

Binomial Graph: A Scalable and Fault-Tolerant Logical Network Topology

Thara Angskun¹, George Bosilca¹, and Jack Dongarra²

¹ Department of Computer Science, The University of Tennessee, Knoxville

² University of Tennessee, Oak Ridge National Lab. and University of Manchester

Abstract. The number of processors embedded in high performance computing platforms is growing daily to solve larger and more complex problems. The logical network topologies must also support the high degree of scalability in dynamic environments. This paper presents a scalable and fault tolerant topology called binomial graph (BMG). BMG provides desirable topological properties in terms of both scalability and fault-tolerance for high performance computing such as reasonable degree, regular graph, low diameter, symmetric graph, low cost factor, low message traffic density, optimal connectivity, low fault-diameter and strongly resilient. Several fault-tolerant routing algorithms are provided on BMG for various message types. More importantly, BMG is able to deliver broadcast messages from any node within $\log_2(n)$ steps.

1 Introduction

Recently, several high performance computing platforms have been installed with more than 10,000 CPUs, such as Blue-Gene/L at LLNL, BGW at IBM and Columbia at NASA [1]. However, as the number of components increases, so does the probability of failure. To satisfy the requirements of such a dynamic environment (where the available number of resources is fluctuating), a scalable and fault-tolerant communication framework is needed. The communication framework is important for both runtime environments of MPI libraries and the MPI libraries themselves. The communication frameworks are based on a logical topology. The design of logical topologies should meet the following important criteria: (a) Low degree - the degree of a node is the number of links incident to the node. The degree number should be kept to a minimum to reduce the state management load on each node. (b) Regular graph - every node has the same degree. (c) Low diameter - the diameter of a graph is the longest shortest path between any two nodes. This number represents the worst case performance (a maximum number of hops) of the topology. (d) Symmetric graph - the average inter-nodal distance should be the same from any source node. (e) No numbers of node restriction - unlike the physical network topology, the logical network topology should be able to handle any number of nodes to support large scale parallel applications. In addition to the mentioned criteria, the routing protocol of the topology should be able to deliver the control messages in both normal and failure circumstances.

There are several existing logical network topologies that can be used in high performance computing (HPC). The fully connected topology is good in terms of fault-tolerance, but it is not scalable because of high degree. The bidirectional ring topology is more scalable, but it is not fault-tolerant. Hypercube [2] and its variants FPCN [3], de Bruijn [4] and its variants, Kautz [5] and ShuffleNet [6] have a number of node restrictions. They are either not scalable or not fault-tolerant. The Manhattan Street Network (2D Torus) [7] is more flexible (no restriction in numbers of node) than Hypercube-like topologies. However, it has a much higher average hop-distance. Variants of k -ary tree, such as Hierarchical Clique (HiC) [8] and k -ary sibling tree (Hypertree [9]) used in SFTP [10, 11], are scalable and fault-tolerant. They are good for both unicast and broadcast messages. However, all nodes in their topologies are not equal (the resulting graph is not regular). Topologies, used in structured peer-to-peer networking based on distributed hash tables such as CAN [12], Chord [13], Pastry [14] and Tapestry [15], are also scalable and fault-tolerant. They were designed for resource discovery in dynamic environment. Hence, they may not be efficiently used in HPC owing to the overhead for managing highly dynamic applications.

This paper presents a new scalable and fault tolerant logical network topology called binomial graph (BMG). BMG provides desirable topological properties in terms of both scalability and fault-tolerance for high performance computing. The structure of this paper is as follows. Section 2 describes the structure and initialization scheme of BMG. Some topological properties of BMG are discussed in section 3 and section 4. Section 5 presents the fault-tolerant routing algorithm, followed by conclusions and future work in the section 6.

