2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing

Hierarchical Network Coding for Collective Communication on HPC Interconnects

Ahmed Shalaby, M. El-Sayed Ragab
Egypt-Japan University of Science and Technology (E-JUST)
P.O.Box 179, New Borg El-Arab City, Alexandria 21934, Egypt
{ahmed.shalaby, m.ragab}@ejust.edu.eg

Victor Goulart
Center for Japan-Egypt Cooperation in Science and Technology
3-8-33 Momochihama, Sawara-ku, Fukuoka 814-0001, Japan
goulart@ejust.kyushu-u.ac.jp

Ikki Fujiwara, Michihiro Koibuchi
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
{ikki, koibuchi}@nii.ac.jp

***Abstract*— Network bandwidth is a performance concern especially for collective communication because the bisection bandwidth of recent supercomputers is far less than their full bisection bandwidth. In this context we propose to exploit the use of a network coding technique to reduce the number of unicasts and the size of transferred data generated by latency-sensitive collective communication in supercomputers. Our proposed network coding scheme has a hierarchical multicasting structure with intra-group and inter-group unicasts. Quantitative analysis show that the aggregate path hop counts by our hierarchical network coding decrease as much as 94% when compared to conventional unicast-based multicasts. We validate these results by cycle-accurate network simulations. In 1,024-switch networks, the network reduces the execution time of collective communication as much as 64%. We also show that our hierarchical network coding is beneficial for any packet size.**

Keywords*— *collective communication; Interconnection networks; network coding; multicast algorithm; high-performance computing.

I. INTRODUCTION

As the scale of supercomputers including custom massively parallel computers and PC clusters increases, network bandwidth per flops (floating-point operations per second) becomes low. It will be more difficult for network bisection bandwidth to reach full-bisection bandwidth in future parallel computers. Recent interconnects, such as InfiniBand and Ethernet, usually use unicast-based multicasts, unlike the prior products, QsNET and QsNET II, which support hardware multicasts in a fat-tree topology of switches [2]. To implement collective communication, a large number of unicasts are simultaneously generated in such commodity interconnects. The unicasts may introduce a large number of packet contentions that are likely to lead to high latency in a multicast. Two approaches to mitigate this problem have received attention: (1) reducing the amount of communications by changing parallel algorithms of scientific applications that communication data in the kernels are between neighboring processes as much as possible [3], and (2) making the network utilization higher and emerging parallel applications, e.g., considering optimal deadlock-free routing and topology of switches. In this study, we use a network coding technique to reduce the number of unicasts and transfer data size in collective communication primarily for *k*-ary *n*-cubes.

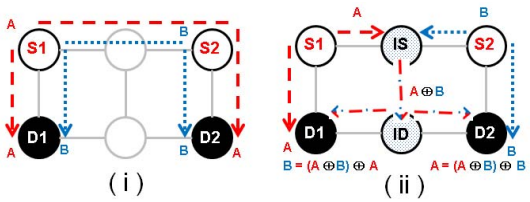

Fig. 1. (i) Unicast-based multicast vs. (ii) that with network coding.

Fig. 1 shows an example of collective communication in which two sources, *S1* and *S2*, multicast data *A* and *B* to destinations, *D1* and *D2*, in a 3×2 2-D mesh with dimension-order routing. Fig. 1(i) shows a conventional unicast-based multicast. The two shared links may cause packet contention. In the case of network coding (Fig. 1(ii)), each source sends a unicast to a single destination. Intermediate node (*IS*) makes a unicast by computing the XOR bit operation to two arrived unicast data. The shared link (*IS-ID*) is used once to send the encoded packet ($A \oplus B$). Once the encoded packet is received at destinations *D1* and *D2*, the original data is restored with the other packet by simply applying the XOR operation again, namely $A = (A \oplus B) \oplus B$ and $B = (A \oplus B) \oplus A$.

