

# Performance of Multicasting Closed Interconnection Networks

Cathy W. Chan      Soung C. Liew  
Department of Information Engineering  
The Chinese University of Hong Kong  
Shatin, N.T., Hong Kong

wcchan4@ie.cuhk.edu.hk, soung@ie.cuhk.edu.hk

## Abstract

*This paper examines the application of a simple yet general packet replication scheme to achieve multicasting in closed interconnection networks. The performance of these networks is studied and a general throughput equation is obtained to express the overall network throughput in terms of the average routing delay of the corresponding point-to-point network. The multicast performance results are thus built on top of the previously-established, and often simpler, point-to-point results. Making use of this formula, we investigate the multicast closed shuffle-exchange network in detail. Analytical results are compared with simulation results to obtain further insights into the network operation and ways to improve the network performance.*

## I. Introduction

A simple way to perform multicasting in a closed interconnection network, such as the shuffle-exchange network and the Manhattan-street network, is to divide the process into replication and routing phases. The first phase generates the number of copies required by the multicast connection and the second phase routes the copies to the destinations. This two-phase approach has been widely used in constructing multicast packet switches by cascading a copy network and a point-to-point switch [1, 2, 3, 4]. In the closed network scenario, however, the same network is used to perform packet replication and routing simultaneously. Specifically, an input packet enters the network and duplicates itself in a "chain-reaction" manner until the required number of copies are generated. Each copy is then routed to one of the destinations of the multicast connection. Hence, the network appears to a packet as a copy network in the first part of its journey and a point-to-point routing network in the sec-

ond. One can think of the multicast process as being divided into replication and routing phases in the time domain, rather than spatially as in the open network scenario.

In this paper, we consider closed multicast networks constructed by  $2 \times 2$  switch nodes. In these networks, the network nodes are also the sources and destinations of the packets being transmitted. Each node consists of two components, a processor and a replication switch (Fig. 1). The processor is responsible for extracting arriving packets destined for the node and injecting new packets originating from it when unoccupied links are available. The replication switch performs packet duplication and routing depending on what packets arrive at it.

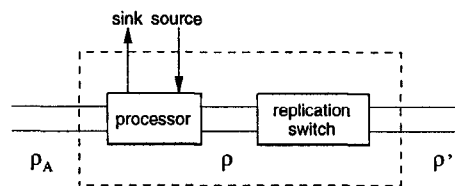


Figure 1: A  $2 \times 2$  switch node.

Packet replication in the closed networks is performed by a simple replication scheme [4] applicable to arbitrary network topologies. Under this scheme, a replicating packet is duplicated at a network node if it is the only packet at the replication switch. Duplication is deferred to the next time slot otherwise. To implement this scheme, each packet has a copy-number ( $CN$ ) field in its header. The  $CN$  value is the number of copies that need to be further generated. When a packet first enters the network, its  $CN$  value is initialized to its fanout request. Each time a replicating packet ( $CN > 1$ ) is successfully duplicated, its

$CN$  value is shared equally between the two resulting packets, one with  $CN = \lceil K/2 \rceil$  and the other with  $CN = \lfloor K/2 \rfloor$ . When a packet's  $CN$  value becomes 1, it enters the routing phase and is directed towards its final destination using the routing algorithm of the network. If two routing packets meet at the replication switch and desire the same output port, one of them is forwarded to the desired link and the other is deflected. If a routing packet meets a replicating one, the routing packet is forwarded to the link it desires and the replicating one to the other link. This is because replicating packets do not have specific destinations and can take arbitrary links. Finally, if two replicating packets meet each other, they are forwarded to the two outgoing links randomly. For both, duplication is deferred to the next node.

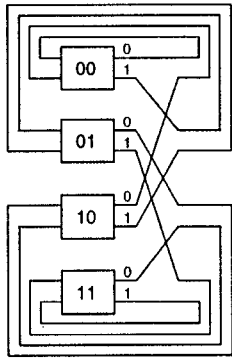


Figure 2: A 4-node closed shuffle-exchange network.