2 Structure of BMG

BMG is an undirected graph $G:=(V,E)$ where V is a set of nodes (vertices); $|v| = n$ and E is a set of links (edges). Each node i , where $i \in V$ and $i=1,2,\dots,n$, has links to a set of nodes U , where $U=\{i \pm 1, i \pm 2, \dots, \pm 2^k | 2^k \leq n\}$ in circular space, i.e., node i has links to a set of clockwise (CW) nodes $\{(i+1) \bmod n, (i+2) \bmod n, \dots, (i+2^k) \bmod n | 2^k \leq n\}$ and a set of counterclockwise (CCW) nodes $\{(n-i+1) \bmod n, (n-i+2) \bmod n, \dots, (n-i+2^k) \bmod n | 2^k \leq n\}$. The structure of BMG can also be classified in the circulant graph family³. A circulant graph with n nodes and jumps j_1, j_2, \dots, j_m is a graph in which each node i , $0 \leq i \leq n-1$, is adjacent to all the vertices $i \pm j_k \bmod n$, where $1 \leq k \leq m$. BMG is a circulant graph where j_k is the power of 2 that is less than or equal n .

Fig. 1(a) illustrates an example of 12-node binomial graph. All the lines represent all connections in the network. The other way to look at the binomial graph is that it is a topology, which is constructed from merging all necessary links being able to create binomial trees from each node in the graph. Fig. 1(b) shows an example of a binomial tree when node 1 is the root node. The arrows point in the direction of the leaf nodes. For each link in a binomial graph, a

³ Although BMG can be categorized into a class of a well-known graph, this particular configuration have never been analyzed.

node which has a higher ID will initiate the connection to the node that has the lower ID. Assume that the nodes are started in order, i.e., node 1 is started before node 2, node 2 is started before node 3 and so on. When node i starts up, it will try to establish a connection to each $u \in U$, where $u < i$. The dashed lines of Fig. 1(c) represent all the connections that are linked to node 9 in the 12-node binomial graph. The arrows point from the initiators (connectors) to the acceptors. Node 9 initiates connections to node 1, 5, 7 and 8, while it accepts connections from node 10 and 11. Their structured initialization procedure is capable of handling any number of nodes in a scalable manner, i.e, there is no number of node restriction for BMG.

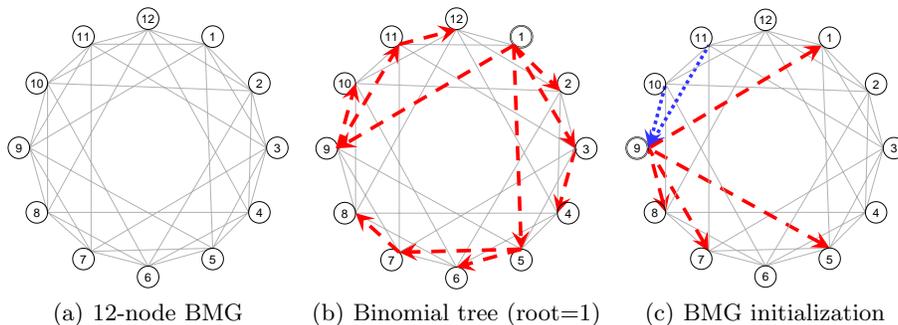


Fig. 1. Binomial graph structure

3 Properties of BMG

This section presents several basic properties in graph theory of BMG such as degree, diameter, average distance, message traffic density and cost factor. Several evaluations comparing BMG with other related topologies are presented. Properties of related topologies were obtained from respective theorems given in the reference papers.

3.1 Degree

If a node in a graph has δ connections that are linked to it, the node has degree δ . The minimum degree δ_{min} of a graph is the smallest node degree, while the maximum degree δ_{max} of a graph is the largest node degree. If every node has the same degree ($\delta_{min} = \delta_{max}$), the graph is *regular*. A *regular* graph also means that all the nodes are equivalent and can use the same routing and fault-handling algorithms. Those algorithms are also simpler compared to non-regular topologies, such as tree-based topologies, because they don't need to handle special cases. BMG is a regular graph. Fig. 1(a) is an example of a 12-node BMG, where each node has degree 6.