In this work we propose hierarchical network coding for collective communication. The hierarchical network coding consists of intra-group and inter-group multicasts that reduce the number of unicasts and the size of transferred data. Our findings of this paper are as follows:

- Through our quantitative analysis, the hierarchical network coding is beneficial as the network size becomes large and the number of multicast nodes increases.
- Through our cycle-accurate network simulation, the hierarchical network coding constantly obtains good performance gain in (1) all the transfer data (packet) sizes evaluated and (2) various overhead latencies to compute XOR data at intermediate nodes.

This paper is organized as follows. Section 2 describes related work. Section 3 describes our hierarchical network coding. Section 4 illustrates the network coding performance on path hop counts. Section 5 shows the results of the broadcast with hierarchical network coding by using a cycle-accurate network simulation. Section 6 draws conclusions of our findings and states our future work.

1066-6192/14 $31.00 © 2014 IEEE
DOI 10.1109/PDP.2014.58

II. Related Work

A. Multicast Communications

Hardware-, path-, and unicast-based algorithms are typical methods for multicasts in interconnection networks [4]. Hardware multicasts, e.g., QsNET II [2], duplicate packets at an intermediate switch for a multicast. Since it reduces the aggregate packet hop counts in a multicast, it efficiently sends data to multiple destinations. The typical implementation usually relies on a fat-tree structure. In addition, the packet header needs logic to control packet duplication at switches and multiple-destination tags. A path-based multicast sends data along a path that includes all destinations, and so requires an efficient multicast-path search, such as a Hamiltonian cycle. Theoretically, this is an interesting topic; however, current conventional interconnects do not always support hardware and also a path-based multicast [4][5].

A conventional way to support a multicast is to do a large number of unicasts. This is called a unicast-based multicast. In a simple unicast-based multicast, each source sends packets to all destinations. This paper refers to this as a "all-at-once" multicast. It is applicable for all multicasts occurring in parallel programming, including MPI_Alltoall, in which a source sends different data to destinations.

When a source scatters the same data to all destinations, a tree-based multicast is practical for reducing both the number of packet contentions and the aggregate packet hops [7]. In a tree-based multicast, first a source sends data to a single destination. Then these two nodes send data to four nodes. For d destinations, $\log_2(d+1)$ unicast steps are required. The effect of the tree-based multicast is evaluated in a high-performance computing (HPC) interconnect prototype [8]. In this work, our target is unicast-based multicasts.

B. Network Coding Applications

Network coding aims to optimize the data flow to improve network throughput and efficiency. Network coding is associated with information theory and was first introduced in 2000 [9]. Network coding has been applied in many fields, for example, distributed storage, wireless networks, file sharing, and multimedia streaming in peer-to-peer networks [11].

Unquestionably, these applications have different characteristics from those of supercomputer interconnects. Different characteristics affect the design of optimization. (1) Supercomputer interconnects usually have a non-random topology of switches and custom deadlock-free routing, e.g., dimension-order routing on *k*-ary *n*-cubes, and each cable usually has the same bandwidth. (2) Parallel applications explicitly generate multicasts at the program level, such as an MPI function. These unique features allow us to use the regularity of topologies and precisely estimate the number of packet hops for optimizing network coding. (3) Another unique feature concerns low-latency requirements, i.e. order of hundreds of nanoseconds. We thus consider the simple XOR bit operation for the encoding rather than other coding solutions in this paper.

To the best of our knowledge, no previous work has explored the use of network coding for efficiently use network bandwidth in supercomputer interconnects.

III. Hierarchical Network Coding

We propose to exploit the use of the network coding technique to the multicast communication scenario in supercomputers. Our proposed method has a hierarchical structure with intra-group and inter-group communications. In this section, we focus on a broadcast. However, we can naturally apply our hierarchical network coding for multicasts, or multiple sources to multiple destinations.

