Multicast broadband cell switching using the recycling method[1]

Daniel F. Hall, Jamil Y. Khan, and Steven R. Weller

School of Electrical Engineering and Computer Science

The University of Newcastle

Callaghan, NSW 2308, Australia.

Email: c9316048@studentmail.newcastle.edu.au,

Jamil.Khan@newcastle.edu.au,

steven.weller@newcastle.edu.au

Abstract

Electronic cell switches are widely used to interconnect computers, processing elements in parallel supercomputers, and line cards in high-speed routers. A shortcoming of cell switches is that they generally cannot support multicast connections without sacrificing either performance or scalability. One promising technique for constructing low complexity, highly scalable multicast cell switches is cell recycling. This paper considers a class of recycling multicast switches which use binary copying to offer complexity comparable to that of unicast switches. The delay and jitter performance of a typical binary-copying architecture is examined under uniform and non-uniform (bursty) traffic. The relationship between the number of times a cell is recycled and the quality of service it receives is determined. Finally, measures to improve performance and fairness in recycling switches are presented. This paper shows that multicast cell switches can be constructed without compromising performance or scalability.

1 Introduction

In a network which supports multicast, hosts can generate a single cell addressed to multiple destinations (called the multicast group), which the network then transparently replicates and delivers. This is far more efficient than forcing hosts to generate and address every copy of a multicast cell themselves. Whether a cell switch supports multicast is therefore an important consideration, particularly given the proliferation of multicast teletraffic generated by broadcast-reliant LAN protocols, audio/video distribution, and high-performance computing clusters. Apart from facing the familiar tradeoffs between performance, hardware cost, and scalability, designers of multicast switches have to choose whether to adapt existing, well understood, unicast designs or turn to specialised multicast switch architectures.

Multistage interconnection networks (MINs) are constructed using small crossbar or shared-memory switches (known as switching elements, or SEs) grouped into stages, with interstage connections providing the connectivity between the outputs of one stage to the inputs of the subsequent stage. Input cells are then routed towards their correct switch outputs stage-by-stage, with the SEs of each stage only required to make local routing decisions. The exceptional scalability provided by MINs has made them a natural candidate for the construction of high-bandwidth switches.

Cell recycling, described in Section 3, has been proposed as a method of adding multicast capability to unicast MINs with a minimum of overhead [1, 2, 3]. In cell recycling, the MIN is required only to make a small number of copies of an input cell. If a greater number of copies are requested, some output cells are recycled back to the switch input side. A criticism frequently leveled at recycling multicast switches is that while they may have low complexity – often not much greater than that of an equivalently sized unicast switch – the more times a cell is recycled, the greater the number of sequential MIN crossings it makes before exiting the switch, and the greater overall delay, delay variance (jitter), and cell loss probability it experiences.

In this paper we study binary copying recycling switches [1], so called because they generate only two copies of a cell in each MIN pass. In Section 4 we compare the efficiency of binary-copying recycling switches based upon the number of internal interstage links each requires to support an arbitrary multicast connection. Section 5 presents new simulation results for a class of binary copying switches which copy cells only in the final MIN stage. This is the simplest binary copying architecture, allowing the construction of both unicast and multicast switches using almost identical MINs. Finally in Section 6 we propose and test techniques which improve the quality of service and fairness experienced by recycled cells. Our work, which shows how fair and scalable multicast switches can be constructed with complexity comparable to unicast switches, has potential application to asynchronous transfer mode (ATM) switches, switching fabrics used to connect line cards in high-speed IP routers [4], and specialised interconnects in highly parallel computers [5].

2 MIN-based Multicast Switches

Multicast broadband switches are required to perform two tasks: cell replication and cell addressing/routing. An input multicast cell must be replicated a number of times equal to the size of the multicast group (the fanout) of the connection. These copies then have to be delivered to their intended switch outputs. How these tasks are performed is largely governed by the type of switching network (SN) used to deliver cells from the switch inputs to the switch outputs. For MIN-based SNs, there exist three approaches to multicast cell delivery: cell replication before routing (CRBR) [6, 7], cell replication while routing (CRWR) [2, 8], and cell recycling (CR) [1, 3]. CRBR cascades a specialised copy network (CN) with a conventional unicast routing network (RN). The copy network used in CRBR switches increases switch complexity and, unless non-blocking, also degrades switch performance. Instead of maintaining separate copying and routing networks, CRWR switches (Fig. 1(b)) integrate both functions into a single SN. This represents a considerable departure from unicast SN designs, requiring the development of new routing algorithms to control both copying and routing processes.

Recognising that the complexity of CRWR switches is primarily a function of the number of cell copies

they must create in a single SN crossing, the SN in recycling switches create comparatively few cell copies. If a greater number of copies are required, some cell copies are then recycled back to the SN inputs where they can go on to generate further copies (Fig. 1(c)). Several sequential recycling operations may be required to create all the copies necessary. The complexity of recycling switches is much closer to that of unicast switches than can be achieved using either CRBR or CRWR. Also, by breaking up the process of copying high-fanout multicast cells over several SN passes, recycling switches avoid the SN congestion problems created in CRBR and CRWR switches when large numbers of cell copies cross the SN at the same time.