For a BMG of size n (having n nodes), each node has a degree δ as shown in Equation (1).

$$\delta = \begin{cases} (2 \times \lceil \log_2 n \rceil) - 1 & \text{For } n = 2^k, \text{ where } k \in \mathbb{N} \\ (2 \times \lceil \log_2 n \rceil) - 2 & \text{For } n = 2^k + 2^j, \text{ where } k, j \in \mathbb{N} \wedge k \neq j \\ 2 \times \lceil \log_2 n \rceil & \text{Otherwise} \end{cases} \quad (1)$$

Although BMG has higher degree than other related topologies, it has lower distance than the related properties. The trade-off between degree and distance for a fair comparison between topologies is discussed later in section 3.5.

3.2 Diameter

The distance $d(i,j)$ between a node i and a node j in a graph is defined as the length of the shortest path from i to j in the graph. The diameter D of a graph is given by $\max(d(i,j))$ over all possible pairs (i,j) of nodes in the graph. The diameter D is the longest shortest path between any two nodes in the graph. The BMG diameter, along with the diameter of related networks such as 2D Torus, binary Hypercube, Chord, $HiC_{(4,h)}$ and 4-ary Hypertree, is shown in Fig. 2(a). BMG has the lowest diameter among them.

3.3 Average distance

The average distance \bar{d}_i of node i is obtained from Equation (2).

$$\bar{d}_i = \frac{\sum_{j=1}^n d(i,j)}{n-1}, \text{ where } i \neq j \quad (2)$$

A graph is symmetric (in the sense of average distance) if all the nodes have the same average inter-nodal distance (\bar{d}_i , where $i \in \mathbb{N}$), which is also the same as average distance of the graph \bar{d} . The average distance \bar{d} of a graph is given by Equation (3).

$$\bar{d} = \frac{\sum_{i=1}^n \sum_{j=1}^n d(i,j)}{n \times (n-1)}, \text{ where } i \neq j \quad (3)$$

Fig. 2(b) illustrates that BMG has the lowest average distance (of a graph) when comparing with 2D Torus, binary Hypercube, Chord, $HiC_{(4,h)}$ and 4-ary Hypertree topologies. The average distance also reflects the unicast performance (in terms of hop).

If node i uses S_i steps to send broadcast messages, the average number of broadcast steps of a graph is computed by $\frac{\sum_{i=1}^n S_i}{n}$. Fig. 2(c) illustrates the average number of broadcast steps of BMG. The number of broadcast steps from any node in BMG is $\log_2(n)$ because a binomial tree can always be created from any node of BMG.

BMG is also a symmetric graph in terms of performance. Both the average inter-nodal distance (unicast) and the number of steps required for broadcast are the same for all nodes.

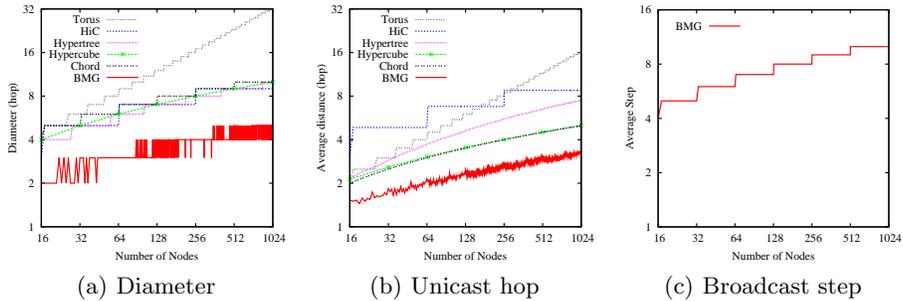


Fig. 2. BMG distance related properties

3.4 Message Traffic Density

Assuming each node is sending one message to a node at average distance \bar{d} , the message traffic in the network can be estimated by the message traffic density (ρ). This factor is given by $\rho = \frac{\bar{d} \times n}{L}$, where L is the total number of links. BMG has the number of links $L = \frac{\delta \times n}{2}$. Fig. 3(a) shows that BMG has the lowest message traffic density when comparing with 2D Torus, binary Hypercube, Chord, $HiC_{(4,h)}$ and 4-ary Hypertree topologies. Nevertheless, ρ of BMG does not exceed 0.5 even for large systems.

