An Efficient Multicast Addressing Scheme for the Self-Routing Multistage Networks

Hong-Ryul Kim and Chae-Tak Lim

Abstract

In this paper, we propose an efficient multicast addressing scheme for the self-routing multistage networks. Using only *N*-bit routing header and the simple hardware logic, the new scheme can efficiently provides all point-to-multipoint connections in single pass through the multistage copy networks. We also designed a hardware logic of switching element to implement the proposed algorithm. Finally, investigation and performance evaluations on the several addressing schemes for implementation of multicasting in ATM switches are performed.

I. Introduction

In future broadband ISDN and ATM network, many types of communication require the transmission of certain information from a source to a given set of destination.

Such could be the case if multipoint video conference, the distribution of a document to a selected number of persons via a computer network or the request for certain information from a distributed database. These types of point-to-multipoint transmission of information are called multicast and there are several proposed methods for multicasting in ATM switches[3, 4, 7, 8].

Many of the switch architectures found in the literature employ multistage interconnection networks(MINs), either buffered or not, as a core routing network(RN) for support of unicast connections. To provide the multicast capability, a copy network(CN) is generally added at the front of a RN as shown in Fig. 1.

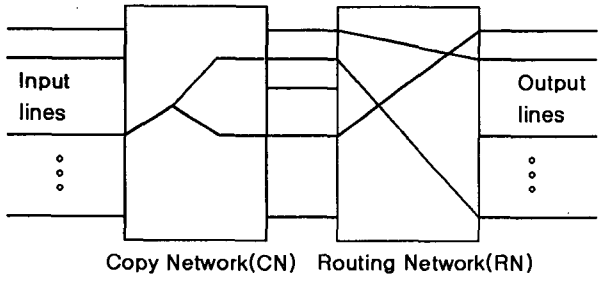


Fig. 1. A multicast packet switch consists of a copy network and a point-to-point routing network.

Manuscript received October 22, 1996; accepted May 2, 1997.

H. R. Kim and C. T. Lim are with Switching theory and parallel processing Lab., Department of Electronic Engineering, Hanyang University.

The CN replicates input packets from various sources simultaneously, and then, copies of multicast packets are routed to their final destination by point-to-point switch. Several multicast addressing schemes are proposed to implement multicasting in self-routing multistage CN[7, 8, 9].

In this paper, we investigate the novel multicast addressing schemes that can provide all point-to-multipoint permutations in single pass through a CN without RN. We propose and analyze the Recursive Partial Header ORing(RPHOR) addressing scheme.

In section II, we describe some multicast addressing schemes and discuss the relation between their routing overhead and their implementations. In section III, we present the new RPHOR multicast addressing scheme and the block diagram of switching elements(SEs).

And we compare the overhead of additional routing bits and the requirement of internal speedup in section IV. Finally, concluding remarks are described in section V.

II. Multicast Addressing Schemes for MIN networks

1. Explicit Addressing Scheme

The explicit addressing scheme is a simple method which puts all the destination addresses into an additional ATM cell header. The cell header structure is shown in Fig. 2. This method has an advantage that all destination addresses can be explicitly examined in the routing header. By examining those valid address fields, a Switching Element(SE) of a multicast path can decide whether to generate a copy to upper link if there is a zero, or to the lower link if there is an one in an appropriate bit of the address fields,

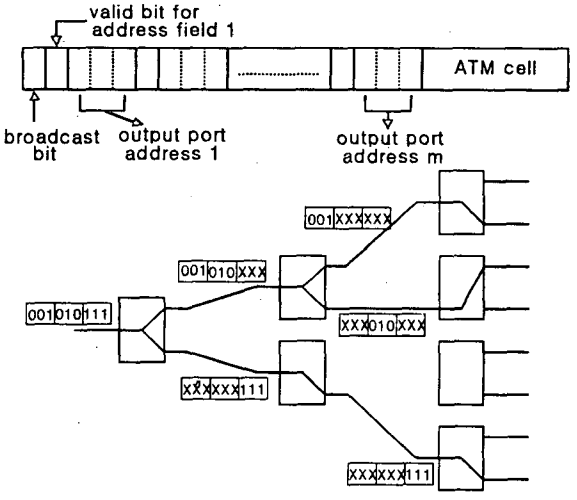


Fig. 2. Explicit addressing scheme.

