Power-efficient Tree-based Multicast Support for Networks-on-Chip

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

Abstract— In this paper, a novel hardware supporting for multicast on mesh Networks-on-Chip (NoC) is proposed. It supports multicast routing on any shape of tree-based path. Two power-efficient tree-based multicast routing algorithms, Optimized tree(OPT) and Left-XY-Right-Optimized tree (LXYROPT) are also proposed. XY tree-based(XYT) algorithm and multiple unicast copies (MUC) are also implemented on the router as baselines. Along with the increase of the destination size, compared with MUC, OPT and LXYROPT achieve a remarkable improvement both in latency and throughput while the average power consumption is reduced by 50% and 45%. Compared with XYT, OPT is 10% higher in latency but gains 17% saving in power consumption. LXYROPT is 3% lower in latency and 8% lower in power consumption. In some cases, OPT and LXYROPT give power saving up to 70% less than the XYT.

I. Introduction

Many-core architectures have become the mainstream for designing System-on-Chip. The traditional bus structure is unable to meet the requirement of performance and is not scalable. The Network-on-Chip (NoC) is proposed to solve the global interconnection problems of these systems. Some NoCs have been developed, such as NOSTRUM [1], RAW [2], TRIPS [3], SPIN [4] etc.

NOC architectures supporting for unicast can be used to implement multicast traffic by replicating multiple unicast messages to different destinations, however this way is inefficient. If the destinations group is large, the injection port of the source node will have difficulty to inject all the packets into NoCs in a limited period, which leads to late startup time for some destinations. Therefore the architectures that support multicast are promising solutions to the scenario.

This paper proposes two power-efficient tree-based multicast routing algorithms for Mesh NoC. OPT tries to use the minimum number of links to construct the multicast tree. LXYROPT, on the premise of keeping the low latency, tries to minimize the links number of the multicast tree. Compared with XYT algorithm, OPT is 10% higher in latency but gains 17% saving in power consumption. LXYROPT is 3% lower in latency and 8% lower in power consumption.

The paper is organized as follows. In section II, a brief review of the related works is presented. In section III, the multicast mechanism and algorithm realization are discussed. In section IV, the multicast protocol and implementation is described. The results of experiment are shown in Section V. The conclusion and future work are given at the last section.

II. Related works

Multicast on off-chip network were well researched in [5, 6, 7, 8]. The results show that multicast on off-chip network has outstanding effect on improving the performance. Benefit from multicast on off-chip network should also be suitable for the communications on-chip. Æthereal [9] and Nostrum [1] both declared multicast is supported on their NoC architecture. Connected-oriented multicasting in wormhole-switched NoC has been presented in [10]. In their scheme, multicast includes establishment, communication and release phases. For the reason that they used just one setup packet to build the whole path, the latency of the setup is long [10]. An ID-tag-based multicast routing is proposed in [11]. In this paper, a flexible mechanism to manage broadcast-flow to share the communication link in NoCs is presented. They used an ID-tag to manage each multicast flow [11]. A tree-based routing algorithm which has hardware supporting for multicast (VCTM) was presented in [12]. Similar to [11], VCTM used ID-tag-based Multicast routing named virtual circuit table (VCT). It constructs the multicast tree incrementally by sending a unicast packet to each destination node. Each setup packet is routed by Dimension-Ordered Routing(DOR) algorithm. The method has the advantage of low latency for the packet . But to some case, it is not power efficient for the destinations that are distributed along the X-axis.

Since the tree-based approaches are easily blocked at branches. They perform badly in the case of high traffic load. Path-based approaches are also be researched to overcome the shortcoming of tree-based. One path-based routing algorithm is Hamilton path algorithm where a unidirectional Hamilton path of the network is constructed [13]. It can be organized as dual path (DP) and multi path (MP) [13]. Another way is to organize based on column path (CP), destinations are partitioned into the $2k$ subsets, that the k is number of the columns in the mesh [14]. An LD path-based algorithm is presented in [15], this paper partitioned the destination into four sets, just like MP [13], but did some optimization on path distance of the subset [15]. Path-based also has a shortcoming at the long latency for the packet transferring. It can be optimized by partitioned the destinations to many sets [13, 14], but it increases the possibility of the injection contention at the source node and reduces the sharing of the link.