Figure 2 shows a four-node shuffle-exchange network that operates on a time-slotted basis. A spatial representation of the network operation over different times is given in Fig. 3 in which successive stages of the network represent the same network nodes but in successive time slots. Two multicast requests are shown as an example. Packet *A* enters the network from node 10 at time slot 0 and three copies ( $CN=3$ ) are to be delivered to nodes 00, 01 and 11. It is immediately duplicated. The lower copy has  $CN=1$  and starts routing at time slot 1, arriving at its destination at time slot 3. The remaining copy is further split into two copies which start routing at time slot 3 and arrive at their destinations at time slot 5. The other packet *B* enters from node 11 at time slot 1 with  $CN=2$ . Its copies start routing at time slot 3 and reach their destinations at time slot 5.

This simple example shows that in closed networks,

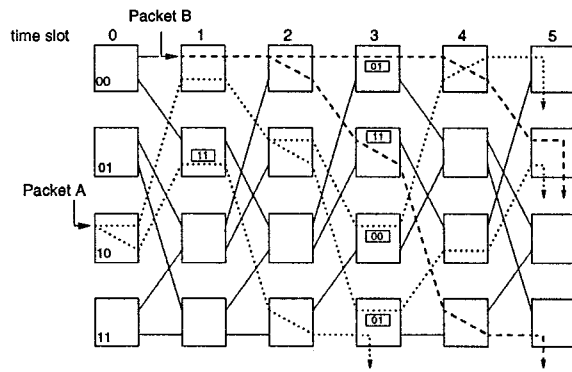


Figure 3: Applying the replication scheme to a closed shuffle-exchange network.

the replication phases of different packets and copies can be of different lengths. This is in contrast to the fixed replication time in the open copy networks [1, 2, 3, 4] in which every packet has to traverse all the stages of the entire copy network before they can be routed to their destinations. The flexibility in the replication time allows packets to start routing and exit the network as soon as possible, freeing more links for use by other packets.

Incorporating packet replication and routing in the same physical network results in more packet interactions and complicates the analysis of the multicast networks. We show in Section II that a general equation can be found to relate the overall network throughput with the average delay of the routing phase. The performance analysis of the multicast networks can then be built on top of the point-to-point routing analysis of the same network and it is much simplified. Based on these results, we study the multicast shuffle-exchange network in Section III and examine performance issues such as network stability and throughput penalty of the general multicast networks. Simulation results and their implications are shown in Section IV and the paper is concluded in Section V.

## II. General Throughput Equation

In this section, we derive a general equation describing the throughput of the closed multicast networks that are built on  $2 \times 2$  switch nodes and operate on a time-slotted basis. The analysis can be extended to networks constructed by switch nodes of other degrees.

To simplify the analysis, we assume that the packets in the network are independent of each other and that

the destinations of packets are distributed uniformly over all the nodes. These two assumptions are commonly used to simplify analysis of point-to-point routing in interconnection networks [7, 8] and their validity is generally accepted. We will adopt these assumptions in our analysis of the multicast network. Simulation results to be presented later show that these assumptions are justified.

Consider a network with  $N$  nodes, each having the structure shown in Fig. 1. Let  $\rho_A$ ,  $\rho$  and  $\rho'$  denote the link loading at each input of the processor, each output of the processor and each output of the replication switch, respectively. Furthermore, let  $\bar{F}$  be the expected fanout of a packet and  $\Lambda$  be the input load to the network, the average number of new packets entering the network per time slot. The average output load of the network is the expected number of routing packets reaching their destinations in a time slot and is equal to  $\bar{F}\Lambda$ . The relationship between these parameters can be obtained by inspecting the network operations in each time slot.

