1998 International Computer Symposium
Workshop on Computer Networks, Internet, and Multimedia
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

EFFICIENT NONBLOCKING MULTICAST COMMUNICATIONS ON BASELINE NETWORKS

W. Lin and E. W. Chen

Institute of Computer Science
National Chung-hsing University
250 Kuo-kuan Road
Taichung, Taiwan ROC
email: echen@cs.nchu.edu.tw

ABSTRACT

This paper explores the problem of efficiently multicasting packets on the baseline network in accordance with a given set of multicast communications. A baseline network is a multistage interconnection network (MIN) with N inputs and N outputs as well as $\log_2 N$ stages of 2x2 switches. The baseline network construction with wraparound connections leads us to propose a new scheme for nonblocking packet multicasts. Previous approaches use a cascade of various MINs to multicast packets in a finite number of steps. Some others use single MINs to recycle and copy packets repeatedly until the multicast is done. Our proposed scheme exhibits a different approach from the previous ones, yet possessing many desirable features of theirs. Our scheme employs a $\log_2 N$-stage baseline network with wraparound connections. It is capable of accomplishing any multiple multicast in four finite passes. The first two passes replicate individual packets simultaneously. Then the subsequent two passes route these packets to the destinations. We demonstrate that paths created in the four passes are all link-disjoint so that nonblocking multicast is achieved.

Keywords: multicast, ATM switches, multistage networks, nonblocking networks

I. INTRODUCTION

As the demand for point-to-multipoint services grows, it becomes increasingly important to incorporate multicast capability in the design of ATM broadband switches. Multicast services deliver messages from a source to an arbitrary number of destinations, providing a variety of practical applications in B-ISDNs, such as teleconferencing and video-on-demand. Many ATM switch architectures ever proposed or built are based on multistage interconnection networks (MIN) for switching packets(cells). A lot of research efforts have been devoted to investigating multicast capability of multistage interconnection networks. Various multicast MINs are proposed. These multicast MINs fall into two broad categories. The first category typically employs a cascade of various MINs. These MINs are dedicated for replicating and routing multicast packets [1-4]. The second category, called recursive scheme, uses a multistage interconnection network with wraparound connections [5-7]. For multicasting a packet, the network each time generates some copies and recycles some of them to produce other copies, using wraparound connections.

Multicast can easily cause network congestion, due to simultaneous presence and multiplication of packet copies from various sources. Multistage interconnection networks offer an attractive solution to the congestion problem. They are effective in controlling packet collisions. At the heart of each proposed multicast MIN is a routing scheme for reducing or eliminating packet collisions. Bubenik and Turner take a randomization approach to the network congestion problem [8]. In their proposed switch, a distribution network is used to randomize packets routing across the switch. Lee employs a banyan network followed by a reverse banyan network as a copy network for replicating packets [2], taking advantage of the banyan network's nonblocking capability in packets concentration. Chen and Kumar use the cube concept to decompose a multicast connection into many intermediate multicast cubes and to form a multicast tree. Connections created within the cubes are disjoint [5]. Sestini proposes a recursive copy scheme for multicast. The scheme recycles and multiplies packet copies by conflict-free connections, while the number of passes is minimized [7]. There are many other research efforts relevant to this subject matter [9-12].

In this paper we present an efficient method for multicasting packets on the baseline network [13]. We

exploit the intrinsic nonblocking property of the baseline network to perform multiple packet multicasts. By multiple packet multicasts we mean that the network carries out several packet multicasts simultaneously. We prove that multiple packet multicasts can be accomplished in four passes on the baseline network with wraparound connections. Our work differs from previous research efforts in several respects. The most important is that we employ a single baseline network to achieve multicast, rather than cascading various multistage interconnection networks for copying and routing packets. Our proposed method is cost-effective in the sense of hardware requirement. Existing copy networks are typically composed of separate segments with such capabilities as concentrating and expanding packets. We show that although the baseline network lacks these capabilities, the combined effect of circulating packets twice through the network can also achieve packet replication. Furthermore, we show that two additional passes through the baseline network can deliver multicast packets to their respective destinations, without using a dedicated routing network. Our work differs from those recursive schemes in that the proposed method provides a deterministic solution to the multicast problem, rather than a probabilistic solution. It guarantees that in four passes all the multicast packets can reach their final destinations.

The remaining discussion is organized as follows. The next section describes the basic structure and routing capability of the baseline network. Section 3 presents a reordering procedure used in the first pass. Section 4 deals with the second pass for replicating multicast packets. Section 5 depicts a routing method employed in the last two passes for moving packets toward destinations. The first few sections basically focus on detailing mechanics of the four passes and proving their correctness analytically. Isolated examples are given to explain essential operations of the four passes. To avoid their interruption and better illustrate the chaining relation among these four passes, in Section 6 we further present a running example with a series of four instances. Each of them uses the output of the preceding instance as the input. Finally, conclusions are drawn in Section 7.

II. BASICS AND NETWORK STRUCTURE

In this section, we formally define three permutations and two bit-manipulation functions. Their notations are presented. We precisely describe the structure of baseline networks. We then prove some basic properties of link-disjoint paths that can be realized on baseline networks. The definitions in this section are somewhat intricate and tedious. However, our solution to achieving efficient multicast on baseline networks turns out to be quite straightforward. In the subsequent sections, we use these definitions to show various routing capabilities of the baseline network required for multicast. We first introduce a notation for the bit reversal permutation.

Definition 1 The bit reversal permutation ρ is defined to be

$$\rho\ (x_{n-1}x_{n-2}...x_0) = x_0x_1...x_{n-1}$$

where $x_{n-1}x_{n-2}...x_0$ is the n-bit representation of a number x in $\{i|\ 0 \le i \le 2^n-1\}$.