or it can broadcast to both output links if it is required to do so. When the number of the output ports is N, this scheme needs $\log_2 N$ bits to route a cell to its destination. Another extra bit, i. e. valid bit, is required to specify whether an address field is valid or not. Therefore, the worst case occurs when a cell is required for multicasting (N-1) destinations, then this scheme needs

$$B_{EA} = (N-1)(\log_2 N+1) \quad (1)$$

bits. Therefore, this method needs O($N\log_2 N$) bits in the cell header [1].

This scheme has a serious problem that the packet header contains a variable length of address bits and that the switching elements have to read all of them. Therefore, the structure of SE becomes more complex according to the size of a switch.

2. *Vertex Isolation Addressing Scheme*

The Vertex Isolation Addressing(VIA) scheme is a self-routing multicast addressing scheme. In VIA scheme, the position of a radix-r switching element can be described by the *(i, j)-tuple*, where,

i : vertical level of the radix tree, $i = 0, 1, \ldots \log_r N-1$;

j : vertex from top in level i, $j = 0, 1, \ldots, r^i-1$;

The radix-r tree split based on different vertical levels and these levels are arranged the *(i, j)-tuples* in lexicographical order {(0, 0), (1, 0), (1, 1), (2, 0), ... }. Since each tuple represents a binary element and each binary element has two output links, upper link and lower link, then use two bits to show whether a binary element is required to generate copies to its next descendants. Based on the value of the *(i, j)-tuple*, a specific multicast binary element only needs to examine two predetermined bit locations in a modified ATM cell as shown in Fig. 3.

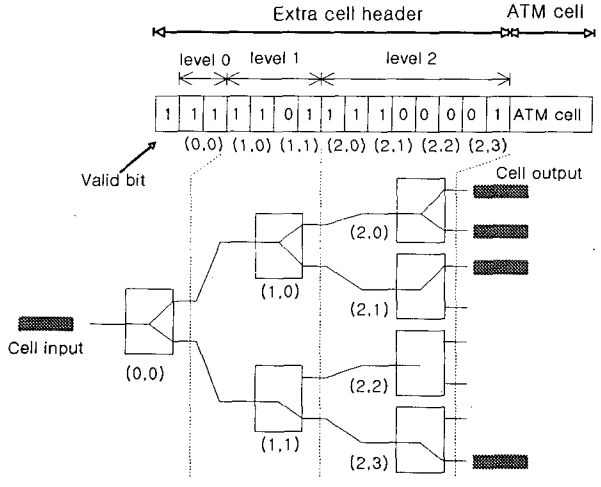


Fig. 3. VIA scheme for multicasting.

In VIA scheme, a VIA address encoder logic is required to generate VIA-address format as shown in Fig. 3 at each input ports and decoder logic to extract a correct *(i, j)-tuple* value at each switching element. A VIA address encoder with 3-bit input (N=8) needs 14 tri-state buffers, 12 AND gates and 17 NOT gates. The hardware complexity of a VIA address encoder is increased by 2(N-1)[1].

The total number of bits required by a radix-r VIA tree for N outputs is

$$B_{VIA} = \frac{r(N-1)}{(r-1)} \quad (2)$$

The value of B_{VIA} is the order of $O(N)$ bits. If r=2, this scheme needs $B_{VIA} = 2(N-1)$ bits[4].

III. Proposed Algorithm

In this section, an efficient self-routing multicast control algorithm, called RPHOR, is proposed and a SE block diagram to achieving RPHOR algorithm is described.

1. *Format of Routing Header*

In proposed multicast addressing scheme, the routing tag bits to multicast a cell are described the *N-bit* for N output ports as follows:

$$(P_0 P_1 \cdots P_k \cdots P_{N-1})_2$$

Each bit of the routing header indicate a destination output port for a multicast request and set to **"1"**, or **"0"** for an active and inactive port, respectively.