*Injection and removal of packets* : At equilibrium, in each time slot, there is on average a total of  $2N\rho_A$  and  $2N\rho$  packets entering and leaving the processors of the  $N$  nodes, respectively. Since an average of  $\bar{F}\Lambda$  packets arrive at their destinations and are removed, and an average of  $\Lambda$  packets are introduced into the network at the processors, we have

$$2N\rho = 2N\rho_A - \bar{F}\Lambda + \Lambda. \quad (1)$$

*Replication of packets* : At the output of the replication switch, there are either zero, one or two packets at the two links, depending on the number of packets at the inputs. If no packets enter the replication switch, no packets emerge at the outputs. If there is a single routing packet at the inputs, there is one packet at the outputs. There can be two packets at the outputs in two cases: either there are two packets at the inputs or there is one replicating packet only and it is duplicated in the switch. Let  $P_r$  be the packet replication probability, the probability that a packet at the input link of the replication switch is a replicating packet. The expected number of packets at the two outputs of the switch is given by

$$2\rho' = 0 \cdot (1 - \rho)^2 + 1 \cdot 2\rho(1 - \rho)(1 - P_r) + 2 \cdot [\rho^2 + 2\rho(1 - \rho)P_r],$$

which simplifies to

$$\rho' = \rho + \rho(1 - \rho)P_r.$$

At equilibrium, all packets at the outputs of the replication switches are fed back to the inputs of the processors. Hence,  $\rho_A = \rho'$  and (1) reduces to

$$P_r = \frac{(\bar{F} - 1)\Lambda}{2N\rho(1 - \rho)} \quad (2)$$

which relates the packet replication probability at the input of the replication switch to the input load and the link loading of the network.

*Routing of packets* : At any replication switch, deflection occurs only if there are two packets present and they desire the same outgoing link. Since replicating packets are free to emerge at any node at the end of their replication process, they do not have any desired outgoing link at the replication switches. Thus, as far as routing and deflection is concerned, they can be ignored and only routing packets need to be considered. The routing phase, therefore, has link loading  $\rho_s = \rho(1 - P_r)$ , called the effective link loading, and throughput  $\bar{F}\Lambda$ . The routing delay is the delay incurred from the time a packet is generated from the replication process (and begins the routing phase) to the time it reaches its final destination. Denote the average routing delay by  $D$ . Applying Little's law on the routing process, we have

$$2N\rho(1 - P_r) = \bar{F}\Lambda \cdot D \quad (3)$$

Substituting (2) into (3) and rearranging gives the general throughput equation:

$$\Lambda = \frac{2N\rho(1 - \rho)}{\bar{F} - 1 + \bar{F}(1 - \rho)D} \quad (4)$$

Equation (4) states that the input load  $\Lambda$  (and hence, the throughput) of a multicast network at a given link loading is related to the network topology through the delay term  $D$  only. For most networks,  $D$  is a function of the network size  $N$  and the effective link loading  $\rho_s = \rho(1 - P_r)$ , which by (2), is in turn related to  $\bar{F}$  and  $\Lambda$ . Thus, we can write

$$D = D(N, \rho, \Lambda, \bar{F}). \quad (5)$$

The exact expression of (5) can be obtained from the analysis of the point-to-point network. By solving it and (4) simultaneously, the network throughput at a particular link loading can be found. Although closed-form solutions may not always exist, the two equations can be solved numerically to obtain the performance of the multicast network.

### III. The Multicasting Closed Shuffle-Exchange Network

We have seen that the performance of the multicast network is related to its topology through the average routing delay of the network. This delay is determined by the routing algorithm, the specification of which includes how packets choose their outgoing links at a switch node and how output port conflicts between packets are resolved. We illustrate how the analysis is completed for the shuffle-exchange network in this section.

In a shuffle-exchange network with  $N=2^n$  nodes, routing is performed by utilizing the destination address bits  $d_n d_{n-1} \dots d_1$  as the routing tag [5]. The routing tag specifies the preferred outgoing links of the packet as it traverses the network, starting from  $d_n$  to  $d_1$ . When two packets at a node prefer the same link, one is chosen to be deflected to the unfavorable link and has to restart the routing process in the next time slot. Two strategies are commonly used in selecting the winning packet. The random contention resolution scheme chooses a packet randomly and the distance-priority scheme gives priority to the packet closer to its destination. It is known that the latter scheme improves the performance of the point-to-point shuffle-exchange network over the former in terms of achievable throughput and network stability [6, 7]. We will investigate whether this improvement carries through to the multicast scenario when these schemes are applied to the routing phase.

#### A. Random Contention Resolution Scheme

In the  $2^n$ -node shuffle-exchange network, when contention is resolved by randomly deflecting a packet, the routing delay at link loading  $\rho$  is given by [6]

$$D = \frac{1 - (1 - q)^n}{(1 - q)^n q}$$

where  $q = \frac{1}{4}\rho$  is the deflection probability of a routing packet. Recall that the link loading of the routing phase in the multicast shuffle-exchange network is equal to the effective link loading  $\rho_s$ . The average routing delay is therefore

$$D = \frac{1 - (1 - \frac{1}{4}\rho_s)^n}{(1 - \frac{1}{4}\rho_s)^n \frac{1}{4}\rho_s} \quad (6)$$

The throughput of the multicast shuffle-exchange network at a particular link loading  $\rho$  can be found

by iterations. Initially, an arbitrary value is assigned to  $\Lambda$ . By (2),  $P_r$  is found and  $\rho_s (= \rho(1 - P_r))$  is substituted into (6) to obtain  $D$ . Finally, a new value of  $\Lambda$  is computed using (4). This process is repeated until  $\Lambda$  converges.

The results for such a multicast shuffle-exchange network with 256 nodes and an average fanout of 8 are shown in Fig. 4 and 5. Figure 4 plots the packet replication probability  $P_r$  and the average routing delay  $D$  against the link loading. In Fig. 5, the per-node throughput, or the average number of packets arriving at their destinations per node per time slot, is given by the average output load  $\bar{F}\Lambda$  divided by  $N$ . The throughput of the same shuffle-exchange network used as a unicast routing network is shown in dashed line for comparison.

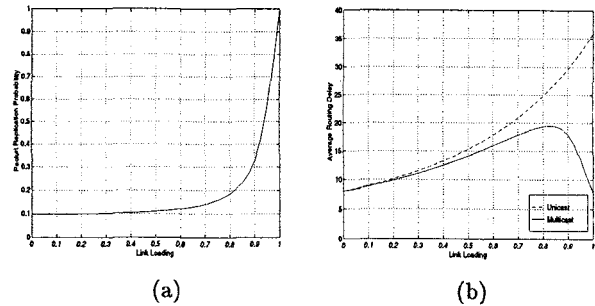


Figure 4: (a) Packet replication probability and (b) routing delay of a multicast shuffle-exchange network using random contention resolution ( $N=256$ ,  $\bar{F}=8$ ).

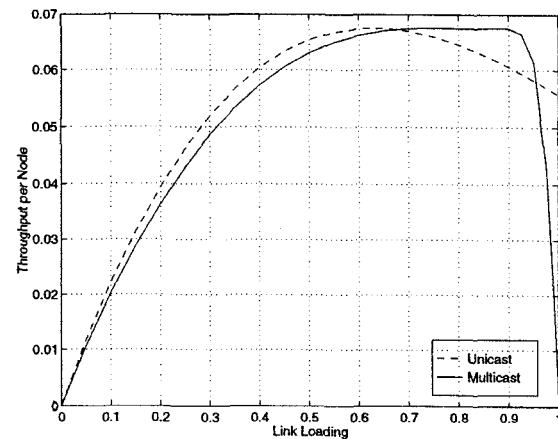


Figure 5: Unicast and multicast throughput of a 256-node shuffle-exchange network using random contention resolution.

At light load, the multicast throughput is very sim-

ilar to the unicast throughput. In this region, there are very few packets in the network and replication processes are completed quickly. The fraction of replicating packets remains small (Fig. 4(a)). Hence, the multicast network behaves like a unicast network for  $\rho \leq 0.6$ .