Note that the real value of $\rho(x)$ depends on bit length used for representing the number x. For instance, when n=4, $\rho(3)$ equals $12(=1100_2)$; however, when n=5, $\rho(3)$ equals $24(=11000_2)$. Next, we introduce two permutations for describing the structure of the baseline network.

Definition 2 The k-bit shuffle permutation σ_k is defined to be

$$\sigma_k(x_{n-1}x_{n-2}...x_0) = x_{n-1}...x_kx_{k-2}...x_0x_{k-1}$$

where k≤n, and $x_{n-1}x_{n-2}...x_0$ is the n-bit representation of a number x in $\{i|\ 0 \le i \le 2^n-1\}$.

Simply speaking, the effect of $\sigma_k(x)$ is a circular left shift of the k low-order bits of the n-bit representation of x. The k-bit shuffle permutation σ_k actually is a generalization of the well-known perfect shuffle; that is, in the case of k=n, $\sigma_n(x_{n-1}x_{n-2}...x_0) = x_{n-2}...x_0x_{n-1}$. Next, we define the inverse permutation for the k-bit shuffle permutation σ_k.

Definition 3 The k-bit unshuffle permutation σ_k^{-1} is defined to be

$$\sigma_k^{-1}(x_{n-1}x_{n-2}...x_0) = x_{n-1}...x_kx_0\ x_{k-1}...x_1$$

where k≤n, and $x_{n-1}x_{n-2}...x_0$ is the n-bit representation of a number x in $\{i|\ 0 \le i \le 2^n-1\}$.

The following definition describes a bit-manipulating function that concatenates two substrings of two n-bit representations to form a new n-bit representation.

Definition 4 Let $x_{n-1}x_{n-2}...x_0$ and $y_{n-1}y_{n-2}...y_0$ be the n-bit representations of two numbers x and y in $\{i|\ 0 \le i \le 2^n-1\}$. Function θ_j is defined to be

$\theta_j(x,y) = x_{n-1}...x_{n-j}y_{n-1}...y_j$,
where $0 \le j \le n-1$.

Note that in the special case of j=0, $\theta_j(x,y) = y_{n-1}y_{n-2}...y_0$. For example, suppose $x=10110_2$ and $y=01101_2$. Then $\theta_3(x,y)=10101_2$.

Definition 5 Let $x_{n-1}x_{n-2}...x_0$ and $y_{n-1}y_{n-2}...y_0$ be the n-bit representations of two numbers x and y in $\{i \mid 0 \le i \le 2^n-1\}$. Function $\pi(x,y)$ yields the longest prefix shared by $x_{n-1}x_{n-2}...x_0$ and $y_{n-1}y_{n-2}...y_0$.

For convenience, we denote the length of $\pi(x,y)$ as $|\pi(x,y)|$. For example, suppose $x=11001_2$ and $y=11101_2$. Then, $\pi(x,y)=11_2$ and $|\pi(x,y)|=2$.

We now turn our attention to the structure of the baseline network. A $2^n x 2^n$ baseline network is used to interconnect a set of 2^n input terminals and 2^n output terminals. An *8x8* baseline network with wraparound connections is illustrated in Figure 1(a). A $2^n x 2^n$ baseline network consists of *n* stages of *2x2* multicast switches. Each switch has two inputs and two outputs. It is capable of switching and broadcasting packets.

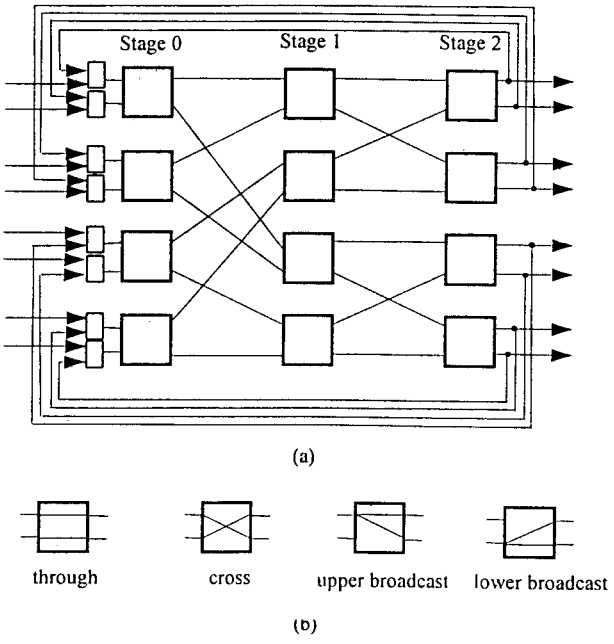

Figure 1. Structure of an *8x8* baseline network with wraparound connections and switch states.

Figure 1(b) shows some of the switching capabilities of the *2x2* switch. Each stage contains 2^{n-1} switches. Components of a $2^n x 2^n$ baseline network are labeled in the following manner. The input terminals, also the output terminals, are numbered in a sequence from *0* to 2^n-1, top to bottom. They are called *terminal addresses*. The stages are numbered from 0 to n-1, left to right.
The switches in a stage are numbered from 0 to $2^{n-1}-1$, top to bottom. Switches of two adjacent stages are linked by the k-bit unshuffle permutation depicted in Definition 2. More specifically, the links between stages i and i+1, for $0 \le i \le n-2$, are arranged in the pattern of σ^{-1}_{n-i}. For instance, in Figure 1(a) we see that the links between stage 0 and 1 are arranged according to the 3-bit unshuffle permutation, and the links between stages 1 and 2 are arranged according to the 2-bit unshuffle permutation.

