http://dx.doi.org/10.5755/j01.eee.18.8.2637 ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 18, NO. 8, 2012

Multicast Path Setup Incorporating Evicting

Wenmin Hu[1], Zhonghai Lu[2], Hengzhu Liu[1], Axel Jantsch[2]
[1]*Computer School, National University of Defense Technology,*
Changsha, P.R. China
[2]*Royal Institute of Technology,*
Stockholm, Sweden
huwenmin@nudt.edu.cn

***Abstract*—In this paper, we propose a novel multicast path setup scheme, which incorporates the evicting process. Compared with the previous work, our scheme either overcomes the limitation in evicting times or reduces the setup latency.**

***Index Terms*—Multicast algorithms, multicast communication, multiprotocol label switching, error analysis.**

I. INTRODUCTION

With the development of applications on many-core chip multiprocessors (CMPs) [1], [2], one-to-many communication and one-to-all communication becoming more common. In paper [3], Jerger et al. proposed an efficient multicast scheme called virtual circuit tree multicasting (VCTM) on Network-on-Chip. Their scheme significantly improves the interconnect performance for a broad spectrum of multicasting scenarios with direct and inexpensive extensions to a state-of-the-art packet-switched router.

In VCTM, they used unicast+setup packets to setup the multicast tree incrementally. The unicast+setup packet is routed in the dimension-order (XY), and updates the corresponding entry in the virtual circuit tree (VCT) table. The update operation is determined according to the result of Id matching. If the Id value of a unicast+setup packet is the same as that of a VCT table entry, a new node is being added to an existing tree (just sets one bit in VCT table entry according the routing direction); otherwise, an old tree is being replaced by a new one (clears the whole entry, and sets the corresponding bit according to the routing direction). As an important factor, Id is stored at both VCT table entry and Destination Set CAM, which is integrated in source node and contains all of its currently active multicast trees. Generally, VCT entry and Destination Set CAM have the same Id. The variation of Id happens when the source node wants to initiate a new multicast communication, of which the destination set is missed in the Destination Set CAM. Hence, an active multicast tree is selected to be evicted, and the Id bit will be changed (eg. if the Id of former VCT is 0, and the new Id will be changed to 1). Then, multicast setup is initiated by injecting several unicast+setup packets, each of which is sent to one destination node. These unicast+setup packets have the same VCT number as the evicted one and the same new Id. While being routed, these packets update the VCT table according to the result of Id matching. After all the packets have been injected into destinations, a new multicast tree is constructed successfully. Unfortunately, there is a potential limitation in this approach. Since the Id bit width is 1, a destination of former multicast tree may be added to the latter multicast tree by mistake, when the number of times to evict the same VCT table entry exceeds two (eg. in the third time, the Id is 0 again, which is the same as the first multicast group).

To avoid the above problem, one may consider using more bits for Id. However, increase in Id bit width only delays the error occurrence, let alone that it increases the area of VCT table and the packet payload. It cannot avoid errors when the number of evicting times exceeds 2^n, where n represents the Id bit width.

One possible solution is to introduce a new type of packet, which is routed like multicast data packet and clears corresponding VCT table entry after getting the outgoing directions [4]. This approach, however, separates the process of setup and evicting, which badly prolongs the setup latency.

In this paper, we first give an example to illustrate the limitation of VCTM. Then we propose a novel multicast path setup scheme incorporating with evicting process. In section 4, we give the result of area evaluation. Finally, the conclusions are given.

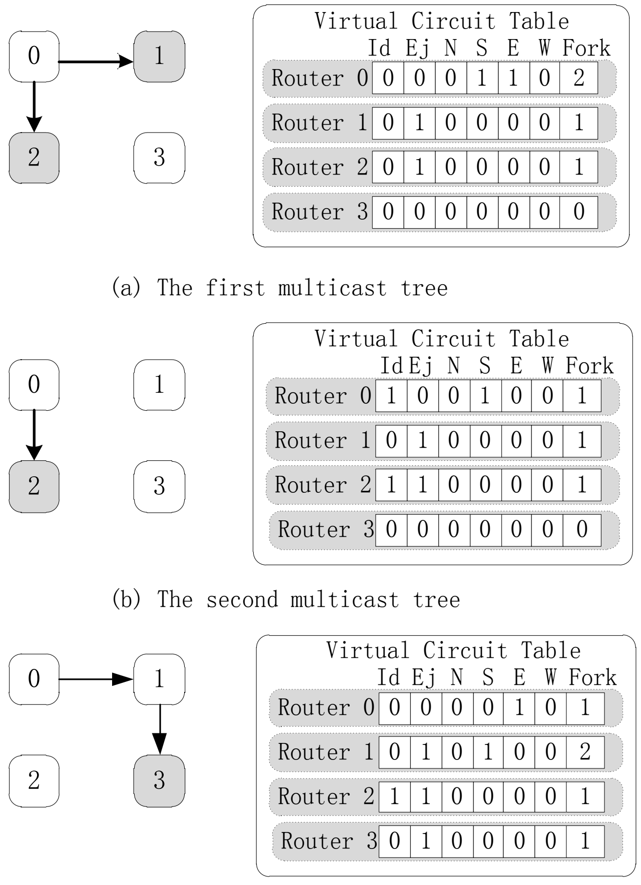
II. EXAMPLE OF ERROR OCCURRENCE

As an example, Fig. 1 walks through the process of error occurrence. To facilitate the explanation, we assume that there is only one entry in the VCT table for each source node; the multicast destination set size could be one.

In the first scenario, as shown in Fig. 1(a), a multicast tree, of which the Id fields are set to 0, from node 0 to node 1 and node 2 is constructed. The S, E field of Router 0's VCT table entry are set to 1, indicating that multicast packet shall be replicated at Router 0 and forwarded to the S port and E port. The Ej (Ejection) fields of router 1 and router 2 are set to 1, meaning that they are the destinations of the multicast. After a time, source node 0 wants to initiate a multicast packet to node 2. Since the destination set is missed in the Destination Set CAM, a new multicast tree will

The research is supported by National Natural Science Foundation of China with Grant No. 60970037, and by Doctor Program Foundation of Education Ministry of China with Grant No.20094307110009.

be constructed. The unicast+setup packet's Id field is set to 1, since the former one is 0. According to [3], matching Id bits indicate that a new node is being added to an existing tree while differing Id bits indicate an old tree is being replaced by a new one. Thus, during the setup period, the VCT table entries in router 0 and router 2 will be cleared and set according to the output direction computed by routing computation unit, as shown in Fig. 1(b). Note that the VCT table entry in router 1 is unchanged, because no unicast+setup packet traverses or updates it. When source node 0 sends a multicast packet to node 3, the Id of unicast+setup packet will be changed back to 0. Upon arrival at router 1, the VCT table entry ought to be cleared and the South bit should be set to 1. Unfortunately, since the Id bits are the same, the Ej bit is not cleared, which means that the subsequent data packets are to be injected into node 1 by mistake (Fig. 1(c)). The error occurs because the later multicast tree's Id is identical to the former one, and the entry of former multicast tree has not been evicted completely.



Virtual Circuit Table

	Id	Ej	N	S	E	W	Fork
Router 0	0	0	0	1	1	0	2
Router 1	0	1	0	0	0	0	1
Router 2	0	1	0	0	0	0	1
Router 3	0	0	0	0	0	0	0

(a) The first multicast tree

Virtual Circuit Table

	Id	Ej	N	S	E	W	Fork
Router 0	1	0	0	1	0	0	1
Router 1	0	1	0	0	0	0	1
Router 2	1	1	0	0	0	0	1
Router 3	0	0	0	0	0	0	0

(b) The second multicast tree

Virtual Circuit Table

	Id	Ej	N	S	E	W	Fork
Router 0	0	0	0	0	1	0	1
Router 1	0	1	0	1	0	0	2
Router 2	1	1	0	0	0	0	1
Router 3	0	1	0	0	0	0	1

(c) The third multicast tree

Fig. 1. An example of error occurrence in the VCTM, and the proposed solution.

III. PROPOSED SCHEME

In this paper, we follow the main idea of VCTM, using multiple unicast+setup packets to construct the multicast tree incrementally. Here we propose a novel approach, which incorporates the evicting with the multicast path setup and is free of evicting times. To avoid adding undesirable destination to new multicast tree, some essential issues must be considered carefully. For instance, when and where to clear the entry, when and where to update one bit of entry. Here we propose three strategies:

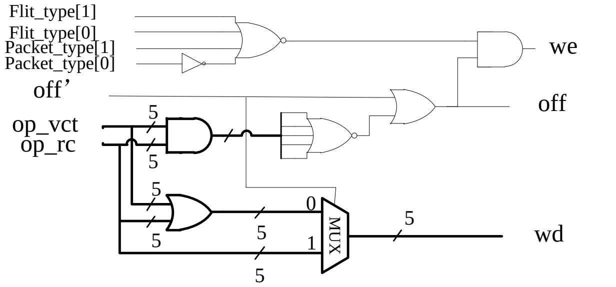
1. If the unicast+setup packet is on the path which is constructed by former unicast+setup packets of the same multicast group, and outgoing direction is a subset of directions of the entry, no operation is applied to the VCT table.

2. If the unicast+setup packet is on the path which is constructed by former unicast+setup packets of the same multicast group while outgoing direction is not a subset of directions in the entry, which means that the packet is about to leave the path or be injected into the local port, the corresponding bit of the VCT table entry is set according to the outgoing direction.

3. If the setup packet is off the path constructed by former unicast+setup packets of the same multicast group, all the bits of VCT table entry will be cleared and the bit will be set according to the outgoing direction.

Head/Body/Tail	MC/UC	off	Vct#,Src	UC Dst	VC#	Payload(Command/Addr)
2 bits	2 bits	1 bit	10 bits	4 bits	3 bits	
00-Head	01 setup	1	0	1	0	

(a) Setup packet format



(b) Logic for setup scheme

Fig. 2. The proposed solution for multicast path setup.

In order to support this scheme, we need one bit in a packet to check whether this packet is on the path constructed by former unicast+setup packets from the same multicast group. We also need additional hardware logic to implement the three aforementioned strategies. As shown in Fig. 2(a), a field called *off* is set in the unicast+setup packet instead of *Id*. Fig. 2(b) shows the logic circuit implementing the proposed strategies, in which *op_vct* is the output directions read from the VCT table according to VCT#, while *op_rc* is the output direction computed by the routing computation unit. The signal *off* is the new value of the *off'*. The signal *we* is the write enable signal to the VCT table, while *wd* is the new value being updated to the entry. The new value will be written to the VCT entry only in two cases. The first case is that, when a setup packet's outgoing direction (*op_rc*) doesn't belong to the directions read from the VCT (*op_vct*) and the *off'* is 0, which is considered in strategy 2, the *op_rc* will be added to the VCT entry, and the *off* field of the packet will be changed into 1 simultaneously. The other case is when the *off* bit is 1, and the *op_rc* is written to the entry directly, which is considered in strategy 3. Once the *off* bit is set to 1, it will never be changed again. As Strategy 1 reduces the number of writing times compared with the VCTM, our scheme is more power-efficient.

Strategy 2 and Strategy 3 solve the problems of the VCTM scheme.

As for the initialization of the *off* bit in a unicast+setup packet, if the unicast+setup packet is the first one injected into network, the original value is 1, and it will clear and reset all the entries from the source router to the destination routers. For other packets, the original values are 0, and they will be changed to 1 when they are about to leave the path constructed by former setup packets. For switch allocation, the selection strategy should be first-come-first-serve (FCFS), which maintains the order of unicast+setup packets.

Two conditions are necessary for error occurrence in the VCTM scheme: (1) The current VCT table entry has not been cleared; (2) The operation to the current entry is just setting some bit. With our proposal, the first unicast+setup packet's initialization achieves the first VCT entry set that are cleared. The subsequent unicast+setup packets with the three strategies are error free, because each of the strategies prevents from satisfying the two necessary conditions simultaneously.

IV. Example of Proposed Scheme

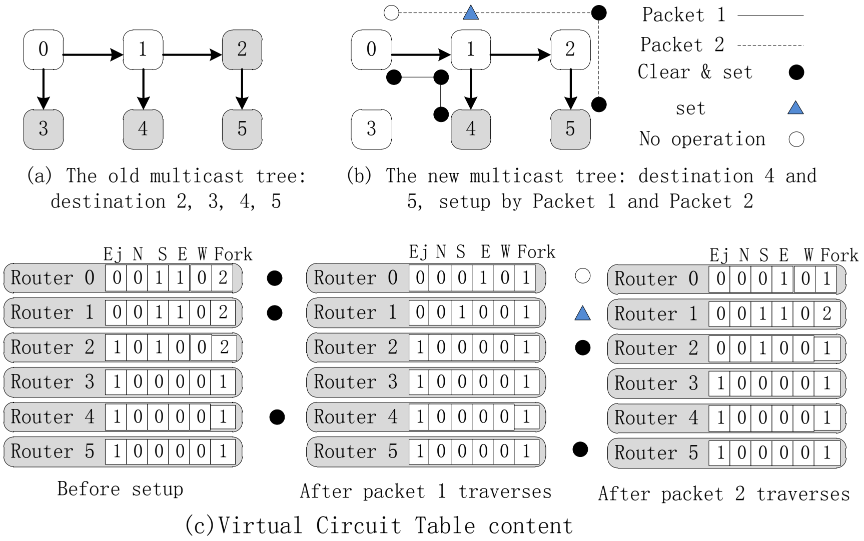
Fig. 3 shows an example for proposed scheme. As can be seen in Fig. 3(a), an existing multicast tree source from node 0, destined for node 2, 3, 4 and 5, which will be replaced by a new multicast tree destined for node 4 and 5. As can be seen in Fig. 3(b), two unicast+setup packets are initiated at the source node 0, destined for node 4 and 5. The first unicast+setup packet clears and sets the table entries in the router 0, 1 and 4. Due to the sub-path (0-1-4) constructed by the first packet, the second one doesn't update the entry of router 0's table (strategy 1). It sets the east bit of router1's table entry (strategy 2), and then clears and sets the entries of router 2, 5 (strategy 3). Fig. 2 (c) shows the change of VCT table for each router during the setup process.

V. Area Evaluation

The VCT and proposed setup logic have been synthesized using a 90 nm CMOS stand-cell technology library from Chartered Semiconductor Manufacturing. Table I shows the synthesis results. Because the Id has been eliminated in our scheme, we evaluated the VCT tables whose entry width is 8 bits (directions: 5, fork:3) and 9 bits (directions:5, fork:3, Id:1), where the difference lies in the area of Id bits. Since there are five setup logic circuits in one router, the area of proposed mechanism is as five times as large as a single setup logic. Compared with the Id induced area, our scheme reduces the area from 90.07% to 98.73% when the number of VCT entries increases from 64 to 512. Furthermore, considering the compare logic (needed for Id matching) in VCTM is removed, our scheme saves more area.

TABEL.1. THE AREA OVERHEAD OF THE VCT AND THE PROPOSED SCHEME (MM^2)

Entries	VCT (9bits)	VCT (8bits)	Id	Setup logic	Reduction
64	23631	21002	2629	261	90.07%
128	47109	42005	5104	261	94.88%
256	94004	83637	10367	261	97.48%
512	185762	165202	20560	261	98.73%



(a) The old multicast tree: destination 2, 3, 4, 5

(b) The new multicast tree: destination 4 and 5, setup by Packet 1 and Packet 2

Before setup

	Ej	N	S	E	W	Fork	
Router 0	0	0	1	1	0	2	●
Router 1	0	0	1	1	0	2	●
Router 2	1	0	1	0	0	2	
Router 3	1	0	0	0	0	1	
Router 4	1	0	0	0	0	1	●
Router 5	1	0	0	0	0	1	

After packet 1 traverses

	Ej	N	S	E	W	Fork	
Router 0	0	0	0	1	0	1	○
Router 1	0	0	1	0	0	1	▲
Router 2	1	0	0	0	0	1	●
Router 3	1	0	0	0	0	1	
Router 4	1	0	0	0	0	1	
Router 5	1	0	0	0	0	1	●

After packet 2 traverses

	Ej	N	S	E	W	Fork
Router 0	0	0	0	1	0	1
Router 1	0	0	1	1	0	2
Router 2	0	0	1	0	0	1
Router 3	1	0	0	0	0	1
Router 4	1	0	0	0	0	1
Router 5	1	0	0	0	0	1

(c)Virtual Circuit Table content

Fig. 3. An example for the proposed setup scheme.

VI. Conclusions

In this paper, we first pointed out a limitation in the multicast path setup scheme in VCTM [3]. In VCTM, when the number of times of evicting the same table entry exceeds 2, undesired destinations may be added to current multicast tree. We exemplified the error occurrence, and analyzed the problem. Based on the analysis, we presented an effective remedy to overcome this problem. Furthermore, our scheme is more area-efficient.

References

[1] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, S. Borkar, “A 5-GHz mesh interconnect for a Teraflops processor”, *IEEE MICRO*, vol. 5, no. 27, pp. 51–61, 2007. [Online]. Available: http://dx.doi.org/10.1109/MM.2007.4378783

[2] D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C.-C. Miao, A. Agarwal, “On-Chip Interconnection Architecture of the Tile Processor”, *IEEE MICRO*, vol. 5, no. 27, pp. 15–31, 2007. [Online]. Available: http://dx.doi.org/10.1109/MM.2007.4378780

[3] N. E. Jerger, L.S. Peh, M. Lipasti, “Virtual circuit tree multicasting: A case for on-chip hardware multicast support”, in *Proc. of the 35th annual international symposium on Computer architecture*, 2008.

[4] W. Hu, Z. Lu, A. Jantsch, H. Liu, “Power-efficient tree-based multicast support for networks-on-chip”, i*n Proc. of 16th Asia and South Pacific Design Automation Conference (ASP-DAC11)*, 2011.