The detailed procedure for hierarchical network coding is as follows. Fig. 3 presents the pseudo-code.

a) Grouping: We divide a given network into a number of groups. In the example of the 2-ary 2-mesh in Fig. 2, the nodes are divided into two groups, depicted as shaded nodes and non-shaded nodes. The details of the grouping are quantitatively discussed in the next section.

b) Intra-group broadcasts: Every node inside a group exchanges transfer data by an existing multicast algorithm, e.g., a tree-based multicast. Every node then obtains all the data of the other nodes in the group. In the example in Fig. 2, data d_1 and d_2 are exchanged between two nodes in the shaded group, whereas data d_3 and d_4 are shared in the non-shaded group.

c) Network coding: We choose one of the nodes in each group as an intermediate node that computes the XOR function to encode packets. Assume that M nodes, $1,2,\dots,M$, have broadcast data, $d_1, d_2, \dots, d_M$, respectively. The intermediate node then generates $M-1$ encoded packets whose contents are $d_1 \oplus d_2, d_2 \oplus d_3, \dots, d_{M-1} \oplus d_M$.

d) Inter-group multicasts of encoded packets: The intermediate nodes exchange all the encoded packets by an inter-group multicast between all pairs of intermediate nodes. As in step (B), an existing multicast algorithm is used to deliver the packets. In the example, the encoded packets $(d_1 \oplus d_2)$ and $(d_3 \oplus d_4)$ are exchanged between the shaded and the non-shaded groups.

e) Intra-group broadcasts of encoded packets: The intermediate node delivers the encoded packets to all the nodes in its group. In the example, $(d_1 \oplus d_2)$ is sent to the other nodes in the non-shaded group, while $(d_3 \oplus d_4)$ is distributed in the shaded group.

f) Inter-group unicasts: Every node sends its data to all the other groups. One of the nodes in the destination group receives the data. In the example, node 1 sends d_1 to node 3 while node 3 sends d_3 to node 1. Similarly, d_2 is sent from node 2 to node 4 and d_4 is sent from node 4 to node 2.

g) Decoding: Every node receives (i) the data from all the nodes in the same group (step B), (ii) all the encoded packets from the other groups (step E), and (iii) the data from a node at each group (step F). Then it restores the non-received data of the group by computing the XOR bit operation. For example, a node obtains $d_1 \oplus d_2, d_2 \oplus d_3, \dots, d_{M-1} \oplus d_M$ from a group (step D) and also data d_1 (step F); then it restores data $d_2, d_3, \dots, d_M$ of the group by computing the XOR bit operation. Every node starts decoding packets as soon as the required data are obtained.

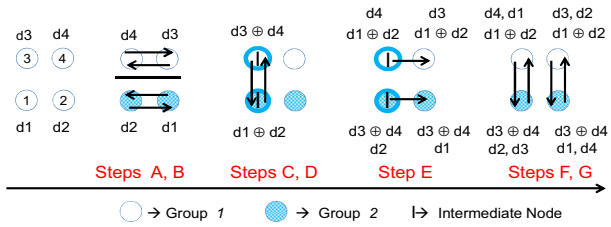

Fig. 2. Hierarchical network coding in 2-ary 2-mesh.

Algorithm: *hierarchical network coding*

N: **number of Nodes inside the network**
G: **number of Groups**
M: **number of Nodes inside each Group**
IN: **Intermediate Node**

Divide *N* to *G* Groups // Step A
for *i* = 1 to *G* {
  for *j* = 1 to *M* {
    $n_{i,j}$ broadcast its packet to (*M* -1) nodes inside each group *G* // Step B
    $n_{i,j}$ sends its packet to one node in each group *G* // Step F
    }
  }
Determine intermediate nodes *IN*
for *i* = 1 to *G* {
  IN_i encodes packets inside each group by computing XOR function // Step C
  IN_i broadcasts the encoded packets to (*G*-1) Intermediates nodes // Step D
  }
for *i* = 1 to *G* {
  IN_i broadcasts the encoded packets to (*M*-1) nodes inside each group // Step E
  }
for i = 1 to *G* {
  for j = 1 to *M* {
    $n_{i,j}$ decodes the received encoded packets by computing XOR function // Step G
    }
  }

Fig. 3. Pseudo-code of the hierarchical network coding.