The baseline network is a multistage interconnection network in which there is a unique path between each input terminal and output terminal. Switches and links traversed by a path can be specified with a combination of binary addresses of input terminal and output terminal. When referring to input terminals and output terminals, we often use their n-bit addresses for representations. This automatically implies that the baseline network is of size $2^n x 2^n$. Now, let $S = s_{n-1}s_{n-2}...s_0$ be an input terminal and $D = d_{n-1}d_{n-2}...d_0$ be an output terminal. The first switch traversed by the path from S to D is switch $s_{n-1}s_{n-2}...s_1$ at stage 0, and the second switch is switch $d_{n-1}s_{n-1}...s_2$ at stage 1, and the third switch is switch $d_{n-1}d_{n-2}s_{n-1}...s_3$ at stage 2, etc. In general, the switch at stage i, $0 \le i \le n-1$, traversed by the path from S to D is

$$d_{n-1}...d_{n-i}s_{n-1}...s_{i+1}.$$

Link traversal is associated with switch traversal. In the i^{th} stage, the traversed switch selects an output port to route the path, based on the value of the $(n\text{-}i\text{-}1)^{th}$ bit of $d_{n-1}d_{n-2}...d_0$. If the bit is a 0 the upper output port is selected, and if the bit is a 1 the lower output port is selected. If only one path requests for a particular output port, it moves to the associated link. When two paths arrive at the same switch and request for the same output port, a link contention occurs. On the other hand, two paths are link-disjoint if they always traverse different switches throughout the network, or if they arrive at the same switch at some stage and request for different output ports. The following theorem establishes the condition where link-disjoint paths can be accomplished. Central to the theorem is a simple test on the n-bit terminal addresses of two paths. The test tells whether two paths come across link contention or they are link-disjoint.

Theorem 1 Suppose a ≠ c and b ≠ d. Let a→b and c→d be two paths through a $2^n x 2^n$ baseline network. The two paths are link-disjoint if and only if

$$|\pi(a,c)| + |\pi(b,d)| < n.$$

Proof. Suppose $|\pi(a,c)| + |\pi(b,d)| < n$ and suppose that the two paths, a→b and c→d, contend a link at some stage j, $0 \le j \le n-1$. The necessary and sufficient condition for the two paths to contend the same link at stage j is:

(1) they arrive at the same switch of stage j; and
(2) they compete the same output port of the switch.

(1) indicates that $\theta_j(b,a)$ and $\theta_j(d,c)$ must have the most significant $n-1$ bits in common. That is,

$$|\pi(\theta_j(b,a), \theta_j(d,c))| \ge n-1.$$

Therefore, we have $a_{n-1}...a_{j+1} = c_{n-1}...c_{j+1}$ if $j < n-1$, and $b_{n-1}...b_{n-j} = d_{n-1}...d_{n-j}$ if $j > 0$. (2) indicates that the two routing bits, b_{n-j-1} and d_{n-j-1}, for a→b and c→d at stage j should be identical. Combining (1) and (2), we have

$$|\pi(a,c)| + |\pi(b,d)| \ge n.$$

This contradicts $|\pi(a,c)| + |\pi(b,d)| < n$. Therefore, paths a→b and c→d must be link-disjoint.

Conversely, if paths a→b and c→d are link-disjoint, either they traverse different switches or they reach the same switch but different output ports at every stage j, $0 \le j \le n-1$. The former implies $|\pi(\theta_j(b,a), \theta_j(d,c))| < n-1$. The latter means $|\pi(\theta_j(b,a), \theta_j(d,c))| = n-1$ and $b_{n-j-1} \ne d_{n-j-1}$. We conclude that

$$|\pi(\theta_j(b,a), \theta_j(d,c))| \le n-1,$$

for $0 \le j \le n-1$. Consequently,

$$|\pi(a,c)| + |\pi(b,d)| < n. \quad \blacksquare$$

For example, in an *8x8* baseline network, paths 4→2 and 6→3 incur link contention. This is indicated by the test of Theorem 1: $|\pi(4,6)| + |\pi(2,3)| = 1+2=3$. The following theorem is an immediate result mainly due to Theorem 1. It shows that if we interchange the output terminals of two link-disjoint paths, the resulting paths remain link-disjoint. Later on this interesting result will help us derive link-disjoint paths for multicast.

Theorem 2 If a→b and c→d are two link-disjoint paths through a $2^n x 2^n$ baseline network, then a→d and c→b are link-disjoint.

Proof. Since a→b and c→d are link-disjoint, we should have

$$|\pi(a,c)| + |\pi(b,d)| < n.$$

Now let us test paths a→d and c→b with $|\pi(a,c)| + |\pi(d,b)|$, we obtain

$$|\pi(a,c)| + |\pi(d,b)| = |\pi(a,c)| + |\pi(b,d)| < n.$$

As a consequence, paths a→d and c→b are also link-disjoint. ■

For example, in an *8x8* baseline network, paths 6→2 and 1→3 are link-disjoint. Then paths 6→3 and 1→2 are link-disjoint.

III. PACKETS REORDERING

The first two passes are essentially responsible for packet replications. A fundamental requirement is that the two passes have to copy packets through link-disjoint paths. The main function of the first pass is to rearrange multicast packets in a certain order to be described at once. The order is chosen for two reasons. First, it ensures that paths created for reordering the packets are link-disjoint themselves. Secondly, as we take the packets in the rearranged order and replicate them in the following pass, paths created for packets replication are also link-disjoint. In essence, the first pass is required to eliminate completely link contentions that may occur in replicating packets. Now, let us look at an important property for avoiding link contention. This property eventually leads us to the proposed reordering procedure applied in the first pass. From now on, unless otherwise specified, we treat terminal addresses as unsigned numbers.