If multicast destinations are {0, 1, 3, 4, 7} in a (8×8) switching network, then the routing header bits are set as shown in Fig. 4-(b); where (P_0, P_1, P_3, P_4, P_7) bits are set to **"1"**, and others are **"0"**.

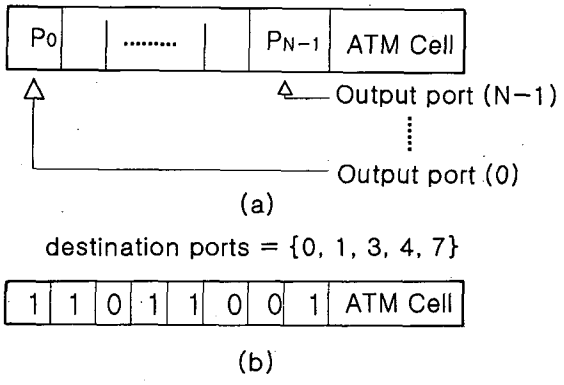


Fig. 4. (a) Routing tag ..... of a multicast address.

Therefore, the total number of bits required by the proposed algorithm is

$$B_{RPHOR} = N \tag{3}$$

the value of B_{RPHOR} is the order of $O(N)$ bits.

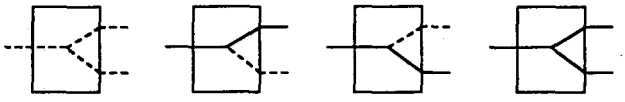
2. *RPHOR Routing Algorithm*

In the multicast switching network, each radix-2 switch has the functions of packet copy and self-routing. Each switching elements on the multicasting path determine whether copy a cell or not by $C_{i,j}(U, L)$-*tuple* value as shown in Fig. 5.

If $C_{i,j}(U, L)$ is **"10"**, then input cell is sent to upper output link only, if $C_{i,j}(U, L)$ is "01", then input cell is sent to lower output link only, and if $C_{i,j}(U, L)$ is **"10"**, then input cell is copied and sent to all output links. Therefore, each radix-2 switch has self-routing characteristics.

To multicasting a cell to several destinations, we propose a new routing algorithm, called RPHOR(Recursive Partial Header ORing), which determine control bits of radix-2 switch at each stage on a multicasting path.

In the proposed algorithm, routing header bits are splitted and ORing at each stage of switching network. The control bits, $C_{i,j}(U, L)$-*tuple*, of a radix-2 switch at each stage is computed by the RPHOR algorithm as shown in Fig. 6. The control bit, $C_{i,j}(U)$, $C_{i,j}(L)$, of *(j-th)* radix-2 switch in level *(i-th)* is determined on the fly by ORing of $\left(\frac{N}{2^{i+1}}\right)$ bits routing header respectively. Each switching element on the multicasting path is switched as shown in Fig. 5, according to the computed control bits. Accordingly, these control bits facilitate the multicasting capability and the self-routing property.



$C_{i,j}(U, L)=00$ $C_{i,j}(U, L)=10$ $C_{i,j}(U, L)=01$ $C_{i,j}(U, L)=11$

Fig. 5. Control conditions of radix-2 switch by the $C_{i,j}(U, L)$.

Recursive Patial Header ORing(RPHOR) algorithm :

For (i, j) binary element :

- Split the routing header up two $\left(\frac{N}{2^{i+1}}\right)$ bits, where $(0 \le i \le \log_2 N-1,\ 0 \le j \le 2^i-1)$;
- Determine the control bit of the upper link by recursive partial header ORing as follows;
$$C_{i,j}(U) = P_0 + \cdots + P_{\left(\frac{N}{2^{i+1}}-1\right)}$$
, where + means a bit-OR operation.
- Determine the control bit of the lower link by recursive partial header ORing as follows;
$$C_{i,j}(L) = P_{\left(\frac{N}{2^{i+1}}\right)} + \cdots + P_{\left(\frac{N}{2^i}-1\right)}$$
- *Control bits of the (i, j) radix-2 switch is*
$$C_{i,j}(U,L) = (\ C_{i,j}(U), C_{i,j}(L)\ )$$

Fig. 6. Recursive Partial Header ORing(RPHOR) algorithm.