3.5 Cost Factor

A network with large diameter usually has a small degree, but it may suffer from high latency, e.g., the ring topology has $\delta = 2$ and $D = \lfloor \frac{n}{2} \rfloor$. On the other hand, a small-diameter network may have a higher degree, e.g., the fully connected topology (complete graph) has $D = 1$ and $\delta = n - 1$. It is desirable to have a logical topology with both small diameter (low latency) and small degree (low state management). Thus, for a symmetric network, a cost factor [16] (ξ) is defined as the product of diameter (D) and degree (δ). In general, the cost of a particular topology [17] can also be defined as the product of the diameter (D) and the number of links (L) in the network.

Fig. 3(b) and Fig. 3(c) present a cost factor comparison. Although 4-ary Hypertree has lower cost factor than BMG, it has significantly higher message traffic density and average distance. The results show that BMG has comparable cost factor to binary Hypercube and Torus. It also has significantly lower cost factor than $HiC_{(4,h)}$, Chord, ring and fully connected topologies.

All of the properties presented in this section illustrate the scalable capability of BMG. It has a reasonable degree, a low diameter, a low average (unicast and broadcast) distance, a low message traffic density and a low cost factor. The capabilities of BMG in terms of fault-tolerance are presented in the next section.

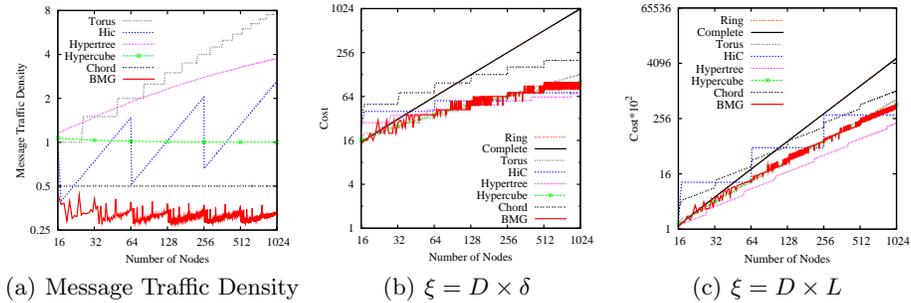


Fig. 3. ρ and ξ properties of BMG

4 Fault Tolerant Capabilities of BMG

This section presents several properties in terms of fault-tolerance to illustrate the fault-tolerant capability of BMG such as connectivity and fault-diameter.

4.1 Connectivity

The node-connectivity (κ) of BMG is defined as the minimum number of nodes of which removal can result in disconnecting the network, i.e., the smallest number of node-distinct paths between any two nodes. The link-connectivity (λ) of BMG is defined as the minimum number of link of which removal can result in disconnecting the network, i.e., the smallest number of link-distinct paths between any two nodes. Fig. 4(a) illustrates that node-connectivity and link-connectivity of BMG are equal to the degree, i.e., the BMG is $\delta - 1$ node fault-tolerant and $\delta - 1$ link fault-tolerant. However, some cases, where δ nodes fail, will result in disconnecting the network. In fact, the percentage for n -node BMG being disconnected by a δ -node fault set is minimal (less than one percent) as shown in Fig. 4(b). Fig. 4(c) compares the node-connectivity and link-connectivity of $HiC_{(4,h)}$ and BMG at the same degree. BMG is twice as robust as $HiC_{(4,h)}$.

For any graph, $\kappa \leq \lambda \leq \delta_{min}$. However if $\kappa = \lambda = \delta_{min}$, the graph is *optimally connected* [18] because the node and link connectivities are as high as possible, i.e., the network is as robust as it could be, and that is the case for BMG. The optimal connectivity is also important in reducing the impact of node destruction on link load [19]. In general, traffic can be distributed over at least λ link-disjoint paths between two nodes. If $\kappa < \lambda$, the number of link-disjoint paths may drastically drop after the loss of a critical node. However, if $\kappa = \lambda$ (optimal connectivity), the node failure can destroy, at most, one of the λ link-disjoint paths.