Theorem 3 Let a and b be two inputs of a $2^n x 2^n$ baseline network. Suppose that c and d are two positive integers, and $0 \le c, d \le 2^n-1$. If $0 < d - c \le b - a$, then paths a→ $\rho(c)$, b→ $\rho(d)$ are link-disjoint.

Proof. First consider the case: a ≥ c. Let $c = c_{n-1}c_{n-2}...c_0$ and $d = d_{n-1}d_{n-2}...d_0$. Suppose that k is the lowest bit position where c and d differ; that is, we have $c_{k-1}...c_0 = d_{k-1}...d_0$ and $c_k \ne d_k$. Therefore, we have $|\pi(\rho(c), \rho(d))| = k$.

Let Δ = a − c. Since c+Δ and d+Δ are still different in bit position k, we have

$$|\pi(\rho(c), \rho(d))| = |\pi(\rho(c+\Delta), \rho(d+\Delta))| = |\pi(\rho(a), \rho(d+\Delta))| = k.$$

This means $|\pi(a, d+\Delta)| < n - k$. Let h = d + Δ and $|\pi(a, h)| = p$. Suppose that $a_{n-1}a_{n-2}...a_0$ and $h_{n-1}h_{n-2}...h_0$ are the n-bit representations of a and h, respectively. Then we must have $a_{n-p-1} = 0$ and $h_{n-p-1} = 1$, because a < h. Now

we have $h \le b$, because $b - a \ge d - c$. Combining these inequalities, we obtain

$$a < h \le b.$$

Suppose $|\pi(a, b)| = q$. Apparently, q must be smaller than or equal to p; otherwise, we would have $b < h$. This gives rise to

$$\begin{aligned} |\pi(a, b)| + |\pi(\rho(c), \rho(d))| &= q + k \\ &\le p + k \\ &< (n-k) + k \\ &< n. \end{aligned}$$

We conclude that paths $a \to \rho(c)$, $b \to \rho(d)$ are link-disjoint.

In the case of $a < c$, paths $a \to \rho(c)$, $b \to \rho(d)$ are also link-disjoint. Because the proof resembles that of the first case in many ways, it is omitted here. ■

It is worthy of mentioning that the bit reversal permutation is the inverse function of itself; i.e., for an n-bit number x, $0 \le x \le 2^n-1$, $\rho(\rho(x)) = x$. Making use of this, we restate Theorem 3 as follows:

Alternative of Theorem 3 Let a and b be two inputs of a $2^n x 2^n$ baseline network. Suppose that c and d are two network outputs, and $0 < \rho(d) - \rho(c) \le b - a$. Then paths $a \to c$ and $b \to d$ are link-disjoint.

Let us look at an example with two paths, $3 \to 1$, and $6 \to 5$, on an 8x8 baseline network. Applying the bit reversal permutation to 1 and 5 in 3-bit representation, we have $\rho(5) = 5$ and $\rho(1) = 4$. By Theorem 3, we assert that the two paths are link-disjoint. Routing of the two paths, $3 \to 1$, and $6 \to 5$, is illustrated in Figure 2. With this in mind, we now present the reordering procedure for routing paths in the first pass.

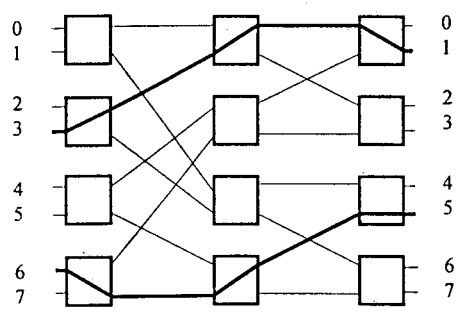

Figure 2. Example of two link-disjoint paths.

Assume that there are k active inputs selected for multicast and each of them will send a multicast packet to a distinct set of destinations. The following procedure is used to create paths for reordering multicast packets. While reordering multicast packets, we merely consider the relative order of active inputs. And destinations of multicast packets are not considered at this moment. The following procedure specifies paths required to accomplish such reordering.

Reordering Procedure

Step 1: Arrange the k active inputs in ascending order and label them as x_0, $x_1,\ldots,x_{k-1}$, starting with subscript 0;

Step 2: Create paths $x_0 \to \rho(0)$, $x_1 \to \rho(1)$,..., and $x_{k-1} \to \rho(k-1)$.

For example, suppose that in a *16x16* baseline, active inputs 2, 5, 11, and 13 are selected for multicast. Using the reordering procedure, we create four paths in the first pass: they are $2 \to 0$, $5 \to 8$, $11 \to 4$, and $13 \to 12$. Next, we show that paths created in the first pass for reordering multicast packets are link-disjoint.

Theorem 4 In a $2^n x 2^n$ baseline network, if $\{x_0, x_1,\ldots,x_{k-1}\}$ is a set of k active inputs arranged in ascending order, and if $\rho(0), \rho(1),\ldots, \rho(k-1)$ are the corresponding outputs, then paths $x_0 \to \rho(0)$, $x_1 \to \rho(1)$,..., and $x_{k-1} \to \rho(k-1)$ are link-disjoint.

Proof. Suppose $x_a, x_b \in \{x_0, x_1, \ldots, x_{k-1}\}$ and $x_a < x_b$. Active inputs $x_0, x_1, \ldots, x_{k-1}$ are arranged in ascending order and they are enumerated from 0 to *k-1*. Therefore, we have $x_b - x_a \ge b - a$. By Theorem 3, we assert that $x_a \to \rho(a)$ and $x_b \to \rho(b)$ are link-disjoint. In other words, all the paths $x_0 \to \rho(0)$, $x_1 \to \rho(1)$,..., and $x_{k-1} \to \rho(k-1)$ are link-disjoint. ■

IV. PACKET REPLICATION

This section describes how multicast packets are duplicated in the second pass. We essentially create a forest of link-disjoint paths for replicating multicast packets. A forest encompasses several duplicating trees, and each of them is responsible for duplicating a particular packet. A duplicating tree consists of a set of paths which originate from the same input and split at some switches of a baseline network. Through the link-disjoint paths, these trees simultaneously replicate packets and make the requested numbers of copies. Now, let us look at a basic property of the paths that constitute the duplicating trees.