We illustrate how the RPHOR algorithm can be used for multicasting in a (8×8) self-routing multistage network. In a radix-2 switch network, a multicasting path can be represented a binary tree as shown in Fig. 7 and each control bit of switch elements on the multicasting path is determined by ORing operation of appropriate bits of the routing header.

Fig. 8 shows an application of the RPHOR addressing algorithm in a (16×16) banyan network. A cell on input port "0" is multicasted to output ports {2, 3, 4, 5, 9, 11, 15}. Each control bit of the switching elements on multicasting path is determined by the RPHOR algorithm as shown in Fig. 8.

To compute the control bits, $C_{i,j}(U, L)$-*tuples*, of a radix-2 switch at each stage by the RPHOR algorithm, some extra OR gate logics are need to perform the RPHOR algorithm at each switching element. For each switching element in $N\times N$ switch, $(N\times1)$ OR gate logic is need at first stage, $\left(\frac{N}{2^k}\times1\right)$ OR gate logic at stage k, (2×1) OR logic at stage $\log_2 N$-2, and no OR logic is needed at last stage. A $(N\times1)$ OR gate can be implemented one NOR gate with N P-MOS transistors and a N N-MOS transistors, and one inverter with a P-MOS transistor and a

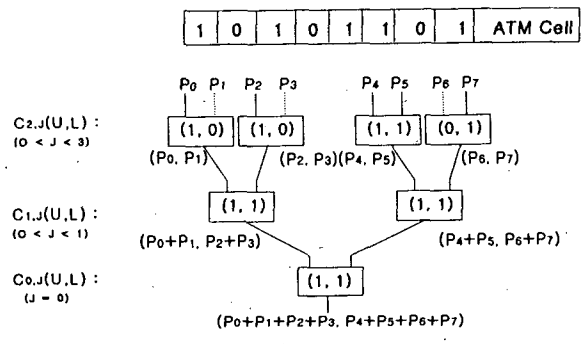


Fig. 7. Example of RPHOR algorithm for N=8.

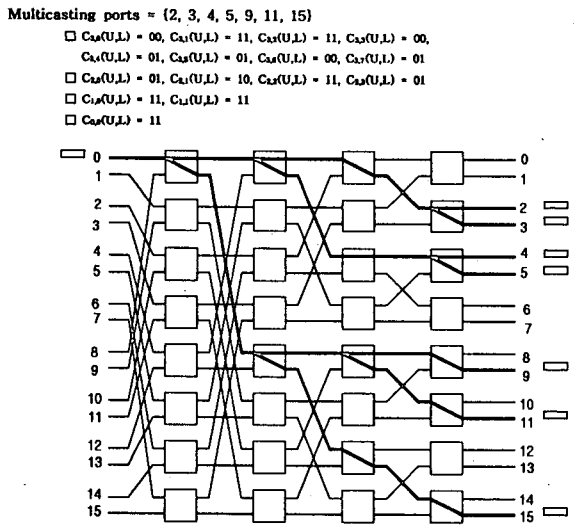


Fig. 8. Multicasting on the banyan network.

P-MOS transistor[5]. Then, the total number of P-MOS/N-MOS transistor is required additionally for the RPHOR algorithm as shown equation (4).

$$T_{total}=2N\sum_{k=0}^{\log_2 N-2}\left(\frac{N}{2^k}+2\right) \quad (4)$$

For a large N system, the number of OR gates of the switching element is too large to implement.

In section 3.3, we provides a simple hardware logic to perform the RPHOR algorithm efficiently.

3. Hardware Logic for the RPHOR

In B-ISDN and ATM network, the multicast packet switch must have capable of fast packet replication and routing. Therefore, the proposed multicast addressing scheme, RPHOR, can be handled by simple hardware implementation.

In Fig. 9, we present block diagram of a radix-2 switching element for our scheme. Each switching element consists of two *Input Controller*(IC) and two *Output Controller*(OC). The IC contains a *RPHOR Logic* and *Packet Controller*(PC).