III. Multicast Mechanism and Algorithm Realization

In this section we propose two Multicast algorithms which both of them are based on tree. A tree can be decomposed to serveral node pairs. The first node of the pair is considered as starting point of a branch while the second node is the end point. The multicast tree is built incrementally by adding branch one by one to the existing tree. The initial tree is just the source node. The first step of constructing multicast tree is to find all the node pairs that forms the tree. The same destinations may be covered by different shape of trees. Different shape of trees may cause different performance and power consumption. The proposed algorithms both are power-efficient.

A. Mechanism

We take an example to illustrate the mechanism of constructing a tree. Fig. 1 shows a multicast tree on a 3×3 mesh NoC. A is the source node, while C and B are the destination nodes and D is the branch node. Assume we use XY routing algorithm to construct the branch of the tree. The tree is decomposed to two branches which is represented by two node pairs: (A,B) and (D,C)

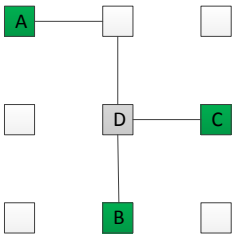

Fig. 1. Example for tree build mechanism

B. Optimized Tree (OPT) Algorithm

OPT is an optimized tree based on west-first turn model [16], which avoids deadlock on mesh network. In order to minimize the link number of the tree, an algorithm similar to minimal spanning tree algorithm is proposed, this make the tree power-efficient. The algorithm is shown in Fig. 2. D_{pair} is defined as the pair set that forms the tree. D_{node} is the set of the nodes covered by the existing tree. It contains the forwarding nodes which are not the destination nodes. It is used as candidates of branch node for proposed algorithm. D is the set of destination group. The first step is to add the most western node to the multicast tree. Add all the nodes in the path that from source node to the most western node into D_{node}. This makes it possible that later finding a node in D_{node} to connect other destination node to conform to the west-first turn model. Add the source node and the most western node as an element of D_{pair}. The most western node is removed from the D. Then if the D is not empty, the algorithm enters a stage similar to minimal spanning tree constructing. Firstly, find a nodes pair (u, v) from D_{node} and D which has the shortest distance and conform to west-first turn model. If there are some pairs have the same shortest distance, Select the pair that v is more western, this makes it more possible later to find a node pair with shortest distance and less branches. Secondly, add (u, v) to D_{pair} . Add all the nodes in the path from u to v to the set D_{node}. Remove v from D. If D is not empty, the sequence will be repeated. By the way, it is possible to find a node pair that u and v are the same node, for the reason that the first path may contain the destination nodes, it doesn't matter, just put the pair to D_{pair}. When D is empty, the procedure is finished.

Algorithm: Generate the optimized multicast tree based on west-first turn model
Input: Destination set D, Source node $s : (x_0, y_0)$;
Output: Pair setD_{pair};
Define: $k(a, b) = |a.y - b.y| + |a.x - b.x|$;
Initial: $D_{node} \leftarrow s, D_{pair} \leftarrow \emptyset$;

1: Find the node $v \in D, \forall a \in D, v.y \leq a.y$. Add (s, v) into D_{pair}, removed v from D, add the nodes on the path from s to v into D_{node}
2: **while** D is not empty **do**
3: $D_{pair\_tmp} \leftarrow \{(u, v) | u \in D_{node}, v \in D, that \forall a \in D_{node}, \forall b \in D, k(u, v) \leq k(a, b)\}$
4: Select $(u, v) \in D_{pair\_tmp}$, that $\forall (a, b) \in D_{pair\_tmp}$, $v.y \leq b.y$
5: Add (u, v) into D_{pair}, remove v from D, add the nodes on the path from u to v into D_{node}
6: **end while**

Fig. 2. Algorithm for generating OPT.

C. Left-XY-Right-Optimized Tree (LXYROPT) Algorithm

OPT is a power-efficient multicast algorithm which optimizes the multicast tree generation globally by using less links, so this may increase multicast latency. To obtain both low latency and low power consumption, we propose another algorithm named Left-XY-Right-Optimized tree (LXYROPT). In this algorithm, the destination set is partitioned to two subsets. One contains the nodes that are left of source node, the other contains the rest. For the destinations that are left of the source node, XY algorithm is used to generate the multicast path. For the rest node, the algorithm takes both the minimum hops for each node and the link sharing into consideration. Fig. 3 shows the detail of LXYROPT. For the optimization, firstly, it should be make sure that the routing distance from source node to destination node on multicast tree is the same as the Manhattan distance from the source node to destination node. Base on this, we select the node u from D_{node}, v from $D_{mid-right}$ that the Manhattan distance between u and v is the minimum, this means that a new destination node is added to existing tree with minimum links. Similar to OPT, the pair(u, v) is added into the D_{mrpair}, v is removed from the $D_{mid-right}$, the nodes from u to v are also added into D_{node}. If $D_{mid-right}$ is not empty, the sequence will be repeated. Otherwise, the procedure is finished.

Algorithm: Generate the LXYROPT multicast tree based on west-first turn model
Input: Destination set D, Source node $s : (x_0, y_0)$;
Output: Pair set D_{lpair}, D_{mrpair};
Define: $k(a, b) = |a.y - b.y| + |a.x - b.x|$;
Initial: $D_{node} \leftarrow s, D_{lpair} \leftarrow \emptyset, D_{mrpair} \leftarrow \emptyset$;

$D_{left} \leftarrow \{(x, y)|(x, y) \in D, y < y_0\}$
$D_{mid-right} \leftarrow \{(x, y)|(x, y) \in D, y \geq y_0\}$
while D_{left} is not empty **do**
    Find a node $v \in D_{left}$, Add (s, v) into D_{lpair}, remove v from D_{left}
end while
while $D_{mid-right}$ is not empty **do**
    $D_{pair\_tmp} \leftarrow \{(u, v)|u \in D_{node}, v \in D_{mid-right}$, that $k(s, v) = k(s, u) + k(u, v)$
    Select $(u, v) \in D_{pair\_tmp}$, that $\forall (a, b) \in D_{pair\_tmp}$, $k(u, v) \leq k(a, b)$
    Add (u, v) into D_{mrpair}, remove v from $D_{mid-right}$, add the nodes on the path from u to v into D_{node}
end while

Fig. 3. Algorithm for generating LXYROPT.

D. An example for proposed algorithms

Fig. 4 shows a multicast tree built with different algorithms. The black node is the source node , the green nodes are the destination nodes. The Fig. 4.a is XYT, which contains 18 hops to finish a multicast. Fig. 4.b is the OPT, which contains 13 hops. Fig. 4.c is LXYOPT, which contains 16 hops. The multiple unicast contains 20 hops. So our proposed algorithms can reduce the total hops for multicast effectively.

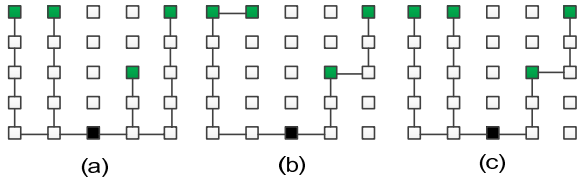

Fig. 4. Example for XYT, OPT, LXYROPT

IV. Multicast protocol and implementation

A special router architecture is proposed to support multicast. We organize this section as follow: firstly, we will introduce the packet format which is suitable for the multicast and unicast. Secondly, the microarchitecture of router is presented. Finally, we will discuss about deadlock-freedom of protocol.

A. Protocol

Each multicast forms a tree connecting the source with the destination set, which is identified by a Multicast ID number to each source node and its destination nodes combination. For a tree-based approach, multicast packet travels along a common path until it arrives at the branch node, where it sends copies of the packet to each branch and continue. Once a multicast tree is setup, the packet will be routed based on the multicast table (MCT) number at each router. At the source node, a table recording the destination set is composed by *n* bits (*n* represents the number of the nodes on NoC) , if the bit is set, means that corresponding node is the destination node. A bit indicating whether the entry is valid is also included. At each router, MCT is partitioned to *n* sub tables corresponding to each source node. Each sub table has 16 entries or more.

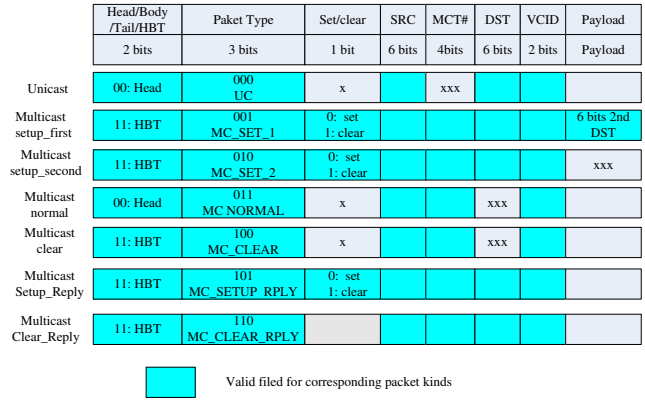

Fig. 5. Packet format.

The proposed router supports any shape of tree-based multicast, because incrementally building up is adopted by decomposing the tree to many branches (pair set). Multicast setup is divided into two periods. During the first period, The setup packet is named MCT_SET_1, whose destination field (DST) is filled with the ID of the first node of the pair from pair set that generated by the proposed algorithms while the second node ID is filled into the first 6 bits of payload field. The packet is routed to the destination just like unicast. While the MCT_SET_1 arrives at the input port of the destination, The Packet Type field of the packet is changed to MCT_SET_2, DST is covered by the first 6 bits of Payload field. Then setup process enters the second period, while traveling to the new destination the corresponding multicast table entry is being updated based on the result of routing. Once the MCT_SET_2 reaches destination, a multicast reply packet (MC_SETUP_REPLY) is sent to source. The setup packet can be sent out without waiting for getting the reply of the former setup packet. Each branch of the tree can be built simultaneously. When the source node receives replies of all the destinations, the setup is finished.

Sometimes when MCT is full, we have to generate a new multicast tree. Therefore we should clear an existing multicast tree. Only the source node has the right to evict the multicasts tree. MC_CLEAR packet will be routed by looking up MCT, after get the routing result, the corresponding table entry will be cleared. When the MC_CLEAR packet sinks at the destination node, the destination node will generate a reply packet (MC_CLEAR_RPLY). Once the source node receives all the reply packets, the multicast tree is evicted. Fig. 5 shows the formats of all the packets that our router supports.

B. Microarchitecture

The router microarchitecture is shown in Fig. 6. The unicast packets are routed via existing hardware (XY routing e.g.).

The multicast normal packets are routed by looking up the multicast table.

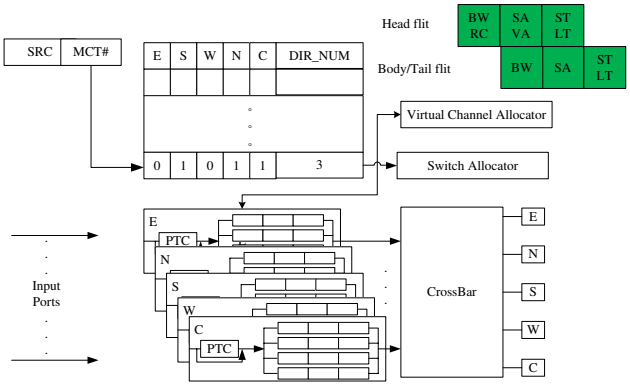

Fig. 6. Router architecture.

Fig. 6 also shows the pipeline stages of the proposed router. Both unicast and multicast follow the pipeline stages: buffer write/routing computation (BW/RC), switch allocation/virtual channel allocation (SA/VA), switch traversal/line traversal (ST/LT). When a head flit arrives at input port, the routing computation begins on the field of DST (unicast) or MCT#(multicast). The only exception is for the MCT_SET_1 packet. If the current router is the destination, the first 6 bits of payload field will be sent to routing computation unit instead of DST field. If the flit is the head flit of MCT_SET_1, during the BW, the packet type convert logic (PTC) block will check if the destination is current router. If so, the PTC will change the packet type filed of the flit from MC_SET_1 to MC_SET_2, and copy the first 6 bits of payload to the DST field. This means that the multicast setup enters the second period.

For unicast routing, the SA and VA are executed at the next cycle, both SA and VA must succeed at the same time, otherwise will retry next time. If the head flit is granted in both SA and VA, the flit traverses the crossbar and finally is transferred to the downstream router.

For the multicast packet routing, the RA may return multiports. The flit is replicated to one port at one SA/VA stage when successfully gets the grant. Only when the flit is successfully transferred to all the destination ports, can the flit be deleted from buffer. To keep the state of each multicast flit, the input virtual channel (VC) reserves a separate VC state register and buffer pointers. It is necessary to forward and control the pipeline stage by using the state register and buffer pointers. If the port belongs to the RC results, then its state register will be set to advanced to SA/VA, otherwise the state will be idle. The buffer pointers contain a head pointer and tail pointers for an input VC, 5 read pointers for all the destination ports. They will be discussed on subsection of deadlock-free.

C. Deadlock-free

To avoid deadlock, multicast tree should be generated based on the turn model that can never cause deadlock, for example XY, West-first, North-last[16]. Another possibility is that the deadlock may happen when multiple multicast existing on the network. As shows in Fig. 7.a, two flits request both north and south output ports, but none of them get two output ports at the same time. The packet *a* holds the southern input channel of the router 2 and tries to forward the head flit to the northern input channel of the router 3. The packet *b* holds the northern input channel of router 3 and tries to forward the head flit to the southern input channel of router 2, so the deadlock is happen.

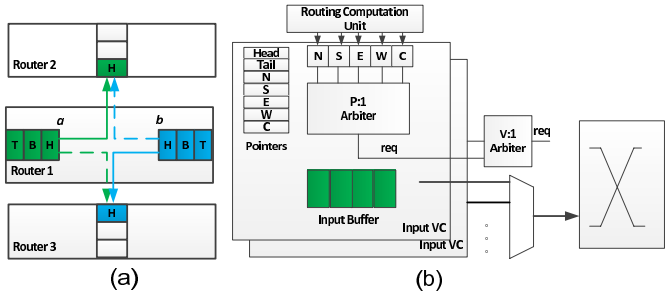

Fig. 7. Deadlock and input virtual channel architecture.

There are some researches to solve this kind deadlock. Partitioning and systematically allocating virtual channels is presented to avoid deadlock in [14]. Kumar proposed a hardware tree-based multicast routing algorithm with deadlock detection and a recovery mechanism [17]. Young proposed a dynamic packet fragmentation to solve the deadlock [18]. This scheme allows to release the hold of an output virtual channel (VC) and enable other packet to use the freed VC when deadlock happens.

The proposed router avoids deadlock by using enough buffer to hold entire packet and reserving buffer pointers for each port in input VC, it can be seen in Fig. 7.b. As can be seen in Fig. 7.a, the packet *a* gets the southern input channel of router 2 but fails to get the northern input channel of router 3. The proposed router can continue transferring the body flit and tail flit to the north output port by advancing the north port pointer *N*. So the whole packet can successfully be transferred to the router 2. After the southern input channel of router 2 is released, the packet *b* can get grant of the input channel and be transferred to router 2. In a similar way, the packet *b* is transfered to router 3 and the input VC is released, then packet *a* can get it. The deadlock is avoided.

V. Experiment

To evaluate the efficiency of the OPT and LXYROPT multicast routing algorithms, two other multicast routing algorithms were also implemented. These algorithms included XY tree-based multicast routing[12] and multiple unicast (MUC). We have developed cycle accurate virtual cut-through NoC simulator implemented in SystemC, which running under the window XP. The simulator calculates the average packet latency and power consumption for the packet transmission.The network parameters are shown in Table I.

For the performance metric, we define the packet latency as the number of cycles between the packet entering into the waiting queue and the time the packet ejection from network [19]. As a baseline, multiple unicast copies (MUC) is used to realize the multicast function. To compare these cases easily, we define the injection rate of MUC as equivalent injection rate as the tree-based multicast. For example, if a tree-based multicast

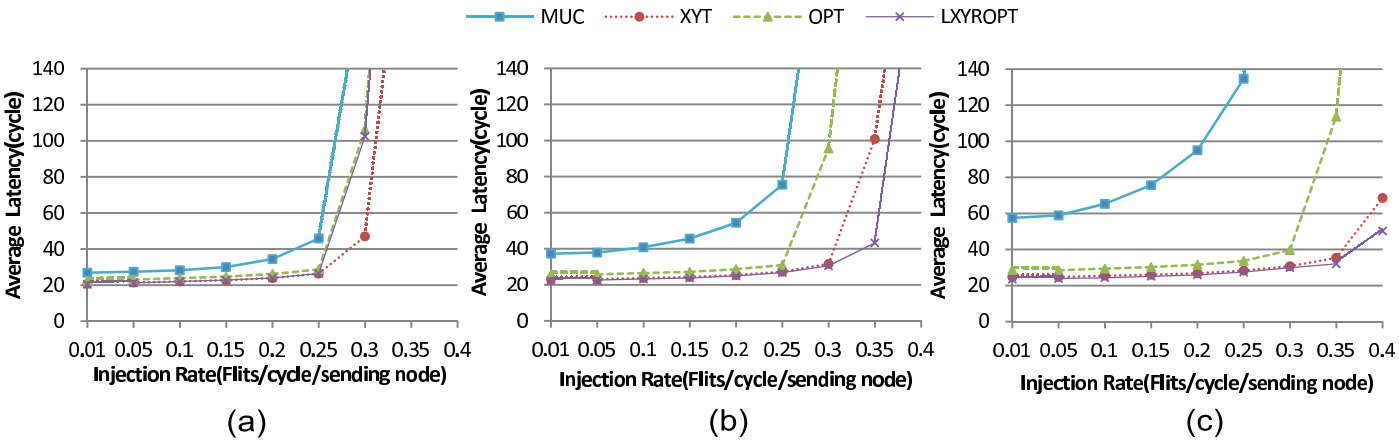

Fig. 8. Performance for the 5,10,20 nodes destination group under only multicast traffic.

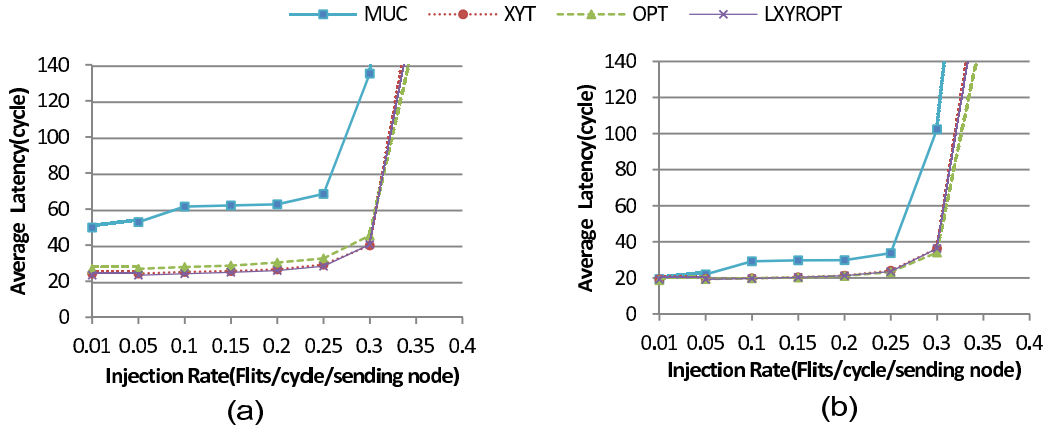

Fig. 9. Performance for mixed traffic.

TABLE I
NETWORK PARAMETERS

Topology	8 × 8 mesh
Routing	Multicast: Tag ID based Unicast: XY
Packet Size	1 flit: Multicast setup packet 3 flits: others
Virtual Channels	4
Buffers per Channel	3
Router ports	3 For corner 4 For border 5 For others
Number of MCT entries	16

group has 5 destinations and its injection rate is 0.1 flit/cycle, so the MUC needs generate 5 packets to the 5 destinations and its actual injection rate is 0.5 flit/cycles, but the results of MUC at this injection rate will be compared with tree-based multicast injection rate of 0.1 flit/cycle.

A. Multicast traffic profile

The first case of simulation is only multicast traffic without any other traffic. In these simulations, the PEs generate 3 flits packets and injects them into the waiting queue using the constant time intervals based on the injection rate. A uniform distribution was used to construct the destination set of each multicast packet. The source node is also selected randomly. We configure three scenarios: 16 source nodes, each node to 5 destination nodes; 8 source nodes, each node to 10 destination nodes; 4 source nodes, each node to 20 destination nodes.

Fig. 8 shows the average communication delay as a function of the average injection rate of sending node. For 5 node destination group(Fig. 8.a), The XYT and LXYROPT lead to the lowest latency among all the multicasting algorithms. OPT is about 10% more than the LXYROPT and XYT. The MUC is 30% more than the LXYROPT and XYT at low injection rate, when the injection rate improve to 0.25 flits/cycle/sending node, the latency of MUC is about 60% more than the XYT and LXYROPT. For 10 node destination group(Fig. 8.b), The LXYROPT lead to the best performance, while the XYT takes 2%, OPT takes 13% and MUC takes 67% more than it (MUC is in case of low injection rate). For 20 node destination group(Fig. 8.c), the LXYROPT is still best in performance, while the XYT takes 5%, OPT takes 20% and MUC takes 144% more than it. Fig. 8 reveals that in the case of MUC with larger size destination group, the network is more susceptible to being saturated. As a result, in such a system packets experience more latency.

B. Unicast and multicast (mixed) traffic profile

We also used a mixture of unicast and multicast traffic where multicast traffic accounted for 20%, which is similar to the scenarios of some cache coherence protocols. The unicast traffic is also uniformly distributed. Fig. 9 shows the average communication latencies against the packet injection rate. Fig. 9.a indicates that the LXYROPT is best in performance while the XYT takes 4%, OPT takes 15% and MUC takes 110%-140% more than the LXYROPT. Fig. 9.b shows the effect that the multicast made to the unicast traffic. Since OPT uses the minimum number of links, it outperforms the others.

C. Power Consumption

We calculate the power consumption by using the library of noxim [20]. The result is calculated and compared under the multicast traffic with different destination group size. We normalize the consumption of MUC as 1. The result is shown in Fig. 10. For 5 nodes group, XYT takes 70%, LXYROPT takes 67% and OPT takes %63 of MUC. For 10 nodes group, XYT takes 60%, LXYROPT takes 55% and OPT takes 50% of MUC. For 20 nodes group, XYT takes 49%, LXYROPT takes 45% and OPT takes 41% of MUC. For the best case, the OPT and LXYROPT gain 96% power consumption reduction compared to MUC and 69% compared to XYT. More power reduction is due to more efficiently using of links and reduction of buffer reading/writing operations.

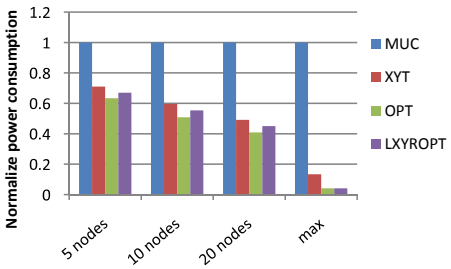

Fig. 10. Power consumption under only multicast traffic

VI. Summary and Conclusion

Two power efficient tree-based algorithms has been presented in this paper. A router which supports any tree-shaped construction is proposed to implement the algorithms. OPT tries to use the minimum number of links to construct the multicast tree. LXYROPT, on the premise of keeping the low latency, tries to minimize the links number of the multicast tree. These algorithms are deadlock-free because they use the west-first turn model to construct the path. The deadlock resulted from the multi-port requirement at a branch node is avoided by using the enough buffer to hold the whole packet and reserving the private buffer pointer for each output port in the input virtual channel.

For the future work, some path-based algorithms would be implemented on our router. Evaluating the performance and power consumption between the tree-based and path-based algorithm will be an interesting topic.

Acknowledgements

This work is supported by National Science Foundation of China under grant No.60970037 and No.60873212. We thank Abbas Eslami Kiasari from the Royal Institute of Technology in Sweden and Chaochao Feng from the National University of Defense Technology in China for their valuable comments to improve this paper.

References

[1] M.Millberg, E.Nilsson, R.Thid, A.Jantsch, "Guaranteed Bandwidth Using Looped Containers in Temporally Disjoint Networks within the Nostrum Network on Chip," *Proceedings of the conference on Design, automation and test in Europe*, vol. 2, pp. 890–895, 2004.

[2] M.B.Taylor, J.Kim, J.Miller, et.al. "The RAW microprocessor: A computational fabric for software circuits and general-purpose programs," *IEEE Micro*, vol. 22, no. 2, pp. 25–35, 2002

[3] K.Sankaralingam, R.Nagarajan, H.Liu, et.al. "Exploiting ILP, TLP, and DLP with the polymorphous TRIPS architecture," *Proceedings of the 30th annual international symposium on Computer architecture*, pp. 422–433, 2003

[4] P.Guerrier, A.Greiner, "A generic architecture for on-chip packet-switched interconnections," *Proceedings of the conference on Design, automation and test in Europe*, pp. 250-256, 2000.

[5] MP.Malumbres, J.Duato, "An efficient implementation of tree-based multicast routing for distributed shared-memory multiprocessors," *Journal of Systems Architecture*, vol. 46,no. 11, pp. 1019–1032, 2000

[6] C.Chiang, L.Ni, "Multi-address encoding for multicast," *Parallel Computer Routing and Communication*, vol. 853, pp. 146–160, 1994.

[7] R.Sivaram, D.K.Panda, C.B.Stunkel, "Efficient Broadcast and Multicast on Multistage Interconnnection Networks using Multiport Encoding," *8th IEEE Symposium on Parallel and Distributed Processing (SPDP '96)*, pp. 36–45, 1996.

[8] J.S.Turner, "An optimal non-blocking multicast virtual circuit switch," *IEEE INFOCOM*, vol. 94, pp. 298–305, 1994.

[9] E.Rijpkema, k.Goossens, A.Radulescu,J.Dielissen, et.al. "Trade-offs in the design of a router with both guaranteed and best-effort services for networks on chip," *Proceedings of the conference on Design, automation and test in Europe*, pp. 350–355, 2003.

[10] Z.Lu, B.Yin,A.Jantsch, "Connection-oriented Multicasting in Wormhole-switched Networks on Chip," *Proceedings of the IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures*, pp. 205–210, 2006.

[11] F.A.Samman, T.Hollstein, M.Glesner, "Multicast parallel pipeline router architecture for network-on-chip," *Proceedings of the conference on Design, automation and test in Europe*, pp. 1396–1401, 2008.

[12] N.E.Jerger, L.S.Peh, M.Lipasti, "Virtual Circuit Tree Multicasting: A case for On-chip hardware Multicast," *Int.Conf.Computer Architecture(ISCA)*, pp. 229–240, 2008.

[13] X.Lin, L.M.Ni, "Multicast communication in multicomputer network," *IEEE Trans, Parallel Distrib. Syst*, vol. 4, no. 10, pp. 1105–1117, 1993.

[14] R.V.Boppana, S.Chalasani, CS.Raghavendra, "Resource deadlocks and performance of wormhole multicast routing algorithms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 9, no. 6, pp. 535–549, 1998.

[15] M.Daneshtalab, M.Ebrahimi, S.Mohammadi, A.Afzali-Kusha, "Low-distance path-based multicast routing algorithm for network-on-chips," *IET Comput.Digit.Tech*, vol. 3, no.5, pp. 430–442, 2009.

[16] C.J.Glass, L.M.Ni "The turn model for adaptive routing," *Int.Conf.Computer Architecture(ISCA)*, pp. 278–287, 1992.

[17] D.R.Kumar, W.A.Najjar, P.K.Srimani, "A new adaptive hardware tree-based multicast routing in k-ary n-cubes," *IEEE Transactions on Computers*, vol. 50, no. 7, pp. 647–659, 2001.

[18] H.K.Young, S.Jeff, D.Jeff "Multicast routing with dynamic packet fragmentation," *Proceedings of the 19th ACM Great Lakes symposium on VLSI*, pp. 113-116, 2009.

[19] B.Towles, WJ.Dally, *Principles and Practices of Interconnection Networks,* Morgan Kaufmann, 2003.

[20] http://noxim.sourceforge.net/