Theorem 5 Suppose that a and b are two positive integers in $\{ i \mid 0 \le i \le 2^n-1 \}$. And suppose that c and d are two outputs of a $2^n x 2^n$ baseline network. If $0 < b - a \le d - c$, then paths $\rho(a) \to c$ and $\rho(b) \to d$ are link-disjoint through the network.

Proof. We will prove that $|\pi(\rho(a), \rho(b))| + |\pi(c, d)| < n$. First consider the case: $a \geq c$.

Let $a = a_{n-1}a_{n-2}...a_0$ and $b = b_{n-1}b_{n-2}...b_0$. Suppose that k is the lowest bit position where a and b differ; that is, $|\pi(\rho(a), \rho(b))| = k$. More explicitly, we have $a_{k-1}...a_0 = b_{k-1}...b_0$ and $a_k \neq b_k$. Now let $\Delta = a - c$; or $c = a - \Delta$. Since $a-\Delta$ and $b-\Delta$ are still different in bit position k, we have

$$\begin{aligned} |\pi(\rho(c), \rho(b-\Delta))| &= |\pi(\rho(a-\Delta), \rho(b-\Delta))| \\ &= |\pi(\rho(a), \rho(b))| \\ &= k. \end{aligned}$$

It is immediate to have the inequality:

$$|\pi(c, b-\Delta)| < n - k.$$

Let $g = b - \Delta$ and $p = |\pi(c, g)|$. Thus we have $p < n - k$. Let $c_{n-1}c_{n-2}...c_0$ and $g_{n-1}g_{n-2}...g_0$ be the n-bit representations of c and g, respectively. Since $c = a - \Delta$ and $g = b - \Delta$, we know $c < g$. Note that $c_{n-1}...c_{n-p} = g_{n-1}...g_{n-p}$; however, c_{n-p-1} and g_{n-p-1} must be different, because $p = |\pi(c, g)|$. As a consequence, we have $c_{n-p-1} = 0$ and $g_{n-p-1} = 1$. (Otherwise, we would have $c_{n-p-1} = 1$ and $g_{n-p-1} = 0$; and this leads to a contradicting result: $c > g$.) Now because $b - a \leq d - c$, we have $g \leq d$. Combining these inequalities, we obtain

$$c < g \leq d.$$

Suppose $|\pi(c, d)| = q$. And q must be smaller than or equal to p. If not, i.e. $q > p$, we have $g_{n-1}...g_{n-p} = c_{n-1}...c_{n-p} = d_{n-1}...d_{n-p}$, $g_{n-p-1} = 1$, and $d_{n-p-1} = c_{n-p-1} = 0$ as well. This results in a contradiction, $g > d$. Now turn our attention to the following:

$$\begin{aligned} |\pi(\rho(a), \rho(b))| + |\pi(c, d)| &= k + q \\ &\leq k + p \\ &< k + (n-k) \\ &< n. \end{aligned}$$

We conclude that paths $\rho(a) \rightarrow c$, $\rho(b) \rightarrow d$ are link-disjoint.

We can likewise prove that $|\pi(\rho(a), \rho(b))| + |\pi(c, d)| < n$ if $c > a$. Therefore, we conclude that $\rho(a) \rightarrow c$ and $\rho(b) \rightarrow d$ are link-disjoint. ■

We give an example to illustrate paths that possess this property. In an *8x8* baseline network, let us create two paths: $2 \rightarrow 3$ and $6 \rightarrow 5$. Taking the bit reversal function on 2 and 6 in 3-bit representation, we have $\rho(2)=2$ and $\rho(6)=3$. We find that $\rho(6)-\rho(2) < 5-3$. Therefore, by Theorem 5, we assert that the two paths, $2 \rightarrow 3$ and $6 \rightarrow 5$, are link-disjoint. Using paths of this kind, we can create trees for replicating multicast packets through a baseline network. Each tree comprises a number of branches and they correspond to the number of packet copies intended to make. For convenience, we use $x \triangleleft Y$ to denote a duplicating tree which originates at an input x and terminates at an output set Y. Note that the set Y should contain an equivalent number of outputs to the replication count of the multicast packet from x. The following method creates simultaneous duplicating trees in a baseline network. Here we suppose a set of k multicast packets, and that they have been reordered in the first pass and they are currently held at inputs $\rho(0), \rho(1),..., \rho(k-1)$.

The Copy Method

1. Starting from output 0, partition the consecutive outputs of a baseline network into k sets, denoted as $Y_0, Y_1,...Y_{k-1}$, such that the number of outputs in Y_i is equal to the number of copies to be made for the packet at input $\rho(i)$.
2. Rooting at inputs $\rho(0), \rho(1),..., \rho(k-1)$, create k simultaneous trees with the k sets of outputs, $Y_0, Y_1,...Y_{k-1}$, respectively.

For example, suppose that three packets are intended for multicast on an *8x8* baseline network. Reordered by the first pass, the three packets currently stay at inputs $\rho(0)$, $\rho(1)$ and $\rho(2)$, and request for three, two and two copies, respectively. Using the copy method, we group the first seven outputs of the *8x8* baseline network into three sets; namely, $Y_0=\{0,1,2\}$, $Y_1=\{3,4\}$, and $Y_2=\{5,6\}$. Then paths of the three trees, $\rho(0) \triangleleft Y_0$, $\rho(1) \triangleleft Y_1$, and $\rho(2) \triangleleft Y_2$, are established for duplicating the three multicast packets.