4.2 Fault diameter

The fault diameter F is the largest diameter of the network when there are $\lambda - 1$ node failures (a maximum number of failure nodes before the network becomes

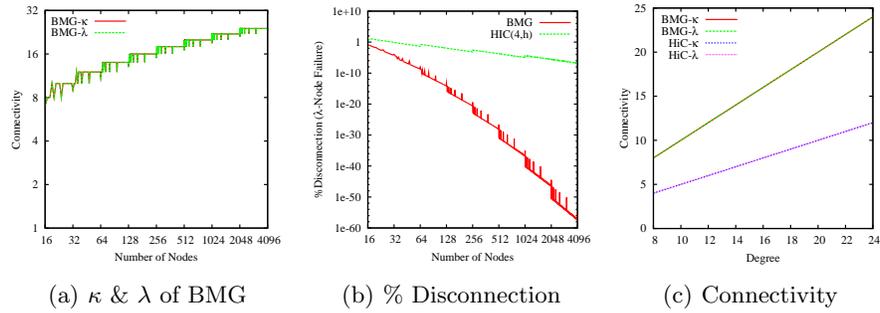


Fig. 4. Connectivity

bipartite). Fig. 5(a) depicts that the fault diameter of BMG is lower than other topologies. There are two classes of graph G distinguished by the relationship between the fault diameter (F) and the diameter (D) of the graph [20], called *strongly resilient* and *weakly resilient*. A graph is considered *strongly resilient* if there exists a constant ϕ such that $F(G) \leq D(G) + \phi$ for all graph sizes n , where $n \in \mathbb{N}$. On the other hand, a graph is considered *weakly resilient* if there exists a constant ϕ such that $F(G) \leq D(G) \times \phi$ for all graph sizes n , where $n \in \mathbb{N}$. BMG is considered *strongly resilient* as shown in Fig. 5(b), where $\phi = 2$, i.e., $F(BMG) \leq D(BMG) + 2$. This indicates that, even under faulty conditions, the performance of BMG will not be severely degraded.

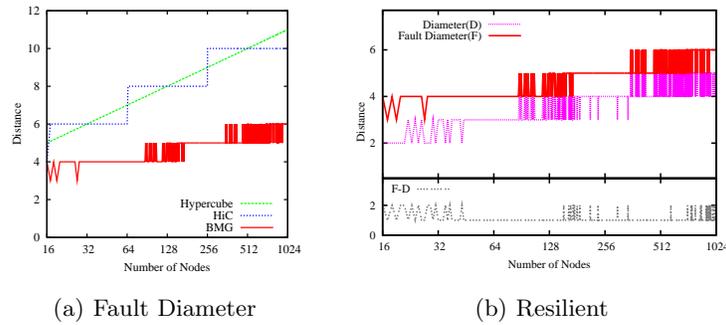


Fig. 5. Fault diameter

5 Fault-tolerant Routing

This section presents several routing algorithms that can handle different message types (unicast, multicast and broadcast) under both normal and failure circumstances. Due to the fact that all nodes in BMG are equal (both regular

and symmetric), each node in the graph can run the same routing algorithm. In BMG, each node only needs to maintain the contact information, node ID and status of its δ direct neighbors. This information will be used for searching the shortest path from a source to a destination in both normal and failure modes.

5.1 Unicast Routing Algorithm

Under normal circumstances (no failures), the unicast message routing could be optimally done by searching for the shortest path between the source and destination. However, the complexity of the algorithm is more than sub-optimal, which could estimate the shortest path with rule-based techniques.

Optimal Algorithm The optimal routing algorithm can use the breadth-first search technique with a modified graph coloring algorithm. Although this algorithm gives the optimal result, the complexity of the algorithm is $O(|V| + |E|)$. A routing table could be used to keep the result of the neighbors sorted by the shortest path from the node itself to all other nodes in BMG. The lower priority neighbor node will be used as an alternative for the next hop when the highest priority neighbor dies. In practice, the routing table can be implemented with a data structure that has the search complexity $O(\log_2 n)$ such as a red-black tree.