3 Recycling Switches

Each virtual circuit (VC) in a $N \times N$ recycling switch is associated with a multicast tree describing the sequence of copying and recycling operations that an input cell undergoes before reaching each output in the associated multicast group $\mathcal{D}$. In a multicast tree, nodes represent SN output ports where cells can be directed to a switch output port, recycled back to a SN input port, or both. The edges between nodes represent crossings from a SN input port to a SN output port, with the maximum node out-degree M the multiplication factor of the SN, or the number of copies of a cell that can be made in a single SN pass. The node depth d ($1 \leq d < \lceil \log_M(MN + M - N) \rceil$) represents the number of SN crossings an input cell must make before reaching the node. Figure 2 shows the multicast tree for the multicast VC $\mathcal{D} = \{0, 2, 3, 4, 5, 6, 7\}$ and its associated routing in a 8×8 SN with $M = 2$. The remainder of this section will discuss approaches to cell addressing and routing in recycling switches.

3.1 Cell addressing and routing

There are few restrictions on the type of MINs which can be used in recycling designs, beyond the basic requirement that they be able to both copy and route cells. This flexibility has led to a variety of MINs being proposed for use in recycling switches. A natural choice are Banyan and Banyan-like[2] MINs because

[2]MINs isomorphic (topologically equivalent) to the Banyan network

of their ability to use self-routing, where the $b \times b$ SEs in each of the n switching stages ($n = \log_b N$) make routing decisions solely based upon a single digit in the base-b representation of the cell's destination port. Self-routing is an attractive feature because no centralised routing control mechanism is needed. However, in recycling switches with $M \geq 2$ some changes are required to the traditional self-routing algorithm to allow the path to more than one destination to be described. The end result of these changes are two multicast routing schemes: generalised self-routing and cube-encoding.

Generalised self-routing (GSR) [7] places each of the n-bit destination addresses into the cell header. Routing for a cell arriving at an SE in stage k is determined by considering the k-th bits of all destination addresses in the header. If they are all 0 or all 1, then the cell will be sent out on link 0 or link 1 respectively without modifying the header[3]. Otherwise, the cell and its copy are sent out on both links, with the destination addresses in each header modified such that the header of the cell copy sent out on link 0 (link 1) contains those addresses in the original header with the k-th bit equal to 0 (1).

First proposed by Chen and Kumar [8], cube encoding uses two n-bit tags, R and C, which are used to control cell routing and copying, respectively. Routing is performed in each MIN stage i as follows. If C_i = 0 then the cell is sent out on link 0 or 1 depending on the value of R_i. Otherwise C_i = 1, and then the cell is copied out of both output links. The set of MIN outlets which can be addressed with a given R,C pair is called a cube. If the output addresses of an input multicast cell cannot be encoded into a single cube, then it must be broken up into smaller cubes each of which address a subset of the required outputs.

3.2 MIN Architectures

Unbuffered MINs are constructed from SEs which contain no internal buffering. If multiple SE input cells contend for a single SE outlet, one is selected for transmission and the others discarded. Unbuffered MINs are appealing because of their low complexity, however effectively dealing with internal blocking arising from multicast cell copying and routing can be a great challenge. The two major classes of unbuffered recycling switches, two-pass and multi-pass, adopt different approaches to avoiding internal blocking.

[3]The subsequent discussion will assume binary ($b = 2$) SEs, however the techniques presented can be easily generalised to SEs of larger size.

Two-pass recycling switches [9, 8] emulate the operation of CRBR switches using a single MIN, with the copying and routing functions performed in two consecutive MIN passes. Each of these two passes are designed to be nonblocking. Multi-pass switches [8] begin by constructing a multicast tree using the minimum subset of cubes sufficient to cover $\mathcal{D}$, subject to the constraint that cubes that would cause internal blocking are not routed in the same MIN pass. Both two- and multi-path recycling switches require excessive network resources, such as complex SEs and/or a large number of internal links, in order to eliminate blocking within a multicast connection.

The SEs in buffered MINs contain buffers on either their input or output side to store cells which lose contention. Recycling switches can use a great variety of buffered MINs, including Beneš, Clos, and Banyan [10, 3, 8, 1, 11]. These switch designs focus on reducing translation memory size and MIN complexity rather than reducing internal blocking. The simplest buffered recycling switch configuration consists of a purely unicast MIN with cell copying performed at the MIN outlets. One of these copies is then recycled, with the other exiting the switch. The price paid for this low MIN complexity is that a maximum of f sequential recycling passes are required to create the $f = |\mathcal{D}|$ copies of an input multicast cell. If the switch creates two or more copies per switch pass, then the number of copies created of the input cell increases exponentially with the number of recycling passes. This exponential cell generating process ensures that a large number of cell copies can be created in a small number of MIN passes. One particularly interesting recycling architecture uses a multiplication factor of two, thereby enjoying an exponential cell generating process while maintaining low MIN complexity. We will refer to this configuration as binary copying.