We now need to show that duplicating trees created by the copy method comprise link-disjoint paths only. This is done in the proof of the following theorem.

Theorem 6 Suppose that $Y_0, Y_1,...Y_{k-1}$ are k nonempty sets of consecutive outputs of a $2^n x 2^n$ baseline network, and suppose that for every $u \in Y_i$ and $v \in Y_j$, where $0 \leq i < j \leq k-1$, u is smaller than v. Then the paths of duplicating trees $\rho(0) \triangleleft Y_0$, $\rho(1) \triangleleft Y_1$, ..., and $\rho(k-1) \triangleleft Y_{k-1}$ are mutually link-disjoint.

Proof. Before starting with the proof, we should make clear what is meant by the last statement of the theorem. More explicitly, it means that if $u \in Y_i$ and $v \in Y_j$, where $0 \leq i < j \leq k-1$, then $\rho(i) \rightarrow u$ and $\rho(j) \rightarrow v$ are link-disjoint.

Since $Y_0, Y_1,...Y_{k-1}$ are k nonempty sets of outputs, we have $j - i \leq v - u$. By Theorem 4, we know that $\rho(i) \rightarrow u$ and $\rho(j) \rightarrow v$ are link-disjoint. Thus we

conclude that duplicating trees $\rho(0) \lhd Y_0$, $\rho(1) \lhd Y_1$, ..., $\rho(k-1) \lhd Y_{k-1}$ are mutually link-disjoint. ■

V. PACKETS ROUTING

In this section, we show how to route multicast packets, after being duplicated, to their final destinations. We intend to show that arbitrary permutations of packets can be realized by passing these packets through the baseline network twice. Our idea behind this is based on a fact that there exists a class of multistage networks, called Benes networks, and they can realize arbitrary permutations [14]. We prove that the combined functionality of two passes through a baseline network can emulate a Benes network, provided switches of the baseline network are properly set in each pass.

Now, let us look at the structure of Benes networks. Shown in Figure 3 is an *8x8* Benes network. A $2^n x 2^n$ Benes network consists of *2n-1* stages of *2x2* crossbar switches and each stage uniformly contains 2^{n-1} such switches. Switch interconnection between adjacent stages proceeds as follows. Interconnection patterns of the first n stages of a $2^n x 2^n$ Benes network are, stage by stage, identical to those of a $2^n x 2^n$ baseline network.

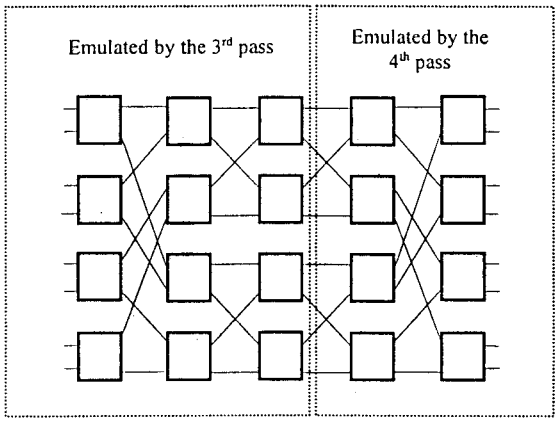

Figure 3. Example of an 8x8 Benes network.

They are unshuffle-based permutations. Interconnection patterns of the remaining *n-1* stages are the mirrored patterns of the first n stages. At this point, it is apparent to see that a $2^n x 2^n$ baseline network can emulate the routing capability of the first n stages of a $2^n x 2^n$ Benes network. Therefore, it requires one pass of the baseline network for such emulation. Paths routing in this pass can be done by using a well-known procedure, called the looping procedure by Opferman and Tsao-wu [14]. This looping procedure is originally used to route paths stage-by-stage for arbitrary permutations through a Benes network. Because of the aforementioned topological relationship, we can directly use the result produced by the looping procedure to route paths through the n stages of a $2^n x 2^n$ baseline network. However, we need to point out immediately that the emulation of the Benes network requires an additional pass to complete the routing of packets multicast. This is explained as follows.

Now we consider emulation of the last *n-1* stages of a $2^n x 2^n$ Benes network using an *n*-stage baseline network. In the subsequent discussion, for convenience we refer the last n-1 stages of a $2^n x 2^n$ Benes network to as partial Benes network, and PBN for short. Stage *n* of the Benes network becomes stage 1 of the corresponding PBN, and stage n+1 of the Benes network becomes stage 2 of the corresponding PBN, and so on. Here we indicate two major differences between the baseline network and the PBN. First, their network structures are completely different. Hardly can we see any close relation between the baseline network and the PBN. Secondly, unlike the baseline network, the PBN does not provide the full accessibility of outputs. Each input can be connected only to some outputs. In a $2^n x 2^n$ PBN, a permissible path a→b, where a = $a_{n-1}a_{n-2}...a_0$ and b = $b_{n-1}b_{n-2}...b_0$, has a characteristic that $b_{n-1}=a_0$. Our main concern here is to emulate the PBN with one pass of the baseline network. The two differences we have pointed out, as a matter of fact, have no effect on the emulation capability of the baseline network for the PBN. The following theorem accounts for this. It shows that all the permissible, link-disjoint paths on the PBN can also be realized on the baseline network.

Theorem 7 A $2^n x 2^n$ PBN can be emulated by a $2^n x 2^n$ baseline network.