Sub-Optimal Algorithm A basic algorithm to estimate the shortest path between nodes is to use a rule-based method that sends the unicast messages to the neighbor that has the closest ID to the destination ID as shown in Algorithm 1. The complexity of the basic unicast routing algorithm is $O(\delta)$.

Algorithm 1 Find neighborID which has the shortest distance to destID

Require: $1 \leq myID \leq n \wedge 1 \leq destID \leq n, n \in \mathbb{N}$

```

1: Min  $\leftarrow \infty$ 
2: Get neighborID of myID
3: for  $i = 0$  to  $(Numbersofneighbor) - 1$  do
4:   Distance  $\leftarrow |destID - neighborID[i]|$ 
5:   if Distance < Min then
6:     Min  $\leftarrow$  Distance
7:     nextHopID=neighborID[i]
8:   end if
9: end for
10: Return nextHopID

```

The basic routing algorithm becomes sub-optimal when the destination is one of the neighbors (or a neighbor of neighbors) of a node (including a source node) of which ID is not the closest ID compared to the destination, e.g., if the node 1 is the source and the node 12 is the destination of unicast messages in

a 23-node configuration. The basic routing algorithm will route messages from node 1 to node 12 using node 9 and node 11 as the intermediate nodes, while the shortest path would use node 5 as the intermediate node to route unicast messages from node 1 to node 12.

To overcome this shortcoming, a unicast message routing algorithm (called variant) has been introduced. It is the variant of the basic algorithm, which allows messages forward to a neighbor which its ID is not the closest ID to the destination ID if the destination is directly connected to the neighbor, e.g., in case of sending unicast messages from node 1 to node 12 on 23-node BMG. Node 1 will forward messages to neighbor node 5 (even node 9 has a closer ID to node 12), because node 12 is directly connected to node 5. Fig. 6(a) illustrates the average (\bar{d}) and maximum (d_{max}) distance of the variant routing algorithm compared with the basic and optimal routing algorithm. Fig. 6(b) and Fig. 6(c) depict that the variant algorithm is marginally better than the basic algorithm in terms of d_{max} and \bar{d} .

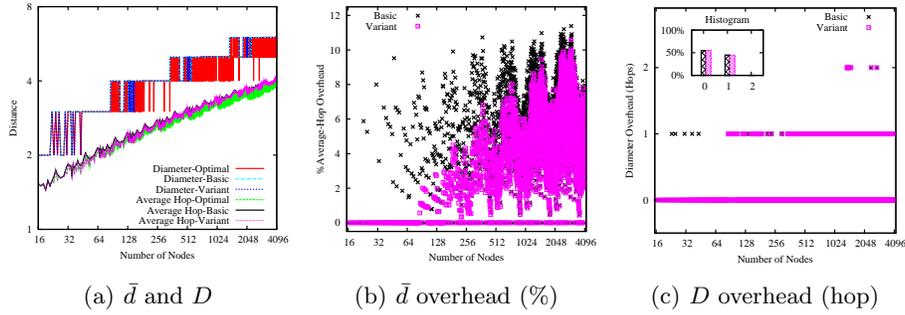


Fig. 6. Variant routing performance

The average d_{max} overhead of a configuration between 16 and 4096 nodes is reduced from 0.454 to 0.449 hops, while the average \bar{d} overhead is also decreased from 5.55% to 4.69%. The complexity of the variant unicast routing algorithm is $O(\delta^2)$. This algorithm could be extended by recursively finding if the destination is directly connected to the neighbor of a neighbor and so on. The complexity of the extended version is $O(\delta^{r+1})$, where r is the recursive level.

In normal circumstances, each of the unicast routing algorithms is loop avoidance, i.e., it prevents problems of duplicate packet, broadcast storm and deadlock. The messages will always get closer to the destination at every hop. However, this is not the case under failure circumstances. The messages can be looped. Thus, additional loop prevention is needed. The loop can be prevented by techniques such as adding transit list fields in the packet header. The transit list fields may contain 3-tuple consisting of incoming neighbor, transit node and outgoing neighbor. These fields do not only prevent a node from sending a packet to the same neighbor twice, but also help in backtracking to the originator of

the packet and finding an alternative route. The transit list may be added to the header only when a node detects that its neighbor, which is in the outgoing direction of the packet, died or when the packet has already had the transit list.