4 Binary Copying Switches

Cusani and Sestini [1] described the first recycling switch to use the binary-copying method. Their work was later extended by Sestini [10] who presented additional architectural details (Fig. 3) and results. The multicast MIN used in their design, constructed using input-buffered 2×2 SEs arranged in an extended-delta topology, is identical to a unicast MIN in all but one respect: the SEs in the final stage are capable

of replicating input cells to both of their outputs if the copy bit in the cell's routing tag is set. The routing tag, which controls cell copying, routing, and recycling for a single switch crossing, is provided to input cells by the Input Port Processors (IPP) and recycled cells by the Multicast Port Processors (MPP). The Input Concentrators act as input buffers, concentrating and storing input and recycled cells until they can be serviced by the MIN. At the MIN outputs, the Output Selectors (OSs) can pass cells to the switch outputs, recycle them back to the switch inputs via recycling lines so additional outputs can be addressed, or both.

The Washington University Gigabit Switch (WUGS) [3] represents the most comprehensive implementation of a multicast virtual circuit switch to use the cell recycling method. The WUGS switch uses a Beneš MIN built using 8×8 shared buffer SEs. Although also a binary copying switch, the WUGS design differs from Cusani and Sestini's proposal in that copying can occur in any switch stage, not just the final stage. Consequently, a more complex routing scheme must be employed, in this case GSR.

Subramaniam [11] proposed a binary copying recycling architecture which, like the WUGS design, allows cell replication to occur in any MIN stage. Subramaniam's switch differs from the WUGS design in that it uses cube-encoding rather than generalised self-routing.

4.1 Performance of binary copying switches

The binary copying recycling switches described in the previous section use different routing techniques: GSR (WUGS), cube-encoding (Subramanian), and cube-encoding using a single copy-bit (Cusani and Sestini). We term these three architectures generalised self routing binary copying (GSRBC), cube-encoded binary copying (CEBC), and restricted cube-encoded binary copying (RCEBC), respectively. One method of comparing these routing schemes is the total number of interstage links l they require to support a given multicast connection. This provides a measure of how efficiently the MIN can support the multicast connection: a multicast tree with a lower l will require fewer interstage crossings, resulting in lower MIN congestion and improved performance. The value of l will be largely determined by the number of low-weight sibling pairs which can be formed, a sibling pair being defined as two outputs which can be reached in a single MIN crossing.

For an n-stage binary multicast switch, define the weight w of a sibling pair as the switch stage in which

binary copying occurs when the switch stages are labeled from n (switch input stage) to 1 (switch output stage). (A node with no sibling does not require copying, so has a weight of 0.) The number of interstage links traversed by an MIN input cell in reaching its output(s) is equal to $n+w$, where w is the weight of the sibling pair. In an $N \times N$ self-routing MIN, the number of outlets which could form a sibling relationship with a given outlet is equal to $N-1$, $\log_2 N - 1$ and 1 under GSRBC, CEBC, and RCEBC respectively. The increased capacity of GSRBC to form sibling pairs means that, given an arbitrary multicast group $\mathcal{D}$, GSRBC will more likely to form an optimal (in terms of l) multicast tree than either CEBC or RCEBC.

Focusing too narrowly on l as a performance metric is unwise, since the switch stage in which copying occurs also impacts performance. For example, when subject to broadcast input traffic a 16×16 RCEBC switch which copies in the first MIN stage supports a maximum throughput (TP) of approximately 0.15, compared to 0.5 for a recycling switch with $M = 1$, despite multicast trees constructed in both scenarios sharing the same value of l ($N \log_2 N$). To copy a cell, a SE requires that its two outlets be free and that the next-stage SEs these outlets are connected have sufficient buffer space to accept a new cell (cells blocked because of insufficient buffer space in the next stage are said to experience backpressure). Because cells copied in the final switching stage do not experience backpressure, they have a lower blocking probability than cells copied in earlier stages. For this reason RCEBC switches copying cells in the final MIN stage are much closer in performance to GSRBC and CEBC than their corresponding l would suggest. This arrangement, which we will refer to as final-stage binary copying (FSBC), also requires minimal MIN overhead, allowing both unicast and multicast versions of the switch can use a similar, low-cost switching fabric. The performance of the FSBC configuration is investigated in the next section.

5 Performance analysis

In this section we extend the work of Cusani and Sestini [1, 10] for the FSBC switch, most notably by presenting results for delay variance (jitter) and by investigating the relationship between the number of times a cell is recycled and the quality-of-service (QoS) it receives. The former is important for determining how efficiently the switch will carry jitter-sensitive traffic (e.g., real-time MPEG encoded video), with the

latter important for both determining the worst-case multicast QoS and gaining insight into how large FSBC switches will perform.

In OPNET [12] we modeled a 16×16 FSBC switch (Table 1) which used a seven-stage Beneš MIN, with cells randomly routed in the first three (distribution) stages, and self-routed in the final four (routing) stages. The first set of results were obtained using Bernoulli (uniform) unicast and multicast input traffic. Two multicast fanout distributions were considered, both based upon a truncated geometric distribution. More specifically, the probability that an arriving cell has a fanout f is given by:

$$\Pr[f = i] = \begin{cases} \gamma(1-\gamma)^{i-1} & , \quad 1 \leq i < N \\ (1-\gamma)^{N-1} & , \quad i = N, \end{cases} \tag{1}$$

where parameter γ was set to two different values, chosen such that the mean fanout $\bar{f} = 4$ and $\bar{f} = 8$. The two scenarios corresponding to $\bar{f} = 4, 8$ are referred to as "Geo4" and "Geo8" respectively.