Proof. Our approach to this proof is to show that any two link-disjoint paths on a $2^n x 2^n$ PBN can also be realized on a $2^n x 2^n$ baseline network. Let a = $a_{n-1}a_{n-2}...a_0$, b = $b_{n-1}b_{n-2}...b_0$, c = $c_{n-1}c_{n-2}...c_0$ and d = $d_{n-1}d_{n-2}...d_0$. Suppose that a→b and c→d are two permissible, link-disjoint paths through a $2^n x 2^n$ PBN. Since they are permissible, we must have $b_{n-1}=a_0$ and $d_{n-1}=c_0$ as well. Now that a→b and c→d are link-disjoint, they traverse different output ports at every stage of the PBN. Namely, we have $\theta_i(a,b) \neq \theta_i(c,d)$, for $1 \le i \le n-2$. Because a and c are two distinct inputs of a $2^n x 2^n$ baseline network, their n-bit representations must be different. That is,

$$a_{n-1}a_{n-2}...a_0 \neq c_{n-1}c_{n-2}...c_0 .$$

Since $b_{n-1}=a_0$ and $d_{n-1}=c_0$, this inequality induces $\theta_{n-1}(a,b) \neq \theta_{n-1}(c,d)$. Combining this and the previous inequality, we assert that $\theta_i(a,b) \neq \theta_i(c,d)$, for $1 \le i \le n-1$, and that

$$|\pi(a,c)| + |\pi(b,d)| < n.$$

Therefore, the two paths, a→b and c→d, are also link-disjoint. ■

VI. A COMPLETE EXAMPLE

This section provides an illustrative example of realizing simultaneous multicasts on a *16x16* baseline network. We intend to use this example to link the operations of the four passes. The running example consists of four successive passes. Snapshots of network mappings are given to illustrate paths routing done in the four passes. Consider the connection set:

$$\begin{pmatrix} 9 & 10 & 7 & 7 & 10 & 2 & 13 & x & 10 & 13 & 2 & 10 & x & 10 & 2 & 7 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \end{pmatrix}$$

Each pair $\begin{pmatrix} i \\ j \end{pmatrix}$ defines a packet delivery from source i to destination j. An x in the first row indicates that no packet is destined to the corresponding output terminal. A packet multicast is indicated by a set of destinations with the same source. This example comprises a unicast from input terminal 9 to output terminal 0, and four simultaneous packet multicasts initiated from input terminals 2, 7, 10, and 13 and destined toward {5,10,14}, {2,3,15}, {1,4,8,11,13}, and {6,9}, respectively.

Pass 1

In this pass, we move the five initial packets to the intermediate network outputs designated by the reordering procedure. Five paths are created to move the packets: 2→0, 7→8, 9→4, 10→12 and 13→2. Mapping of the five paths is shown in Figure 4.

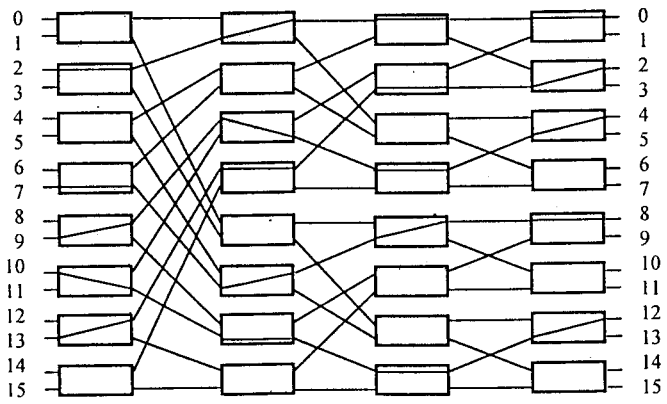

Figure 4. Paths routing in the first pass.

Pass 2

In this pass, the five multicast packets are replicated in the following manner:

(1) packet at 0, which originated from terminal 2, requires three copies;
(2) packet at 2, which originated from terminal 13, requires two copies;
(3) packet at 4, which originated from terminal 9, requires one copy;
(4) packet at 8, which originated from terminal 7, requires three copies; and
(5) packet at 12, which originated from terminal 10, require five copies.

Using the copy method, we create five duplicating trees. Through them, the five packets are copied and spread across the network outputs in the bit reversal order. Mapping of the five duplicating trees is shown in Figure 5.

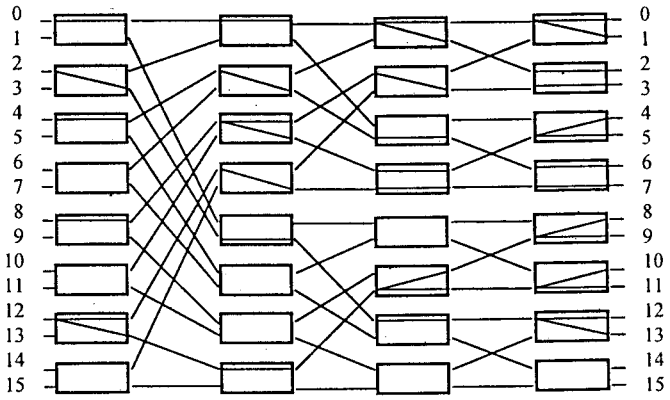

Figure 5. Paths for replicating the five packets in the second pass.