5.2 Multicast Routing Algorithm

The multicast from any nodes in BMG is the capability to send messages to several destinations (1 to m , where $m < n$). Unlike the IP multicast, multicast group management (group creation and termination) is not required. The multicast group members are embedded in the message header. The multicast routing algorithms are all based on unicast algorithms. All capabilities in both normal and failure circumstances are the same, except that the multicast messages can also be split at an intermediate node. The messages will be split, if the shortest paths to those destination nodes are not in the same direction from the intermediate node point of view. However, if there is more than one shortest path to a destination, the intermediate node will choose the next hop which can go along with other destinations. When a node receives a multicast message, it will first determine the header and choose the next hop for each multicast destination according to the shortest path to them. The node will recreate the header corresponding to the direction of each next hop. Messages that contain the largest number of hops will be forwarded first to increase network throughput by utilizing multiple links simultaneously.

5.3 Broadcast Routing Algorithm

Broadcast messages from any node in BMG are handled by creating a binomial spanning tree from the source. Under normal circumstances, only the links of the binomial spanning tree of BMG are used to prevent loop (broadcast storm problems). There are two steps involving the next hop calculation. The first step is to create a binomial spanning tree using the source node as the root node of the tree. The second step is to calculate the next hop. The next hop is chosen from the children of each node according to the spanning tree that has the highest cost among its children. The cost is computed from the number of steps used to send a message to all nodes in the subtrees of children.

There are two algorithms for creating a binomial spanning tree and finding a parent and children of a node, according to the binomial spanning tree from the source. The first algorithm creates the tree by choosing the closest ID first as shown in Fig. 7(a), while the second algorithm chooses the closest ID last as shown in Fig. 7(b)

Both of the algorithms only use the existing links in BMG to create the binomial tree from any node in BMG. This means that broadcast messages from any node in BMG can always be delivered within $O(\log_2 n)$ steps.

In case of failure, a broadcast message is encapsulated into a multicast message, and then the message is sent from a parent of the failed node to its children in the binomial spanning tree. The children will de-capsulate the multicast message and continue to forward the initial broadcast message. If the children of the

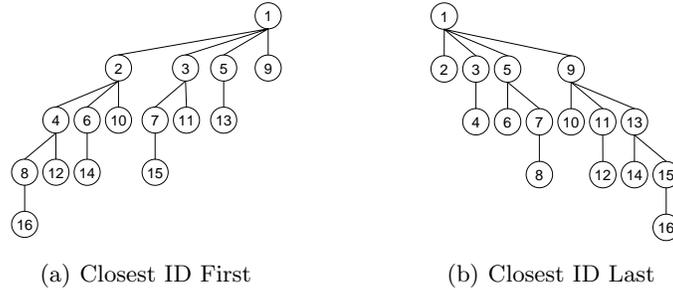


Fig. 7. Binomial Broadcast Tree

failed node also die, the message will be rerouted to its grandchildren automatically.

Under normal circumstances, all of the broadcast routing algorithms in BMG are loop avoidance because they always send broadcast messages using links that exist in the binomial spanning tree. Under failure circumstances, the rerouting mechanism is based on multicast messages, so the loop avoidance procedures are inherited from the multicast algorithms.

6 Conclusions and Future Work

This paper presents a scalable and fault tolerant topology called binomial graph (BMG). The capabilities in terms of both scalability and fault tolerance have been analyzed. The results show that BMG provides desirable topological properties for high performance computing. It has scalable capability properties such as reasonable degree, regular graph (every node has the same degree), low diameter, low average distance (both unicast and broadcast), symmetric graph (in the sense that average inter-nodal distance is the same from any source node), no numbers of node restriction, low cost factor and low message traffic density. The BMG also has good fault-tolerant properties such as optimal connectivity, low fault-diameter, strongly resilient and good optimal probability in failure cases. The paper also discusses several routing algorithms that can deliver the shortest path, while it maintains low complexity. These routing algorithms are suitable under both normal and failure circumstances.