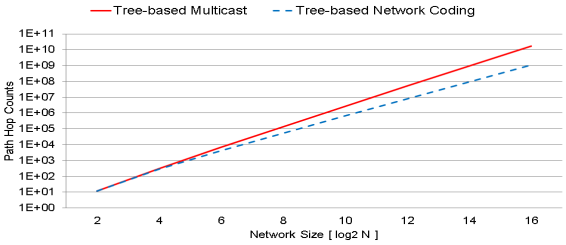

Fig. 4. Aggregate hop counts for tree-based multicast and that with hierarchical network coding on *k*-ary 2-mesh.

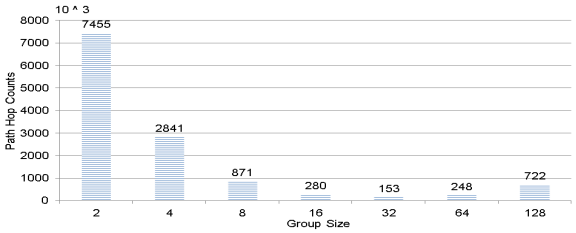

Fig. 5. Aggregate hop counts of hierarchical network coding for different group sizes in 16-ary 2-mesh.

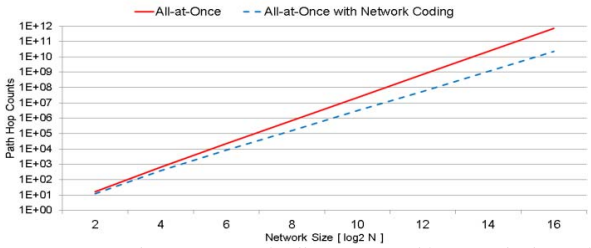

Fig. 6. Aggregate hop counts of all-at-once multicast and that with hierarchical network coding in *k*-ary 2-mesh.

IV. Quantitative Analysis

We quantitatively evaluate our hierarchical network coding when applied to *k*-ary *n*-cube topologies with minimal routing. In this evaluation, we highlight the parameters of the hierarchical network coding that benefit collective communication. We evaluate the impact of network size and the influence of group size on the aggregate path hop counts in a multicast. We show that our approach improves the multicast performance in both tree-based and all-at-once multicasts. Finally, we evaluate our approach on several network topologies.

A. Network Size

We evaluate the performance of the hierarchical network coding when applied to various *k*-ary *n*-mesh topologies. Fig. 4 plots the aggregate hop counts of unicasts of a tree-based multicast and that with the hierarchical network coding in various network sizes. The y-axis is logarithmic. Fig. 4 shows that the aggregate hop counts of unicasts drastically increase as the network size increases in both methods. However, at each network size, we observe the benefit of the hierarchical network coding, it reduces the aggregate path hop counts by as much as **94%** when compared to original tree-based multicast.

B. Group Size

An important concern of our hierarchical network coding is its group size, i.e., the number of nodes belonging to each group. The best group size minimizes the aggregate hop counts of unicasts. The coordinates of the intermediate nodes affect the total hop counts of unicasts when multicasting the encoded packets. We optimize the group size and the coordinates of the intermediate nodes to reduce the number of unicasts and their total hop counts in *k*-ary *n*-cubes. Fig. 5 shows the aggregate hop counts of hierarchical network coding in a 16-ary 2-mesh with $N = 256$ nodes. We varied the group size from 2 to 128. The best group size is 32 nodes per group. We use the best group size from quantitative analysis in the rest of this paper.

C. Multicast Algorithm

Since the hierarchical network coding uses a multicast algorithm for data exchange, we evaluate the influence of the multicast algorithm on its performance by the comparison of tree-based and all-at-once multicasts.Fig. 6 plots aggregate hop counts of an all-at-once multicast and those with the hierarchical network coding in various network sizes. Similar to the case of the tree-based multicast, we can observe that the benefit of the hierarchical network coding increases as the network size increases. For example, the hop counts required by network coding for this all-to-all scenario in the 256-ary 2-mesh is **32 times** less than that for the all-at-once multicast.