Results are reported for mean throughput (Fig. 4(a)) and mean cell delay (Fig. 4(b)) versus offered load for unicast, Geo4, and Geo8 traffic. Since each cell entering the switch may give rise to multiple output cells, the offered load u is defined as the product of the input load and the average fanout. All results are shown using 95% confidence interval, obtained using the method of batch means. Delay and jitter are reported in units of cell slots, or the time it takes to transmit a cell (equal to $2.7\mu s$ for 53 byte cells at 155 Mb/s).

From Fig. 4(a) the maximum switch throughput under unicast, Geo4, and Geo8 traffic is equal to 0.51, 0.56, and 0.67 respectively, thus illustrating a feature of recycling switches with $N \geq 2$: as the mean connection fanout $\bar{f}$ increases, the more outputs will be able to participate in low-weight cubes, the lower the MIN congestion and the greater the switch TP for the same offered load. This explains the results in Fig. 4(b) where, for low to moderate offered loads, delay decreases with increased $\bar{f}$, despite higher $\bar{f}$ requiring on average a greater number of MIN crossings. Further consideration of Fig. 4(a) allows us to determine that the minimum switch speed advantage (the ratio of the internal to external switch speed) required to achieve 100% TP under uniform input loads is approximately 2.0, assuming large (but finite) input buffers.

Switch Size	16×16
SE Size	2×2
No. of Switch Stages	7
SE Input Buffer Size	1
SE Forwarding Mode	Cut-through
Speed Advantage	1
IC Buffer Size	32 cells

Table 1: Parameters of simulated switch

Figure 5(a) shows the minimum, maximum, and mean crossing delay for offered loads in the interval $[0.4, 0.5]$, the interval in which congestion first becomes an issue. The minimum delay (Geo4 Min, Geo8 Min) is associated with cells making a single MIN crossing ($d = 1$); the maximum delay (Geo4 Max, Geo8 Max) is associated with cells making the maximum number of MIN crossings ($d = 4$). This demonstrates that the delay associated with $d = 4$ is substantially higher than that of $d = 1$ for both Geo4 and Geo8 traffic. While the maximum crossing delays observed were greater than the mean delay values, they were still less than 12 cell slots even for high offered loads. For low offered loads ($u \leq 0.41$), maximum delay variances of less than 10 cell slots squared were observed for all traffic types. Delay variance for unicast and Geo4 traffic rose rapidly with offered load, showing that it, like delay, is primarily a function of MIN congestion and input queue length. The reader is referred to [13] for further architectural details and results obtained when the proposed speed advantage was applied.

Although using Bernoulli traffic allows the maximum, or best case, network performance to be determined, real network traffic tends to contain non-uniformities which can severely degrade performance. One form these non-uniformities can take are traffic bursts. Traffic burstiness, exhibited by key broadband services such as compressed video, file transfer, and IP traffic, is characterised by relatively short sequences of source activity followed by long idle periods. Most multicast switches perform poorly when subject to such bursty traffic. Because multicast bursts are broken up by successive passes through the input buffers and switching fabric, recycling switches would seem to be more resilient to bursty multicast traffic than either CRBR and CRWR switches. While we confirmed that the mean burst length of recycled traffic was lower than that of input traffic, this was found to have little positive effect on switch performance with maximum throughput falling and delay and delay jitter increasing with increasing burst length. The reasons for this were attributed to:

1. Even though the recycling traffic is less bursty than input traffic, the combination of both arriving at an IC increases delay and delay variance.

2. If a bursty connection is recycled back to the same IC on which it arrived, then if the input burst is long enough the same burst can appear on both of the IC's inputs.

Bursty performance can be improved by ensuring that bursts are not recycled back to the same IC on which they arrived, which requires only a small modification to the algorithm used to construct multicast trees. An alternative suggested by Turner [14] replaces the recycling lines with a distribution network. This serves to spread the effect of recycled bursts over more than one IC.

5.1 Effect of cell recycling on multicast traffic

The complexity and performance of recycling switches is primarily a function of their multiplication factor, M. As M increases, so does the MIN complexity of a switch. At the same time, the maximum multicast tree depth – the maximum number of times a multicast cell must be recycled before exiting the switch – is reduced. The fewer times cells are recycled, the lower delay, jitter, and cell loss probability they experience [15], so increasing M improves multicast QoS and vice versa.

Figure 2 shows an example of a multicast tree and the associated copying and recycling operations that occur in a recycling switch with M=2. Here the multicast VC $\mathcal{D} = \{0, 2, 3, 4, 5, 6, 7\}$ results in cells destined for the output 'deepest' in the multicast tree $\{5\}$ making the greatest number of SN crossings (three), whilst those destined for the 'shallowest' outputs $\{0,3\}$ require just one crossing. This raises the obvious question regarding the relationship between the number of MIN crossings a cell makes before exiting the switch, d, and the cell loss probability (CLP) ϕ, mean cell delay μ, and delay variance σ^2 (jitter) it experiences. Let ϕ_x, μ_x, and σ_x^2 quantify the QoS a cell experiences during the x^{th} MIN crossing, and $\hat{\phi}_x$, $\hat{\mu}_x$, and $\hat{\sigma_x}^2$ the total (accumulated) QoS after the x^{th} crossing. Assuming that successive RN crossings

are independent, then after d RN crossings:

$$\hat{\mu}_d = d \times \mu_1, \tag{2}$$

$$\hat{\sigma_d}^2 = d \times \sigma_1, \tag{3}$$

$$\hat{\phi}_d = 1 - (1 - \phi_1)^d. \tag{4}$$

Figure 6 shows the effect that increasing d has on a cell's QoS in a 32×32 FSBC switch. The consequences of the approximately linear[4] relationship between d and a cell's delay, delay variance, and CLP are:

1) Destinations in the same multicast VC can receive different QoS, which is unfair. As the maximum multicast tree depth d_{max} equals $\lceil \log_M N \rceil$, for switches with large N and small M the difference may exceed an order of magnitude.

2) Switch scalability is reduced. Because $\hat{\phi}_d$, $\hat{\mu}_d$, and $\hat{\sigma}_d^2$ increase linearly with d, and d_{max} increases with the switch size N, then $\hat{\phi}_{d_{max}}$, $\hat{\mu}_{d_{max}}$, and $\hat{\sigma}_{d_{max}}^2$ all increase with switch size.

This lack of fairness and scalability is regarded by others as a major disadvantage of using the recycling method. In the following section, we investigate methods which address these concerns.

6 Improving performance of recycling switches

If we are to improve the scalability and fairness of the recycling method, the following objectives must be achieved:

1) Minimise $\hat{\phi}_{max}$, $\hat{\mu}_{max}$, and $\hat{\sigma}_{max}^2$.

2) Minimise the difference between maximum and minimum QoS within a VC i.e., minimise $\hat{\phi}_{max} - \hat{\phi}_{min}$, $\hat{\mu}_{max} - \hat{\mu}_{min}$, and $\hat{\sigma}_{max}^2 - \hat{\sigma}_{min}^2$.

3) Minimise the difference in worst-case QoS between VCs with different fanouts (e.g., between unicast and multicast connections).

[4]For small ϕ_1, $1 - (1 - \phi_1)^d \approx d \times \phi_1$

The first of these objectives concerns improving the worst-case multicast QoS, with the next two concerned with improving switch fairness within and between VCs. Some methods of achieving these objectives include:

1) Over-engineering the SN (through applying a greater speed advantage, using larger internal buffers, etc.) until the QoS for cells which are recycled the maximum number of times is acceptable.

2) Reducing the maximum depth of the multicast tree d_{max} by increasing M or limiting the maximum VC fanout.

3) Prioritising multicast traffic to reduce its delay and CLP.

The first approach is not scalable, since as d_{max} increases with increasing switch size, the SN complexity required to maintain a given worst-case QoS also increases. If mean VC fanout is low, these excess resources largely go to waste. Measures to reduce d_{max} are also problematic: increasing M increases SN and routing complexity, whereas restricting the maximum multicast fanout limits the types of multicast traffic the switch can carry. The most promising approach involves prioritising switch traffic with multicast cells provided preferential delay and loss treatment.

We now compare the performance of three prioritisation schemes when applied to our 16×16 FSBC switch model: no prioritisation (NP), multicast priority (MP), and crossing priority (CP). Baseline measurements are obtained using NP, which, as the name suggests, does not prioritise cells. Because the hardware complexity of most priority disciplines is proportional to the number of priorities supported [16] we have focused on strategies requiring just two priorities regardless of switch size, MP and CP. These use slightly different strategies to boost the QoS offered recycled traffic, MP by providing absolute priority to multicast traffic over unicast traffic, and CP by assigning cell priority based on MIN crossing number, d, with cells making their first MIN crossing (i.e., with $d = 1$) given low priority, and recycled traffic (i.e., with $d \geq 2$) given high priority.

The behaviour of these priority schemes depends upon the nature of switch input traffic as described by the fanout distribution of switch VCs weighted by their respective loads. A useful characterization of the switch input load is R, the ratio of high priority traffic to total traffic measured at the MIN inputs. The

greater R, the less effective a priority scheme will be at improving the QoS of high priority traffic, and the greater the starvation experienced by low priority traffic. Let u_{input} be the probability that a cell arrives at a particular switch input port during a given time slot, and let f_i be the probability that this cell has fanout i. R_{CP} is simply the recycled load divided by the total load, while R_{MP} is equal to the sum of input multicast load and recycled load divided by the total load, i.e.,

$$R_{CP} = u_{recycled} / \left(u_{input} + u_{recycled} \right), \tag{5}$$

$$R_{MP} = \left(\sum_{i=2}^{N} u_{input} f_i + u_{recycled} \right) / \left(u_{input} + u_{recycled} \right). \tag{6}$$

From (5) and (6), we can make some predictions about how CP and MP should behave. Intuitively MP will improve multicast QoS compared to NP regardless of R_{MP} and switch load; however, for high recycling loads $R_{MP} \rightarrow 1$, so unicast traffic experiences starvation, and multicast performance will not be noticeably improved. CP seeks to improve switch fairness by increasing, relative to NP, ϕ_1, μ_1 and σ_1^2 while reducing ϕ_i, μ_i, and σ_i^2 for $2 \leq i \leq d_{max}$. This should reduce the difference between minimum and maximum QoS values both within and between VCs. CP should also perform better than MP under high recycling loads as $R_{MP} \geq R_{CP}$. ϕ_1, μ_1, and σ_1^2 will be greater under CP than NP, so whether CP represents an improvement over NP for a given d will depend on the amount ϕ_i, μ_i, and σ_i^2 is reduced for $2 \leq i \leq d$. This is determined by the switch architecture, the type of prioritisation used, and the composition of the switch input traffic.