There are several improvements that we plan for the near future. Making the routing algorithm aware of the underlying network topology (in both the LAN and WAN environments) will greatly improve the overall performance for both the unicast and broadcast message transmissions. This is equivalent to adding a function cost on each possible path and integrating this function cost to the computation of the shortest path. In the longer term, we hope that BMG will become the basic logical topology of the runtime environments within the FT-MPI and Open MPI libraries.

References

1. Dongarra, J.J., Meuer, H., Strohmaier, E.: TOP500 supercomputer sites. *Supercomputer* **13** (1997) 89–120
2. Saad, Y., Schultz, M.H.: Topological properties of hypercubes. *IEEE Transactions on Computers* **37** (1988) 867–872
3. Ohring, S., Das, S.K.: Folded Petersen cube networks: New competitors for the hypercubes. *IEEE Transactions on Parallel and Distributed Systems* **7** (1996) 151–168
4. Sivarajan, K.N., Ramaswami, R.: Lightwave networks based on de Bruijn graphs. *IEEE/ACM Trans. Netw.* **2** (1994) 70–79
5. Panchapakesan, G., Sengupta, A.: On a lightwave network topology using Kautz digraphs. *IEEE Transactions on Computers* **48** (1999) 1131–1138
6. Karol, M.J.: Optical interconnection using shuffle-net multihop networks in multi-connected ring topologies. In: *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols*, New York, USA, ACM Press (1988) 25–34
7. Maxemchuk, N.F.: Regular mesh topologies in local and metropolitan area networks. *AT&T Technical Journal* **64** (1985) 1659–1685
8. Campbell, S., Kumar, M., Olariu, S.: The hierarchical cliques interconnection network. *Journal of Parallel and Distributed Computing* **64** (2004) 16–28
9. Goodman, J.R., Sequin, C.H.: Hypertree: A multiprocessor interconnection topology. *IEEE Transactions on Computers* **30** (1981) 923–933
10. Angskun, T., Fagg, G.E., Bosilca, G., Pješivac-Grbović, J., Dongarra, J.: Scalable fault tolerant protocol for parallel runtime environments. In: *Recent Advances in PVM and MPI*. Number 4192 in LNCS, Springer (2006) 141–149
11. Angskun, T., Fagg, G.E., Bosilca, G., Pješivac-Grbović, J., Dongarra, J.: Self-healing network for scalable fault tolerant runtime environments. In: *Proceedings of 6th Austrian-Hungarian workshop on distributed and parallel systems*, Innsbruck, Austria, Springer-Verlag (2006)
12. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content addressable network. Technical Report TR-00-010, Berkeley, CA (2000)
13. Stoica, I., Morris, R., Karger, D., Kaashoek, F., Balakrishnan, H.: Chord: A scalable Peer-To-Peer lookup service for internet applications. In: *Proceedings of the 2001 ACM SIGCOMM Conference*. (2001) 149–160
14. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science* **2218** (2001) 329–350
15. Zhao, B.Y., Kubiawicz, J.D., Joseph, A.D.: Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, UC Berkeley (2001)
16. El-Amawy, A., Latifi, S.: Properties and performance of folded hypercubes. *IEEE Transactions on Parallel and Distributed Systems* **2** (1991) 31–42
17. Louri, A., Neocleous, C.: A spanning bus connected hypercube: A new scalable optical interconnection network for multiprocessors and massively parallel systems. *Journal of Lightwave Technology* **15** (1997) 1241–1252
18. Gibbons, A.: *Algorithmic graph theory*. Cambridge University Press (1985)
19. Dekker, A.H., Colbert, B.D.: Network robustness and graph topology. In: *ACSC '04: Proceedings of the 27th Australasian conference on Computer science*, Darlinghurst, Australia, Australian Computer Society, Inc. (2004) 359–368
20. Krishnamoorthy, M.S., Krishnamurthy, B.: Fault diameter of interconnection networks. *Computers and Mathematics with Applications* **13** (1987) 577–582