The RPHOR logic needs five D flip-flops, two transmission gates and one OR gate as shown in Fig. 10. and it determines the

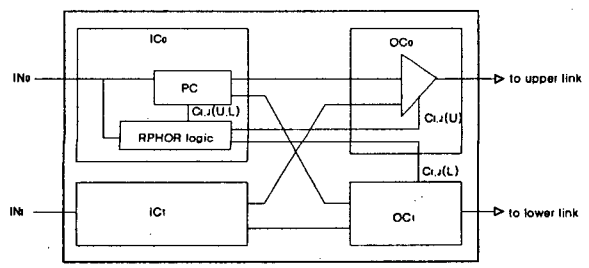


Fig. 9. Hardware logic of radix-2 switch.

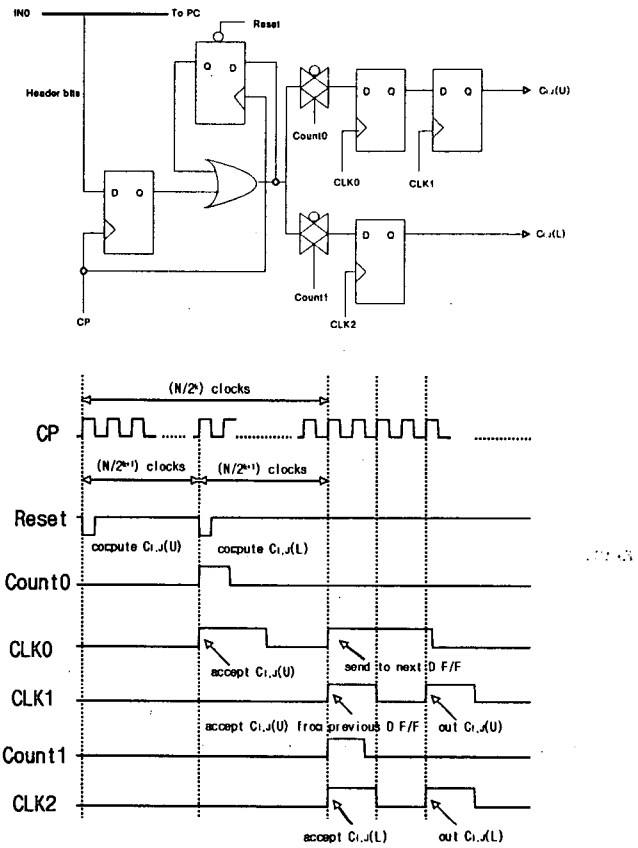


Fig. 10. RPHOR logic and Timing diagram.

$C_{i,j}(U, L)$ value in each stage on the fly by recursive ORing operations. Therefore, we can simplify the complexity of a SE using the simple RPHOR logic. The RPHOR logic of SE has a regular structure at any stage but only needs to be controlled by appropriate clock signals.

By the computed control bits, the PC splits header bits and copy cell. The procedure of packet replication and routing within switch elements at stage k is illustrated in Fig. 11. The packet header is splitted as two $(\frac{N}{2^k})$ bits header. This splitted header leads the cell stream according to its $C_{i,j}(U, L)$ value. The OC's arbitrate requests for the outgoing links by the control bits from the RPHOR logic.

IV. Performance Evaluation

1. Routing Tag Bits and Internal Speedup

To multicast a cell to several destinations in ATM radix-r tree and banyan network, extra routing header bits are required to specify where output ports are valid, and the internal speedup of switch processing capability is also required due to extra header bits. Therefore, an efficient addressing scheme and appropriate routing algorithm are need to improve the performance of multicast packet switch.

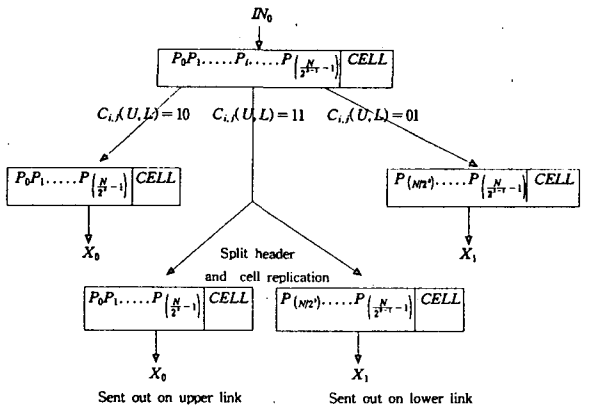


Fig. 11. The procedure of packet replication and routing within a node at stage k.

For a multicast session to perform all output permutations during single pass of input cell, the greatest statistical lower bound of routing tag(B_1) equals to $B_1 = N + \frac{1}{2}\log_2 N + \log_2\sqrt{\frac{2}{\pi}}$ (*when* $N \to \infty$) bits, and the deterministic lower bound (B_2) equals to $B_2 = \log_2(2^N - 1) \cong N$ bits[4]. Both lower bounds have $O(N)$ number of extra bits for representing multicast addresses. Table 1. shows the value of B_1 and B_2 against different values of N.

An ATM switch must operate in higher speed of internal links than that of the external links in order to allow self-routing based on processing of the internal routing header bits [6, 7]. The internal speedup factor due to the multicasting header is defined as shown equation (5).

$$Speedup = \frac{(Header\ bits + Cell\ size)}{Cell\ size} \quad (5)$$

Therefore, if a multicast addressing scheme uses more routing bits, it needs the more internal speedup.

For RPHOR addressing scheme, the length of routing header always increases in proportion to the number of output ports (N). So, the bits of routing header equal to the deterministic lower bound, B_2.

We evaluate the operational costs of RPHOR and other schemes by tabulating equations (1), (2), and (3) for B_{EA}, B_{VIA}, and B_{RPHOR} in terms of extra bits and internal speedup (comparing in byte count) requirement in radix-2 tree network as shown in Table 2.

Table 1. Lower bounds, B_1 and B_2.

N	32	64	128	256	512	1024
$\log_2 N$	5	6	7	8	9	10
[B_1] bits	35	67	132	260	517	1029
speedup	1.094	1.179	1.321	1.623	2.226	3.433
[B_2] bits	32	64	128	256	512	1024
speedup	1.075	1.151	1.302	1.604	2.208	3.415

Table 2. Comparison of multicast addressing schemes.

Addressing scheme	N	32	64	128	256	512	1024
(Header complexity)	$\log_2 N$	5	6	7	8	9	10
Explicit	B_{EA} bits	186	441	1016	2295	5110	11253
($O(\log_2 N)$)	speedup	1.453	2.057	3.396	6.415	13.057	27.547
VIA(r=2)	B_{VIA} bits	62	126	254	510	1022	2046
($O(N)$)	speedup	1.151	1.302	1.604	2.207	3.415	5.830
RPHOR	B_{RPHOR} bits	32	64	128	256	512	1024
($O(N)$)	speedup	1.075	1.151	1.302	1.604	2.207	3.415

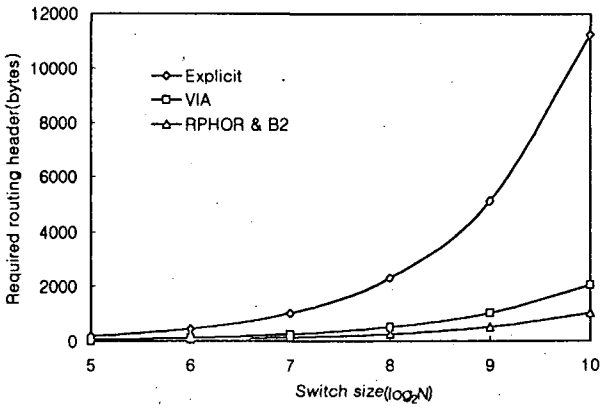


Fig. 12. Comparison of routing header bits.

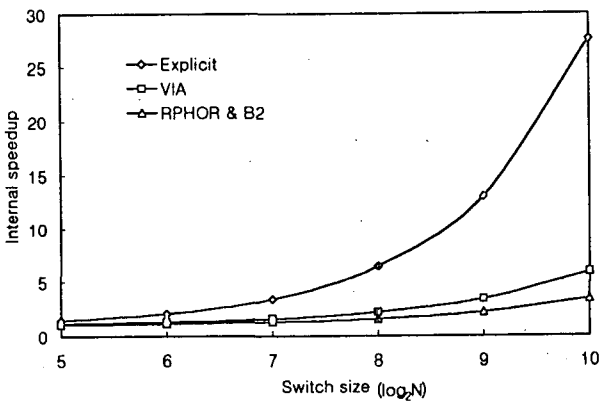


Fig. 13. Comparison of internal speedup.

The internal speedup indicates the processing overhead due to additional routing header in switching network.

From Table 2 and Fig. 12, we can see that the RPHOR scheme needs lower bits compared with other schemes and the required routing bits by the RPHOR equals to the deterministic lower bound.

Also, we see that the RPHOR addressing scheme drastically reduce the speedup requirement over efficient routing algorithm. Table 2 and Fig. 13 show the comparison among three different methods. With internal speedup of around two, a cell with RPHOR is able to specify 512 output addresses, but the explicit

addressing scheme and the VIA addressing scheme are only able to specify 64 and 256 output ports, respectively.

2. Hardware complexity

It's very difficult to compare exactly the required hardware for each addressing schemes. Therefore, we compare the number of the P-MOS/N-MOS transistors which are required to implement the major additional hardware logics for each addressing scheme. For the explicit addressing, each radix-2 switching element needs two $(N\text{-}1)(\log_2 N+1)$ bits comparator logics to interpret the variable length routing tag. A two bits comparator can be implement using a (2×1) AND gate logic. A (2×1) AND gate logic is constructed one CMOS NAND gate, is constructed using the pair of P-MOS and N-MOS transistors, and one CMOS invert with one pair of P-MOS and N-MOS transistor. For $(N\times N)$ switch, total numbers of the P-MOS/N-MOS are as follows;

$$T_{(total\ Tr(s)\ byBit\ Comp.)} = \{(N-1)(\log_2 N+1)\}\times(N\times\log_2 N) \quad (6)$$

In VIA scheme, each input port should has a $2(N\text{-}1)$ bits VIA encoder(VIAE) and each radix-2 switch element needs two $2(N\text{-}1)$ bits counters. For N=8, a VIAE consists of 14 tri-state buffer, 8 (3×1) AND gates, 4 (2×1) AND gates and 17 NOT gates[1]. For simplicity, we consider only the overhead of tri-state buffer for VIA scheme. A tri-state buffer is constructed using 4 P-MOS, 4 N-MOS and one invert gate. For $(N\times N)$ switch, total numbers of the P-MOS/N-MOS by the tri-state buffers of the VIAE are as follows;

$$T_{(total\ Tr(s)\ by\ 3-state\ buffer)} = N\times\{2(N-1)\times 10\} \quad (7)$$

In the proposed scheme, each radix-2 switch element needs two RPHOR logics as shown in Fig. 10. A D flip-flop is constructed using 4 NAND gates. Each NAND gate is constructed using 2 P-MOS and 2 N-MOS transistors. A transmission gate needs one P-MOS and one N-MOS transistor. A (2×1) OR gate is constructed using one NOR gate with 1 P-MOS, 1 N-MOS and one invert gate. For $(N\times N)$ switch, total numbers of the P-MOS/N-MOS by the RPHOR logic are as follows;

$$T_{total\ Tr(s)\ by\ the\ RPHOR\ logic} = 28\times N\times\log_2 N \quad (8)$$

From Fig. 14, we can confirm that the RPHOR scheme needs less hardware in each radix-2 switching elements than conventional addressing schemes.

V. Conclusions

In this paper, we propose an efficient addressing scheme for multicasting environment in ATM radix-2 tree switching network, like as banyan network, using routing bits of the deterministic

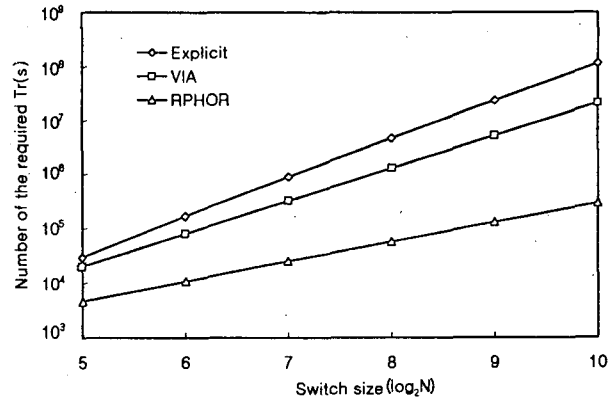


Fig. 14. Comparison of the hardware complexity.

lower bound. The RPHOR scheme retains the self-routing property and shortens the length of the routing tags for multicasting function compared with the explicit addressing scheme and the VIA schemes.

We confirmed that the proposed addressing scheme needs less routing header bits. Moreover, the RPHOR drastically reduced the speedup requirement over other schemes. Then, the RPHOR addressing scheme needs less hardware in each switching elements than the explicit addressing and the VIA scheme. The number of extra routing tag bits and the required internal speedup are also compared.

However, we cannot justify and compare their complexity of the extra hardware exactly. The choice should also depend on designers choice among the provided technology and budget.

References

[1] K. L. Eddie and A. Leon-garcia, "Multicast and Self-Routing in ATM Radix Trees and Banyan Networks," *Proc. IEEE INFOCOM '95*, Vol. 2, pp. 951-959, 1995.

[2] F. A. Tobagi, "Fast Packet Switch Architectures for Broadband Integrated Services Digital Networks," *Proc. IEEE*, Vol. 78, No. 1, pp. 133-167, Jan. 1990.

[3] X. Chen, V. Kumar, "Multicasting Routing in Self-Routing Multistage Networks," *Proc. INFOCOM '94*, Vol. 1, pp. 306-314, 1994.

[4] P. C. Wong and M. S. Weung, "Pipeline Banyan - A Parallel Fast Packet Switch Architecture," *Proc. of ICC '92*, pp. 882-887, 1992.

[5] N. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design-A Systems Perspective*, 2'nd edition, Addision Wesley, 1993.

[6] F. Masetti and C. Raffaelli, "System Design and Evaluation of ATM Replicated Banyan Switches," *Proc. of ISS '92*, pp. 364-368, 1992.

[7] J. S. Turner, "Design of a Broadcast Packet Switching Network," *IEEE JSAC*, Vol. 36, No. 6, pp. 734-743, June 1988.

[8] T. T. Lee, "Nonblocking Copy Networks for Multicast Packet Switching," *IEEE JSAC*, Vol. 6, No. 9, pp. 1245-1467, Dec. 1988.

[9] T. T. Lee, R. Boorstyn, E. Arthurs, "The Architecture of a Multicast Broadband Packet Switch," *Proc. IEEE INFOCOM '88*, Vol. 1, pp. 1-8, 1988.

Hong-Ryul Kim received the B.S. and the M.S. degrees in Electronic Eng. from Hanyang University, Korea, in 1986 and 1988 respectively. Since 1988, he has been working for Hyundai Electronics Industrial Co., as a chief engineer. He is currently working towards the Ph.D. degree in electronic engineering from Hanyang University. His research interests are ATM switching system, Multimedia communication, and High speed cell switching. He is a member of the KITE.

Chae-Tak Lim received the B.S. in Electrical Eng. from Junbuk National University, the M.S. degree from University of Hawaii, and his Ph.D. degree in Electronic Eng. from Hanyang University, Korea, in 1957, 1969 and 1974, respectively. From 1962 to 1976, he was a associate professor in the department of Electrical Engineering at Junbuk National University. Since 1967, he has been at Hanyang University. He was a visiting professor at Technology University of Denmark from 1975 to 1976. He was the president of Korea Institute of Telematics and Electronics in 1991. Currently he is a professor at the department of Electronic Engineering, Hanyang University. His research interests include switching theory, circuit & systems, image processing and parallel processing. He is a member of the KITE, KISS, and KICS.