As  $\rho$  increases, the packet replication probability  $P_r$  begins to increase with the link loading, giving rise to a smaller effective link loading  $\rho_s$ . For the same overall link loading  $\rho$ , there are actually fewer routing packets in the multicast network than in the unicast one. Contention between packets in the network is less severe in the multicast network and packets can be delivered with a smaller routing delay (Fig. 4(b)). Hence, the multicast throughput is higher than the unicast one.

When the link loading becomes very high, most of the packets in the network are replicating. Routing packets experience fewer conflicts and can reach their destinations in a shorter time. However, due to the large number of packets in the network, packet duplication is very difficult and the number of routing packets generated is small. On one hand, we expect the throughput to increase because of the smaller routing delay. On the other hand, we also expect the throughput to decrease because of the small number of routing packets in the network. The actual throughput is determined by the relative effects of the two opposing factors and, as indicated in Fig. 5, it is found to be decreasing with increasing  $\rho$  in the high load region.

In the extreme case that the network is saturated, all packets in the network are replicating and none of them can have a chance to replicate or leave the network. These packets keep circulating in the network and no new packets can enter. This is a deadlock situation in that the overall status of the network remains unchanged over time. Since no routing packets can be obtained from replication, the saturation throughput of the network is zero.

The overall throughput of the multicast shuffle-exchange network is thus an increasing function of the link loading at light load and a decreasing function at high load. In general, for any networks, operation in regions in which the throughput decreases with link loading is unstable [6], meaning that any attempt to operate the network in such a region will evolve to a stable operating point either at the beginning or at the end of the unstable region. In our network, this sta-

ble operating point is the saturation point where the network is deadlocked: all links are occupied by replicating packets ( $\rho=1$  and  $P_r=1$ ) and the throughput is zero.

Note that this instability is not specific to the shuffle-exchange network but is a property common to all closed multicast networks that perform packet replication and routing in the same physical network. This is revealed by the throughput equation (4). The overall throughput of the multicast network is zero at  $\rho=0$  and  $\rho=1$ . This requires the throughput to be an increasing function of the link loading in some regions and a decreasing function in others, in which the network would be unstable. This is independent of the value of the routing delay and hence, independent of the network topology and routing algorithm used. Consequently, the multicast networks can easily become deadlocked and useless when operating in the high load region. One possible solution for breaking deadlocks will be discussed in the next section.

Another performance issue we are interested in is whether adding multicast support to the closed networks results in any improvement or penalty in network performance in terms of the maximum achievable throughput. Intuitively, the introduction of the replication process uses up part of the network resources. It is impossible for the multicast network to have a better performance than the unicast network, which is purely responsible for routing packets to their destinations. This can be verified by observing that the overall throughput of the multicast network at a particular link loading  $\rho$  is equal to the throughput of the embedded routing process with effective link loading  $\rho_s = \rho(1 - P_r)$ . This routing throughput is in turn equal to the throughput of the same unicast network operating at  $\rho_s$ . Thus, the multicast throughput at any link loading must be achievable by the unicast network and the maximum multicast throughput must be smaller than (or equal to) the maximum unicast throughput.

Figure 5 shows that for the shuffle-exchange network, the maximum multicast throughput is approximately equal to the maximum unicast throughput. This indicates that the introduction of multicast connections does not result in any throughput penalty in the shuffle-exchange network that resolves conflicts by the random contention resolution scheme.