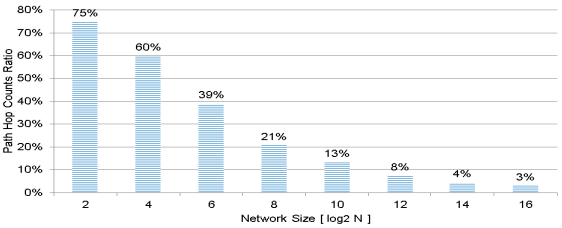

Fig. 7. Aggregate hop count ratios for all-at-once hierarchical network coding against all-at-once multicast on *k*-ary 2-torus.

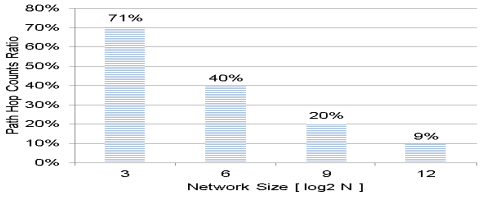

Fig. 8. Aggregate hop count ratios of all-at-once hierarchical network coding against all-at-once multicast on *k*-ary 3-mesh.

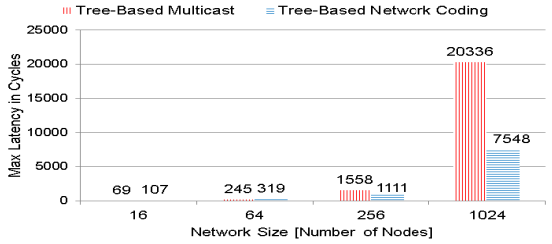

Fig. 9. Execution time for tree-based multicast and that with hierarchical network coding on *k*-ary 2-mesh.

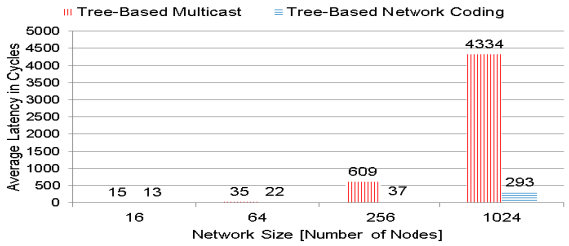

Fig. 10. Average packet latency for tree-based multicast and that with hierarchical network coding on *k*-ary 2-mesh.

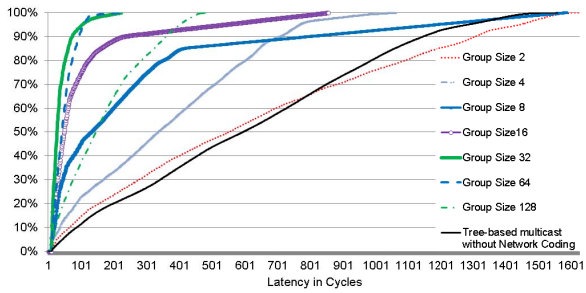

Fig. 11. Cumulative distribution function of packet latency in cycles for tree-based multicast and that with hierarchical network coding in 16-ary 2-mesh.

D. Topology

We evaluate the performance of the hierarchical network coding when applied to different *k*-ary *n*-cube topologies. The main difference between a mesh and a torus in our hierarchical network coding is the path hop counts in the inter-group unicast step, because each unicast uses the wraparound links to reduce its path hop counts. Fig. 7 shows the hop ratios between all-at-once multicasts and those with the hierarchical network coding in the *k*-ary 2-tori. Fig. 8 shows the hop ratio between the all-at-once multicast and that with the hierarchical network coding in *k*-ary 3-meshes. Similar to the case for the *k*-ary 2-meshes, we observe that a similar good gain is obtained. We consequently consider that the hierarchical network coding is efficient to reduce the aggregate path hop counts in k-ary n-cubes, especially for the large network size.

