engineering, Indian Institute of Technology, Kanpur during December 1994-July 1998. Since July 1998, he is with the Deptt of Electrical Engineering of Indian Institute of Technology, Bombay since July 1998. His research interests are in communication systems and networks, switch architectures, distributed computing, performance analysis and queueing systems.