## B. Distance-Priority Contention Resolution Scheme

When output port conflicts in the shuffle-exchange network are resolved by the distance-priority contention resolution scheme, the network throughput and average routing delay are given by [6]

$$\Lambda = 2N\pi(1) \left(1 - \frac{\pi(1)}{4}\right),$$

$$D = \frac{2N\rho_s}{\Lambda} = \frac{\rho_s}{\pi(1)(1 - \frac{\pi(1)}{4})} \quad (7)$$

where  $\pi(1)$  is obtained by iteratively solving the following recursive definition of  $\pi(i)$  [6]

$$\pi(i) = \begin{cases} \pi(i+1) - \frac{\pi^2(i+1)}{4}, & 1 \leq i \leq n-1 \\ -\pi(1) + \frac{\pi^2(1)}{4}, & i = n. \\ \rho_s, & i = n. \end{cases} \quad (8)$$

When this routing delay is used in the iterations for computing the throughput of the multicast network, we obtain the results shown in Fig. 6. As before, the network has 256 nodes and the solid line shows the multicast throughput where the average fanout is 8 and the dashed line shows the unicast throughput. Compared with the random scheme, the distance-priority scheme can achieve a higher network throughput because of the reduction in routing time. This improvement is the same as in the unicast shuffle-exchange network. But unlike the point-to-point scenario, the multicast network remains to be unstable when the distance-priority scheme is adopted. This is in agreement with our previous observation that instability in multicast networks is independent of the routing algorithm and contention scheme used.

Figure 6 also shows that the maximum unicast throughput of the shuffle-exchange network using distance-priority contention resolution is achieved when the network links are all occupied by routing packets. Due to the presence of replicating packets, this can never be achieved in the multicast network. Hence, the maximum multicast throughput is smaller than the maximum unicast throughput. In general, throughput penalty results when the proportion of replicating and routing packets in the multicast network always gives rise to an effective link loading smaller than the link loading required to achieve the maximum throughput in the unicast network. For other networks, the introduction of multicast connections does not reduce the maximum achievable throughput of the network.

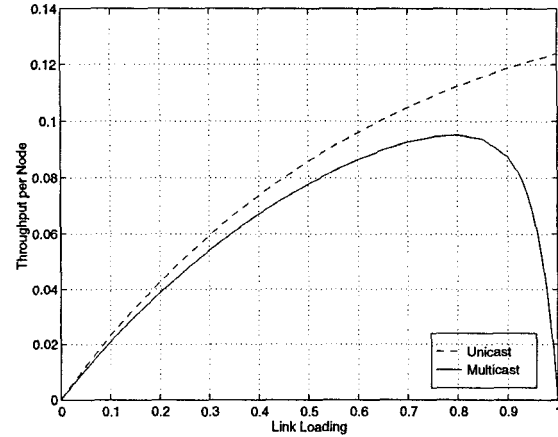


Figure 6: Unicast and multicast throughput of a 256-node shuffle-exchange network using distance-priority contention resolution.

## IV. Simulation Results

In this section, we show the simulation results for the multicast shuffle-exchange network. In our simulations, packets generated at each source node are stored at an input queue outside the processor. A packet is injected into the network if at least one of the output links of the processor is empty. At each time slot, a packet is generated at a node with probability  $\Lambda_o/N$ , where  $\Lambda_o$ , the total number of packets generated per time slot, is known as the offered load of the network. The offered load is varied to simulate the network under different link loadings. In each simulation run, statistics on the number of packets in and leaving the network is collected only after the network reaches equilibrium. These results are averaged to obtain the link loading and per-node throughput of the network on a per time slot basis. The throughput is plotted against the link loading and compared with the analytical results.

The fanout requests  $F$  of the packets are truncated-geometrically distributed:

$$P\{F = k\} = \frac{(1-p)p^{k-1}}{1-p^{N-1}}, \quad 1 \leq k \leq N-1 \quad (9)$$

with average fanout

$$\bar{F} = \frac{1}{1-p} - \frac{(N-1)p^{N-1}}{1-p^{N-1}}$$

where the parameter  $p$  is varied to generate the desired average fanout. We assume that no multicast connection requires copies to be sent to its source node so that the maximum fanout is  $N-1$ . The minimum

fanout is one and corresponds to point-to-point routing packets.

The distribution of the destination addresses, on the other hand, is uniform. All nodes, except the source node, are equally likely to be chosen as one of the destinations of a multicast connection.

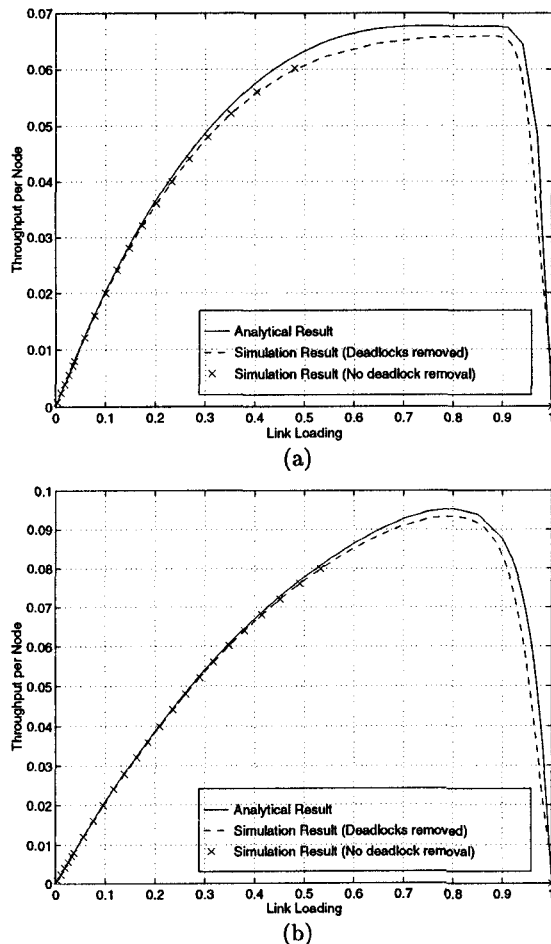


Figure 7: Simulation results of the multicast shuffle-exchange network using (a) random and (b) distance-priority contention resolution.

The simulation results for the multicast shuffle-exchange network are shown in Fig. 7. The crosses represent the results obtained from simulation and the solid lines are the analytical results. The throughputs achieved at small link loadings ( $0 \leq \rho \leq 0.5$ ) agree with those predicted by numerical analysis except that the simulated values are slightly smaller than the analytical values. This discrepancy is due to the independence assumption used in the analysis. In sim-

ulations and actual implementations, packets may be correlated in such a way that there is a higher level of contention among them, causing the results to deviate slightly from the analytical ones.

As the multicast throughput approaches the maximum value (where  $0.5 < \rho < 0.9$ ), no data points can be obtained. In this region, any slight variations in the offered load or throughput can easily shift the operating point to the negative-slope area ( $0.9 < \rho < 1$ ), in which the network runs into saturation very quickly. Consequently, network operation cannot maintain equilibrium in either the “close-to-maximum” or the negative-slope regions and no data can be collected for  $\rho > 0.5$ .

To prevent the network from becoming useless due to deadlocks, we employ a deadlock-breaking mechanism in the network. An age counter is added to the header of each replicating packet. The age counter is incremented in each time slot, regardless of whether duplication is successful. When the age of a replicating packet reaches the lifetime limit  $T_{max}$ , the packet is discarded. In this way, old packets are removed and the links can be freed for duplication or injection of new packets. Hence, packet circulations are eliminated and deadlocks do not exist in the network.

The dashed lines in Fig. 7 show the results of applying this deadlock-breaking mechanism using  $T_{max}=40$ . At light loads, the results are similar to those obtained previously. When the network load increases, we are able to obtain results in the “unstable” regions. By removing some of the replicating packets and their embedded copies, the rate at which packets (and copies) leave the network is large enough to balance the rate at which they enter the network. Packet accumulation does not occur and the network can, therefore, operate at equilibrium in regions in which it was previously unstable. Thus, the network is now stable at all link loadings.

By varying  $T_{max}$ , it is found that the deadlock-breaking mechanism does not guarantee network stability. Figure 8 shows the results at different  $T_{max}$ . When the limit is small ( $T_{max}=10$ ), the network throughput is much smaller than the analytical throughput. The replicating packets are removed so soon that many replication requests are not performed, leading to a smaller throughput than the network can achieve. On the other hand, when the limit is too large ( $T_{max}=200$ ), the network becomes unsta-

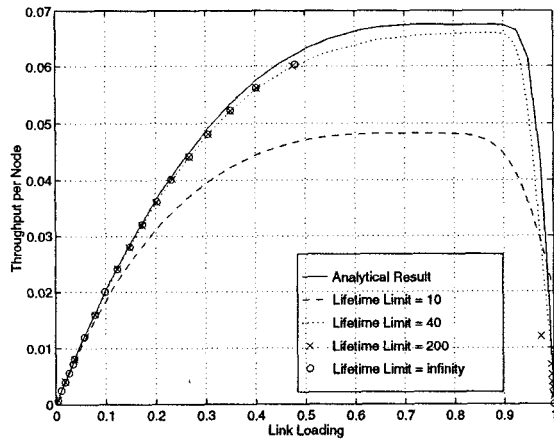


Figure 8: Simulation results of the multicast shuffle-exchange network using deadlock-breaking mechanism with different lifetime limits.

ble again. Because the replicating packets are allowed to stay in the network for too long, the network is easily saturated with replicating packets. Although the network is now deadlock-free (packets are removed from time to time), it cannot operate at equilibrium at the high load region. This is similar to when no lifetime limit is imposed ( $T_{max} = \infty$ ). We conclude that the deadlock-breaking mechanism can stabilize network operation only when a suitable lifetime limit is chosen.

A final remark is that the deadlock-breaking mechanism is not a good solution to the instability problem due to inevitable packet loss. Discarding packets may lead to retransmission by the upper layers, thereby increasing the load of the network. Carefully designed access control schemes should be used to prevent instability while maintaining the performance of the closed multicast networks.

## V. Conclusions

In this paper, we have extended the general packet replication algorithm to closed interconnection networks. We showed that the performance of this kind of multicasting networks can be summarized by a general throughput equation. With this equation, any closed multicast networks can be analyzed based on the analysis of point-to-point routing in the network. We investigated the closed shuffle-exchange network using this method as well as by simulation. By comparing the results, we addressed some performance issues of

the closed multicast networks. An important property common to these networks is network instability which cannot be overlooked in the design of these networks.

## References

- [1] T.T. Lee, "Nonblocking Copy Networks for Multicast Packet Switching," *IEEE JSAC*, Vol. 6, No. 9, Dec. 1988, pp. 1455-1467.
- [2] A. Huang and S. Knauer, "Starlite: A Wideband Digital Switch," *IEEE GLOBECOM '84*, 1984, pp. 121-125.
- [3] J.S. Turner, "Design of a Broadcast Packet Switching Network," *IEEE INFOCOM '86*, 1986, pp. 667-675.
- [4] S.C. Liew, "A General Packet Replication Scheme for Multicasting with Application to Shuffle-Exchange Networks," *IEEE Trans. on Commun.*, Vol. 44, No. 8, Aug, 1996, pp. 1021-1033.
- [5] D.H. Lawrie, "Access and Alignment of Data in an Array Processor," *IEEE Trans. on Computers*, Vol. C-24, No. 12, Dec. 1975, pp. 99-109.
- [6] S.C. Liew, "On the Stability of Shuffle-Exchange and Bidirectional Shuffle-Exchange Deflection Networks," to appear in *IEEE Trans. on Networking*.
- [7] A. Krishna and B. Hajek, "Performance of Shuffle-Like Switching Networks with Deflection," *IEEE INFOCOM '90*, 1990, pp. 473-480.
- [8] A.G. Greenberg and J. Goodman, "Sharp Approximate Models of Deflection Routing in Mesh Networks," *IEEE Trans. on Commun.*, Vol. 41, No. 1, Jan. 1993, pp. 210-223.