We are more concerned with reducing worst-case delay and delay variance rather than CLP. This decision is justified by considering that if the SN load is less that the maximum SN throughput, we can always reduce cell losses by increasing the IC buffer size. The major limitation to this approach – apart from the (small) increase in IC complexity – is that additional delay and delay jitter is introduced. If multicast delay and delay variance can be managed through MP and CP, then increasing IC buffer size is a more effective method of reducing multicast CLP.

Higher priority cells are provided preferential treatment within the switch whenever resource contention occurs, for example when two or more cells compete for access to a switch link or queue. Our work considers an input-buffered Beneš MIN where contention (hence delay) can occur both within the ICs

Simulation Set	1		2		3	
Priority Scheme	CP	MP	CP	MP	CP	MP
Scenario 1	5.7	9.5	7.3	10.8	11.8	12.2
Scenario 2	20.7	25.7	24.2	26.7	24.8	26.6
Scenario 3	1.4	1.8	2.1	2.2	5.4	5.4
Scenario 4	9.4	9.0	10.7	10.8	14.9	12.4

Table 2: Reduction in Maximum Delay (%)

and MIN. The IC queueing delay of high-priority traffic can be reduced by using a service discipline other than first-in first-out (FIFO) such as static priority (SP) [17]. MIN delay results from propagation delay and SE output contention, both of which increase with the number of switch stages. Little can be done to reduce MIN propagation delay except by using cut-through switching [18], however SE contention delay can be reduced through preferentially rather than randomly selecting cells for transmission should SE output contention occur [19]. The prioritization mechanisms were chosen to maximise delay differentiation between low- and high-priority traffic, with a SP service discipline applied to the ICs, and absolute priority given to high priority traffic when SE output contention occurs.

The switch input workload was created by multiplexing two groups of independent, identically distributed (i.i.d.) sources, one consisting of purely unicast sources and the other purely multicast sources. Multicast fanout was based upon a modified truncated geometric distribution where an arriving multicast cell has a fanout i with probability:

$$\Pr[f = i] = \left(\frac{1-\gamma}{\gamma - \gamma^N}\right)\gamma^{i-1}, \qquad 2 \leq i \leq N \tag{7}$$

By appropriately setting the values of $u_{unicast}$ and u_{mcast}, it is possible to alter both the switch input load u, and ratios R_{CP} and R_{MP}. We considered four input load conditions:

Scenario 1 - light switch load with low multicast load,

Scenario 2 - heavy switch load with low multicast load,

Scenario 3 - light switch load with high multicast load,

Scenario 4 - heavy switch load with high multicast traffic.

These input loads were applied to the following three switch configurations, and the mean delay and delay variance observed for the NP, MP, and CP strategies.

Simulation Set	1		2		3	
Priority Scheme	CP	MP	CP	MP	CP	MP
Scenario 1	25.2	36.5	27.1	40.0	31.7	27.5
Scenario 2	70.0	78.2	71.2	76.2	54.6	57.7
Scenario 3	8.1	9.5	9.5	9.4	18.5	15.0
Scenario 4	31.9	29.0	30.1	31.1	39.1	31.7

Table 3: Reduction in Maximum Variance (%)

Set 1 - 16×16 FSBC switch, Bernoulli input traffic,

Set 2 - 32×32 FSBC switch, Bernoulli input traffic,

Set 3 - 16×16 FSBC switch, bursty input traffic.

Sets 1 and 2 are concerned with examining the behaviour of MP and CP for different switch sizes under Bernoulli input loads. Set 3 examines the performance MP and CP for non-uniform input traffic (interrupted-deterministic process with a mean burst size of 8.0).

Tables 2 and 3 display the maximum delay and delay variance results for CP and MP as a percentage reduction of the maximum NP values. From these results we can conclude that both CP and MP represent an improvement over NP for all of the switch sizes and input loads we trialed. As expected, the greatest improvement is offered when the proportion of multicast traffic is low (less than 10% of total offered load) and the switch load is high (Scenario 2). This is the most common traffic scenario encountered in both LANs and WANs. When offered load was low (Scenario 1 and Scenario 3) the absolute improvement is less dramatic, but could prove useful in sufficiently large switches. The results for Set 2 show that these improvements scale well with increased switch size.

Comparing MP and CP, under uniform traffic MP was the more effective at reducing the worst-case delay and delay variance except in Scenario 4. This must be weighed against the tendency of MP to starve unicast cells of service under high multicast loads. Under bursty loads, the performance of CP relative to MP increased. CP also proved fairer than MP as it reduced the difference between the QoS experienced by unicast and multicast cells, and within multicast connections.