V. CYCLE-ACCURATE SIMULATION

We evaluate the performance of the hierarchical network coding more precisely by using the cycle- accurate network simulator called BookSim [12]. We implement tree-based and all-at-once multicast communication scenarios with dimension-order routing on k-ary n-cube topologies. The number of virtual channels is set to four. A header flit requires at least three clock cycles to be transferred to the next router or host. Virtual cut-through switching is used as the switching technique on each router. We set the packet length for one flit as a default. The default overhead to compute XOR at an intermediate node is set to one cycle. We evaluate the execution cycles (maximum end-to-end latency) of all-to-all broadcast in tree-based multicast. By contrast, the execution time of a broadcast is the sum of the maximum latency of all sequential steps.

A. Network Size

Fig. 9 plots the execution time of all-to-all broadcasts for the tree-based multicast and the same with the hierarchical network coding. The y-axis represents the simulation cycles; thus, lower values are better. Fig.10 illustrates the average packet latency in which each host received all the data. The performance tendency is consistent with quantitative analysis. The hierarchical network coding speeds up the all-to-all broadcast communications by **three times**. It also improves the average packet latency by **14 times** in the 1,024-switch network. Another finding is that the hierarchical network coding achieves better performance in large network sizes. Since the tree-based algorithm adds delay to multicast communications due to synchronization. The sum of the two overheads (hierarchical network coding and tree-based synchronizations) dominates the execution time in small network sizes. Thus, both methods have similar performance in a small network. In contrast, as the network size becomes larger, the synchronization delay by the tree-based multicast (without network coding) strongly affects the execution time, and thus the hierarchical network coding improves the performance drastically.

B. Group Size

To show the impact of group size on the performance, we implemented all possible sizes of groups in a 16-ary 2-mesh. Fig. 11 illustrates the CDF of the latency of all packets. We can observe that the grouping affects not only the execution time but also the total latency of all packets. We can also observe that all packets of the 32-group case have latency less than 200 cycles and approximately 80% of the packets have less than 100 cycles, whereas the 128-group case is worse than the tree-based multicast without the hierarchical network coding. Approximately 80% of the 128-group packets have more than 100 cycle latency.

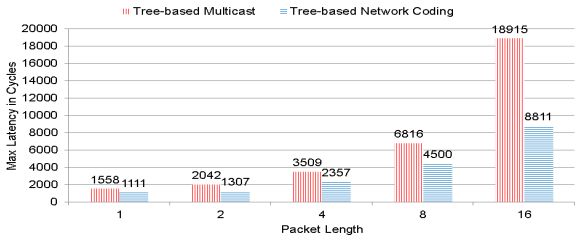

Fig. 12. Execution time for various packet lengths (1, 2, 4, 8, 16 flits) for tree-based multicast and that with hierarchical network coding in 16-ary 2-mesh.

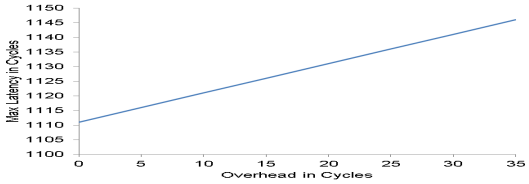

Fig. 13. Execution time for various latency overhead at intermediate nodes in 16-ary 2-mesh.

C. Packet Length

Generally, long packets increase the possibility of incurring packet contentions under a heavy traffic load that may seriously degrade the performance in the hierarchical network coding. We investigate the performance under various packet lengths. The hierarchical network coding with the configuration of the best grouping (32 groups) in the 16-ary 2-mesh with different packet lengths (1, 2, 4, 8, 16 flits) was evaluated. Fig.12 shows the execution time for each packet length when compared to the original tree-based multicast. The hierarchical network coding always improve the execution time. As the packet length increases, it is more beneficial, up to 54% for the 16-flit packet transfer.

D. Latency Overhead in Network Coding

We evaluate the hierarchical network coding with different overhead latencies to compute the XOR bit operation at the intermediate nodes. Fig.13 shows the execution time, including the overhead for computing network coding at the intermediate node in a 16-ary 2-mesh. The X-axis represents the overhead clock cycles. Surprisingly, the overhead only marginally affects the end-to-end latency. It is not a bottleneck for the collective communication.

E. Message Combining vs. Network Coding

In message combining scenario: in each group, intermediate node combines incoming multiple packets into one message, then sends the combined message to other intermediate nodes. While in hierarchical network coding scenario: in each group, intermediate node generates a packet by computing XOR function for each two incoming packets. Thus, hierarchical network coding generates larger number of packets. However, the size of combined packet is the same size of incoming packets. While in message combining, the size of combined message is the sum of packets to be combined. Furthermore, in message combining scenario, intermediate nodes should wait to receive all packets to start combining operation. Obviously, the hierarchical network coding is expected to reduce total end-to-end latency when using small packet size. In order to compare between the two scenarios, we implemented them on network size 16 and group size 32, best group size. The hierarchical network coding end-to-end latency is 1276 cycles while it is 1832 cycles for message combining. These results confirm our expectation.

VI. Conclusions

In this work we proposed to exploit the use of network coding for relaxing the relatively low network bandwidth problem in collective communication in HPC off-chip interconnects. Our network coding is a hierarchical multicast structure with intra-group and inter-group unicasts. Since it reduces both the number of unicasts and the transfer data size, good performance was obtained in various combinations of the (unicast-based) multicast algorithm, the topology, and the transfer data size when the proper group size is set.

Quantitative analysis results show that the hierarchical network coding is beneficial as network size becomes large: a 94% improvement is obtained in a 4,096-switch network with a conventional tree-based multicast. Cycle-accurate network simulation results validate the quantitative analysis results in various topologies, multicast algorithms, packet sizes and overhead latencies to compute the XOR bit operation in intermediate nodes. Our network coding improves the execution time of collective communication by up to 64% in the 32-ary 2-mesh. Our future work will attempt to analyze the case for complex encoding computation so that more than two packets are aggregated to the resulting encoded packet. Since this may further reduce the number of unicasts in a multicast.

Acknowledgments

This work was partially supported by Strategic Information and Communications R&D Promotion Program, Ministry of Public Management, Home Affairs, Posts and Telecommunications. Japan.

References

[1] D.E Atkins, et al. 2003. Evolutionizing Science and Engineering Through Cyberinfrastructure. Report of the National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure.

[2] F Petrini, S Coll, E Frachtenberg and A Hoisie. 2001. Hardware- and Software-Based Collective Communication on the Quadrics Network. IEEE International Symposium on Network Computing and Applications.

[3] K. Asanovic, et al. , 2006, The Landscape of Parallel Computing Research: A View from Berkeley, EECS Department, University of California, Berkeley. UCB/EECS-2006–183.

[4] J Duato, S Yalamanchili and L Ni. 2002. Interconnection Networks: an engineering approach. Morgan Kaufmann.

[5] InfiniBand Trade Association. http://www.infinibandta.org.

[6] Myricom.http://www.myricom.com/scs/myrinet/m3switch/guide/myrinet-2000_switch_guide.pdf

[7] R Kesavan and D Panda. 2001. Efficient Multicast on Irregular Switch-Based Cut-Through Networks with Up-Down Routing. IEEE Transactions on Parallel and Distributed Systems (vol.12,no.8),808–828.

[8] M Koibuchi, K Watanabe, T Otsuka and H Amano. 2005. Performance Evaluation of Deterministic Routings, Multicasts, and Topologies on RHiNET-2 Cluster. IEEE Trans. Parallel Distrib. Syst. (vol. 16, no. 8), 747-759.

[9] R Ahlswede, N Cai, S Li, and R Yeung. 2000. Network Information Flow. IEEE Transactions on Information Theory, (vol. 46), 1204-1216.

[10] M Wang and B Li. 2007. Random Push with Random Network coding in Live Peer-to-peer Streaming. IEEE Journal on Selected Areas in Communications (vol. 25, no. 9), 1655–1666.

[11] M Medard and A Sprintson. 2011. Network Coding: Fundamentals and Applications. Elsevier Science & Technology Publisher.

[12] https://nocs.stanford.edu/cgi-bin/trac.cgi/wiki/Resources/BookSim