Pass 3

At the end of the 2nd pass, we obtained 14 replicated packets. The 3rd pass works with the 4th pass to perform the paths routing for these replicated packets. The two successive passes set up 14 connections to move the packets toward their final destinations. We need to figure out the intermediate destinations for the 14 packets in the 3rd pass. To do so, we execute the looping procedure with the following connection set on a *16x16* Benes network

$$\begin{pmatrix} 6 & 7 & 3 & 4 & 8 & 0 & 12 & 14 & 9 & 13 & 1 & 10 & 15 & 11 & 2 & 5 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \end{pmatrix}$$

Here we add two artificial connections with idle terminals - $\begin{pmatrix} 14 \\ 7 \end{pmatrix}$ and $\begin{pmatrix} 15 \\ 12 \end{pmatrix}$ - to the original connection set. The reason for this is to obtain a complete permutation set as a requirement by the looping procedure. Figure 6 illustrates the mapping of the 14 paths for moving the packets. Note that the two artificial connections added earlier are removed from this figure.

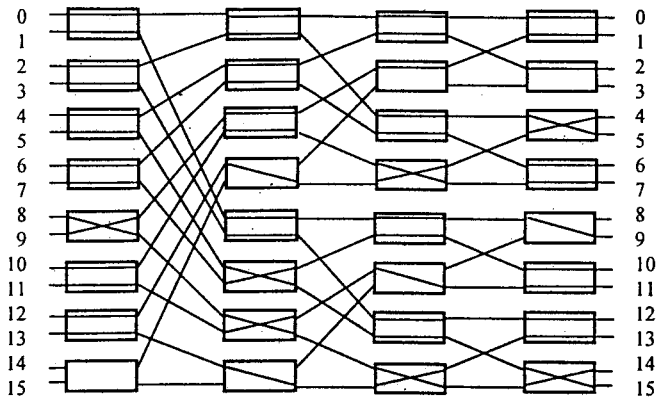

Figure 6. Paths for moving the 13 packets toward their intermediate destinations.

Pass 4

The last pass routes the 14 replicated packets to the final destinations. Routing information for them is quite straightforward. We simply use the respective destination addresses given earlier to route the packets.

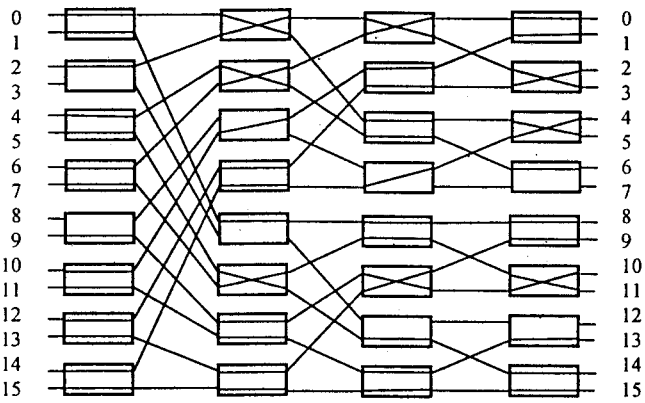

Figure 7. Paths for moving the 13 packets to their final destinations in the last pass.

VII. CONCLUDING REMARKS

This paper has described an efficient routing scheme for multicasting packets on the baseline network. The proposed scheme employs wraparound baseline networks to multicast simultaneous packets from different sources to distinct sets of destinations. This is in contrast to other proposed multicast schemes which use a cascade of various multistage interconnection networks for packets concentration, expansion and distribution. We have proved that the baseline network can realize multiple multicasts in four passes. Of our primary concern is to create link-disjoint paths for moving packets toward their destinations. We have shown that the four passes each involve link-disjoint paths only. As a consequence, nonblocking packet multicast can be accomplished on the baseline network. This differentiates our proposed method from those recursive schemes which recycle packets in a varying number of passes.

REFERENCES

[1] G. W. Richards and F. K. Hwang, A two-stage rearrangeable broadcast switch network, IEEE Trans. on Commun., 33 (10) (Oct. 1985) 1025-1035.

[2] T. T. Lee, Nonblocking copy network for multicast packet switching, IEEE J. Selected Areas Commun., 6 (9) (Dec. 1988) 1455-1467.

[3] J. S. Turner, An optimal nonblocking multicast virtual circuit switch, Proceedings of INFOCOM 94, Toronto, (June 1994) 298-305.

[4] C. T. Lea, A multicast broadband packet switch, IEEE Trans. on Commun., 41 (4) (Apr. 1993) 621-630.

[5] X. Chen and V. Kumar, Multicasting routing in self-routing multistage networks, Proceedings of INFOCOM 94, Toronto, (June 1994) 306-314.

[6] J. park, L. Jacob and H. Yoon, Performance analysis of a multicast switch based on multistage interconnection network, Proceedings of INFOCOM 97, (April 1997).

[7] F. Sestini, Recursive copy generation for multicast ATM switching, IEEE/ACM Trans. on Networking, 5 (3) (June. 1997) 329-335.

[8] R. G. Bubenik and J. S Turner, Performance of a broadcast switch, IEEE Trans. on Commun., 37 (1) (Jan. 1989) 60-69.

[9] Y. Xiong and L. Mason, Multicast ATM switches using buffered MIN structure: a performance study, Proceedings of INFOCOM 97, (April 1997).

[10] R. Lin, C. H. Lam and T. T. Lee, Performance and complexity of multicast cross-path ATM switches, Proceedings of INFOCOM 97, (April 1997).

[11] S. C. Liew, Multicast routing in 3-stage Clos ATM switching networks, IEEE Trans. on Commun., 42, (2-4) (Feb./Mar./Apr. 1994) 1380-1390.

[12] P. Giacomazzi and A. Pattavina, Providing multicast services by the ATM shuffleout switch, Proceedings of Globecom 1994, 1483-1489.

[13] C. L. Wu and T. Y. Feng, On a class of multistage interconnection networks, IEEE Trans. on Comput., 29 (8) (Aug. 1980) 694-702.

[14] D. C. Opferman and N. T. Tsao-wu, On a class of rearrangeable switching networks, Bell System Tech. Journal, (May-June 1971) 1579-1600.