The results, which are in accordance with our predictions, demonstrate that worst-case multicast delay and delay variance can be significantly improved by applying MP and CP. This improvement is not appreciable under low switch loads where fixed components tend to dominate, however for such low loads,

delay and delay variance are too low to pose much of a problem regardless of the value of d. The greatest improvement was observed when the switch was congested and the proportion of multicast traffic was low. The reader is referred to [13] for further results and discussion of how prioritisation effects other aspects of switch design, particularly the positioning/dimensioning of output and resequencing buffers.

7 Conclusion

In this paper we have established that a 16×16 FSBC switch with a judiciously applied speed advantage can provide 100% throughput with low delay and jitter. For larger recycling switches where the delay and jitter of recycled cells can become an issue, simple prioritisation techniques can significantly improve worst-case multicast QoS and switch fairness with little increase in switch complexity. Of the two priority schemes introduced, CP and MP, MP provides the superior worst-case multicast performance except where multicast and input loads are simultaneously high. On the other hand, CP represents a good compromise between improving worst-case multicast performance and improving fairness.

References

[1] R. Cusani and F. Sestini, "A recursive multistage structure for multicast ATM," *Proc. IEEE INFOCOM '91*, vol. 3, pp. 1289–1295, Apr. 1991.

[2] Y. Xiong and L. Mason, "Multicast ATM switches using buffered MIN structure : a performance study," *Proc. IEEE INFOCOM '97*, vol. 3, pp. 924–931, Apr. 1997.

[3] T. Chaney, J. Fingerhut, M. Flucke, and J. Turner, "Design of a Gigabit ATM switch," Washington University, St. Louis, Department of Computer Science, Washington University, St. Louis, Technical Report WUCS-96-07, Feb. 1996.

[4] (2004) Cisco Carrier Routing System. Cisco Systems. [Online]. Available: http://www.cisco.com/

[5] M. Yokokawa, "Present status of development of the Earth Simulator," *Proc. IEEE Int. Workshop on Innovative Architectures for Future Generation High-Performance Processors and Systems*, pp. 93–99, Jan. 2001.

[6] J. Turner, "Design of a broadcast packet switching network," *IEEE Trans. Commun.*, vol. 36, no. 6, pp. 734–743, June 1988.

[7] T. T. Lee, "Nonblocking copy networks for multicast packet switching," *IEEE Trans. Commun.*, vol. 6, no. 9, pp. 1455–1467, Dec. 1988.

[8] X. Chen and V. Kumar, "Multicast routing in self-routing multicast networks," *Proc. IEEE INFOCOM '94*, pp. 306–314, Apr. 1994.

[9] J. Park and Y. Yoon, "Cost-effective algorithms for multicast connection in ATM switches based on self-routing multistage networks," *Computer Communications*, vol. 21, pp. 54–64, 1998.

[10] F. Sestini, "Recursive copy generation for multicast ATM switching," *IEEE/ACM Trans. Networking*, vol. 5, no. 3, pp. 329–335, Apr. 1997.

[11] S. Subramaniam and A. Somani, "Multicasting in ATM networks using MINs," *Computer Communications*, vol. 19, pp. 712–722, 1996.

[12] **OPNET Modeler**. OPNET Technologies Inc. [Online]. Available: http://www.opnet.com

[13] D. Hall, "Recycling multicast ATM switches," Master's thesis, School of Electrical Engineering and Computer Science, University of Newcastle, 2006.

[14] J. Turner, "An optimal nonblocking multicast virtual circuit switch," *Proc. IEEE INFOCOM '94*, vol. 1, pp. 298–305, June 1994.

[15] S. Shimamoto, W. Zhong, Y. Onozato, and J. Kaniyil, "Recursive copy networks for large multicast ATM switches," *IEICE Trans. Commun.*, vol. E75-B, no. 11, pp. 1208–1219, Nov. 1992.

[16] S.-W. Moon, J. Rexford, and K. Shin, "Scalable hardware priority queue architectures for high-speed packet switches," *IEEE Trans. Comput.*, vol. 49, no. 11, pp. 1215–1227, Nov. 2000.

[17] L. Kleinrock, *Queueing Systems Volume II: Computer Applications.* John Wiley & Sons, 1976.

[18] I. Widjaja, A. Leon-Garcia, and H. Mouftah, "The effect of cut-through switching on the performance of buffered Banyan networks," *Computer Networks and ISDN Systems*, vol. 26, pp. 139–159, 1993.

[19] Y. Zhaohui and J. Yih-Chyun, "Performance analysis of a Banyan based ATM switching fabric with packet priority," *Proc. IEEE Conference on Local Computer Networks*, pp. 78–89, Oct. 1996.

Figure captions

Fig. 1 Three MIN-based multicast architectures a) Cell replication before routing b) Cell replication while routing c) Cell recycling

Fig. 2 A multicast tree a) and its associated routing b) in a 8×8 switch with $M = 2$

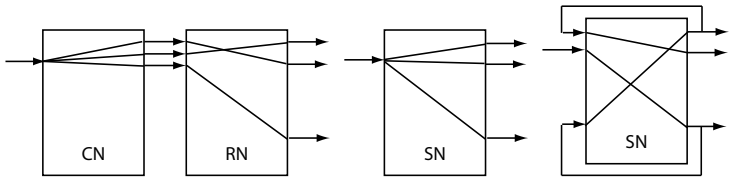
Fig. 3 Cusani and Sestini's design for a binary-copying recycling switch

Fig. 4 Throughput and delay vs. offered load for a 16×16 FSBC switch

Fig. 5 Throughput, delay, and delay variance results for a 16×16 FSBC switch for offered loads in the range [0.4,0.5]

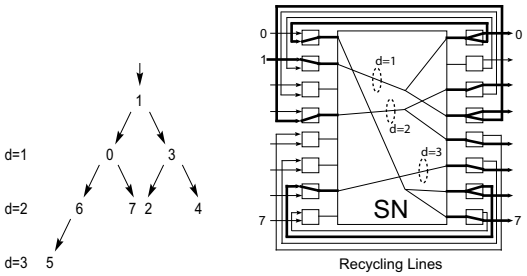
Fig. 6 Per-MIN crossing delay, delay variance, and CLP in a 32×32 FSBC switch. Clearly shown is the linear relationship between d and QoS

Figures



(a) (b) (c)

Figure 1:



(a) (b)

Figure 2:

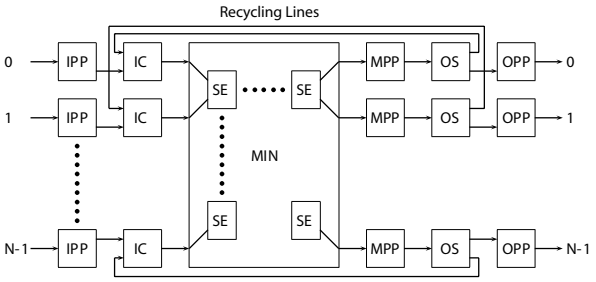
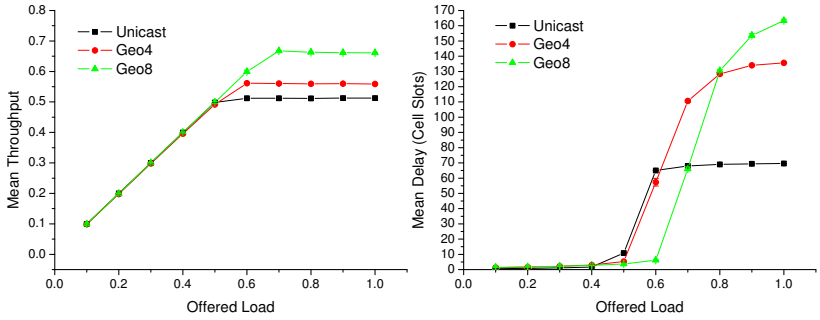
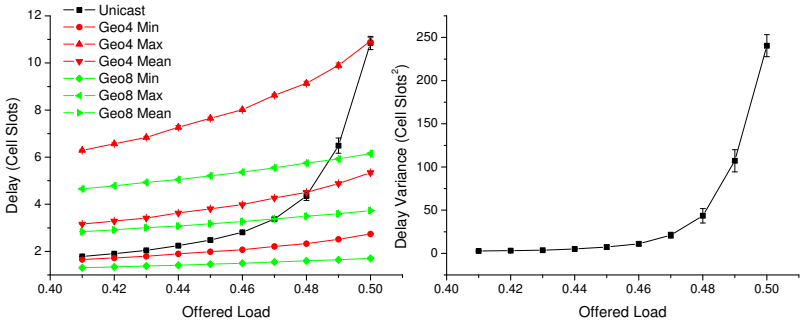


Figure 3:

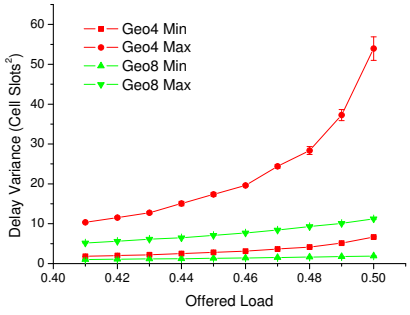


(a) Throughput (b) Delay

Figure 4:

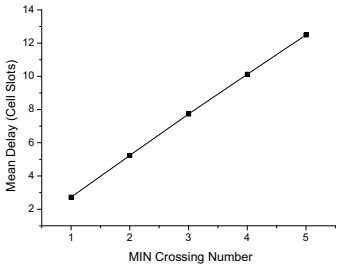


(a) Delay (b) Unicast Delay Variance

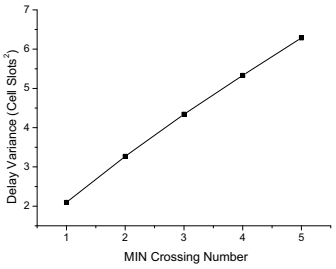


(c) Multicast Delay Variance

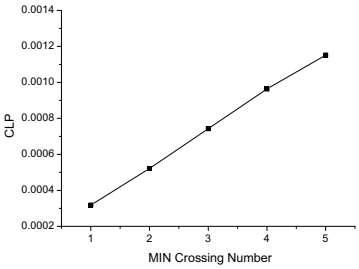
Figure 5:



(a) Mean Delay



(b) Delay Variance



(c) CLP

Figure 6: