Clustering Multicast on Hypercube Network*

Lu Song, Fan BaoHua, Dou Yong, and Yang XiaoDong

College of Computer Science, National University of Defense Technology,
Changsha, Hunan 410073, People's Republic of China
lusong@nudt.edu.cn

Abstract. Multicast communication is one of the general patterns of collective communication in multiprocessors. On hypercube network, the optimal multicast tree problem is NP-hard and all existing multicast algorithms are heuristic. And we find that the existing works are far away from optimal. So this paper aims to design an more efficient algorithm to reduce the communication traffic of multicast in hypercube network. We propose a clustering model and an efficient clustering multicast algorithm. Compared with the existing related works by simulation experiments, our heuristic algorithm reduces the communication traffic significantly.

1 Introduction

A multicast communication in networks means that a source node sends a message to some destination nodes. The multicast algorithms are to determine the paths routing the message to the destination nodes. One goal of the researches on multicast is to reduce the communication traffic.

Depending on the different underlying switching, Lin and Ni[1] formulated the multicast communication problem in multicomputers as three different graph problems: the Steiner tree (ST) problem, the multicast tree (MT) problem, and the multicast path (MP) problem. The optimization of all these three multicast problems on hypercube networks has been proved to be NP-hard [1,2,9]. Lan, Esfahanian and Ni proposed a famous algorithm (LEN's MT algorithm)[2] for multicast tree problem on hypercube networks. Another heuristic algorithm for multicast tree problem was proposed by Sheu[3].

However, the existing algorithms are optimal or approximately optimal only under some special conditions. Moveover, routing following Sheu's algorithm may lead to a cycle in some case, which means the correctness of the algorithm can not be always guaranteed. In this paper, we propose a clustering model for hypercube and design a heuristic multicast algorithm. The idea is putting the neighboring nodes together to form a cluster. Then choose for the routing path based on the clusters.

In section 2, the definition of multicast tree and optimal multicast tree [7] are presented. We also propose two properties of multicast set, global-info and

* Supported by PCSIRT (Program for Changjiang Scholars and Innovative Research Team in University) of China under Grant No. IRT0446.

M. Gerndt and D. Kranzlmüller (Eds.): HPCC 2006, LNCS 4208, pp. 61–70, 2006.

locality, which are used for message routing. Then we outline the LEN's MT algorithm and Sheu's algorithm. In section 3, we propose a clustering model for hypercube, the relevant clustering algorithm and the clustering multicast algorithm with an illustration. Sections 4 presents the performance analysis and the result of simulation experiments. The final section is about the conclusions.

2 Related Works

A multicast communication can be supported by many one-to-one communications, which is called unicast-based multicast. In this method, system resources are wasted due to the unnecessary blocking caused by nondeterminism and asynchrony [5,6,7]. Even without blocking, multicast may reduce communication traffic and latency considerably. In this section, we state the multicast tree problem and outline LEN's and Sheu's algorithms.

2.1 Multicast Tree

Multicast tree problem is formulated as a graph problem by Lin and Ni[1]. And the optimal multicast problem is originally defined by [4].

Definition 1 (Multicast Tree). *Given a graph $G = (V, E)$, a source node u_0, and a multicast set $M \subseteq V$, a multicast tree is a subtree $G_T = (V_T, E_T)$ of G such that $M \subseteq V_T$ and for each $u \in M$, the path length from u_0 to u is equal to length on G.*

Definition 2 (Optimal Multicast Tree). *Given a graph $G = (V, E)$, the optimal multicast tree (OMT) $G_{OMT} = (V_{OMT}, E_{OMT})$ from source vertex u_0 to destination set $M = \{u_1, u_2, \cdots, u_k\}$ is a multicast tree of G such that $|E_{OMT}|$ is as small as possible.*

Here, we propose two properties of multicast set, global-info and locality. The global-info describes the distribution of the destination nodes. The locality describes the extent of neighboring. They are both used for message routing.

Definition 3 (Global-info). *The global-info consists of (1) the number of destination vertices $\|M\|$, (2) the counter of relative address (see definition 9) of all destination vertices on each dimension, $t \triangleq t_1 t_2 \cdots t_n$, $t = \sum_{i=1}^{k}(bitxor(u_i, u_0))$.*

Definition 4 (Locality). *The locality is the extent of nearness between vertices inside destination set, which can be measured by an array of distance D–array. There is no locality between deferent destination subset.*

$$D\text{–}array = \begin{bmatrix} H(u_1,u_1) & H(u_1,u_2) & \cdots & H(u_1,u_k) \\ H(u_2,u_1) & H(u_2,u_2) & \cdots & H(u_2,u_k) \\ \cdots & \cdots & \cdots & \cdots \\ H(u_n,u_1) & H(u_n,u_2) & \cdots & H(u_n,u_n) \end{bmatrix} .$$

2.2 LEN's MT Algorithm

In LEN's algorithm [2], when an intermediate node w receives the message and the destination set M, it has to check if it is a destination node itself. If so, it accepts the message locally and deletes itself from M. Then, it has to compute the relative address of all the destination nodes. For a destination u, the ith bit of u's relative address is 1 if u is different from w on dimension i. Hence, for each dimension $i(0 \leqslant i \leqslant n-1)$, LEN's algorithm counts how many destination nodes whose ith bit of the relative address is 1. After that, it always chooses a particular dimension j with the maximum count value. All destination nodes whose jth bit of the relative address is 1 are sent to the neighbor of w on jth dimension. Then, these nodes are deleted from destination set M. This procedure is repeated until the multicast set M becomes empty.

LEN's MT algorithm votes for the paths to route the message depending on the global-info, without considering the locality between the destination nodes. Hence, the result from LEN's algorithm is far from optimal. Let us give an example. Consider Q_5, suppose the source node is 00000(0) and the multicast destination set M = {01010(10), 11101(29), 10001(17), 11111(31), 11100(28), 01011(11), 00010(2), 10110(22), 11110(30), 11011(27)}. The result from LEN's algorithm is shown in figure 1(a), where the gray nodes are destination nodes and the nodes with dashed line are intermediate nodes. A better result from our algorithm is shown figure 1(b).

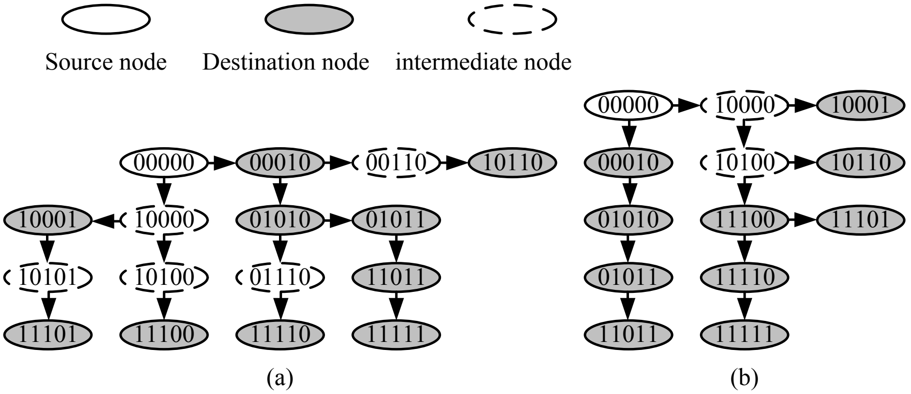

Fig. 1. The results from LEN's MT algorithm and our algorithm

2.3 Sheu's Algorithm

Sheu's algorithm [3] consists of two phases, i.e. the neighbors linking phase and the message routing phase. The neighbors linking phase is executed only on the source node. It links the destination nodes in the multicast set M which are adjacent. After this phase, the multicast set becomes a neighboring forest in which each element is a root of tree. The routing phase is executed on each intermediate nodes in the multicast tree. Similar to LEN's MT algorithm, this phase votes for the paths to routing the message.

Sheu' algorithm votes for the paths depending on the global-info and the locality between the nodes. However, the neighbor linking phase make the global-info injured without reserving weight of the trees. Under particular condition, the result from Sheu's algorithm has a cycle. Here is an example. Consider Q_9, suppose the source node is 000000000(0) and the multicast destination set $M = \{000010000(16), 000011000(24), 000011100(28), 000001111(15), 000011111(31), 110011111(415), 001111111(127)\}$. There is a cycle in the result from Sheu's algorithm shown by figure 2, where two branches arrive node 000011111(31). In the neighbor linking phase, the global-info losses the node 000011111(31) which should be used to vote for the paths to node 110011111(415) and 001111111(127). It means that Sheu's MT algorithm is not always right. In the rest of this paper, we leave Sheu's algorithm out of account.

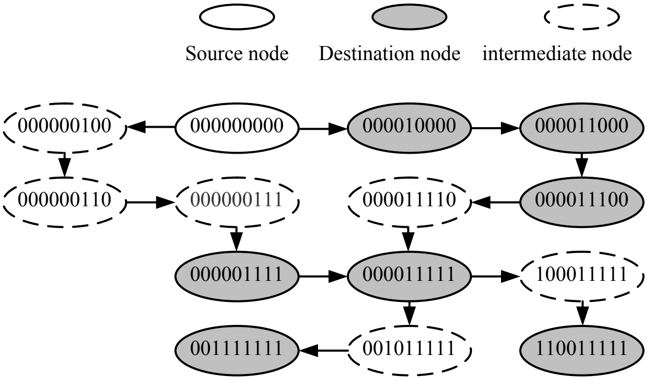

Fig. 2. A cycle existed in result of Sheu's algorithm

3 Clustering Multicast Algorithm

In this section we propose a clustering model. Based on the model, our multicast tree algorithm consists of two phases, the clustering phase and the clustering multicast phase. The Algorithm 1 describes the clustering phase executed only on the source node. The clustering multicast phase is executed on each intermediate node in the multicast tree. This procedure is shown by algorithm 2.

3.1 Clustering Model

Since one of the goals in the MT problem is to minimize the traffic, a multicast tree is better if it has fewer nodes in the multicast tree. Actually, there are at least $k + 1$ nodes to form a tree for a 1-to-k multicast communication. Suppose the multicast set contains only two adjacent node u and v. The multicast tree formed by putting two nodes on the same path will never be worse than that formed by putting them on different paths, because the former case never leads to additional traffic. It is also suitable for the case that node u and node v are neighboring, $(H(u, u_0) \gg H(u, v), H(v, u_0) \gg H(u, v))$. The nearness between nodes is called locality (Definition 4). Based on the idea of locality, we propose a clustering model which can split the multicast set into clusters.

Based on the definition of hypercube [8,10], we propose the definition of subcube, expansion subcube, distance and relative address.

Definition 5 (Hypercube). *A hypercube is defined as a graph $Q_n = (V, E)$. The vertex set V of Q_n consists of all binary sequence of length n on the set $\{0, 1\}$, $V = \{x_1x_2 \cdots x_n | x_i \in \{0, 1\}, i = 1, 2, \cdots, n\}$. Two vertices $u = u_1u_2 \cdots u_n$ and $v = v_1v_2 \cdots v_n$ are linked by an edge if and only if u and v differ exactly in one coordinate, $E = \{(u, v) | \sum_{i=1}^{n} |u_i - v_i| = 1\}$.*

Definition 6 (Subcube of Q_n). *A subcube H_k of Q_n is a binary sequence $b_1b_2 \cdots b_{n-k}$ of length $(n-k)$ which presents a subgraph containing 2^k vertices and $k \cdot 2^{k-1}$ edges. H_k can be denoted as $b_1b_2 \cdots b_{n-k} \star\star\star$ inside of which the vertex has a form like $b_1b_2 \cdots b_{n-k}x_{n-k+1} \cdots x_n (x_j \in \{0, 1\}, n-k+1 \leqslant j \leqslant n)$.*

Definition 7 (Expansion subcube). *A expansion subcube of destination set M denoted as $expan(M)$ is the minimal subcube containing M. $expan(M) = \{H_k | k \leqslant i (\forall i, M \subseteq H_i)\}$.*

Definition 8 (Distance). *The distance between vertices is defined as the Hamming distance, $H(u, v) = \sum_{i=1}^{n} |u_i - v_i| (u = u_1u_2 \cdots u_n, v = v_1v_2 \cdots v_n)$. The distance between vertex sets is defined as $H(U, V) = \min_{u_i \in U, v_j \in V}(H(u_i, v_j))$.*

Definition 9 (Relative address). *The relative address of vertex u to vertex v is a binary sequence $R(u, v) \triangleq r_1r_2 \cdots r_n$, where $r_i = u_i \oplus v_i (u = u_1u_2 \cdots u_n, v = v_1v_2 \cdots v_n)$.*

Using definition 5,6,7,8,9, we present the definition of cluster.

Definition 10 (Cluster). *Cluster is a set of destination nodes that are near each other. There are two metrics of a cluster c, weight and degree. (1)Weight. Weight is defined by the number of nodes in the cluster. $W(c) = \|c\|$.(2)Degree. Degree is defined as the dimension of the expansion subcube of cluster c. $D(c) = k(H_k = expan(c))$.*

Actually, considering each node as a cluster, LEN's algorithm is a kind of generalized clustering multicast algorithm where the degree of each cluster is zero. It means that LEN's algorithm doesn't use the locality between destination nodes. As mentioned above, the result from LEN's algorithm is far away from optimal.

In next two subsections, we propose a clustering algorithm to split the multicast set into clusters and a clustering multicast algorithm based on clusters with their priorities.

3.2 Clustering Algorithm

In this subsection, we propose a clustering algorithm (algorithm 1) which is used to split the multicast set into clusters. In algorithm 1, all the destination nodes are classified by their distance to the source node. Then we find adjacent nodes that are directly linked by an edge in the graph of hypercube, and put these adjacent nodes together to form a tree. After step 2, we get a set of trees. At step 3, we combine the trees with the one whose expansion subcube contains the other's. The combination forms a cluster. Thus, the cluster set is generated by our clustering algorithm.

Algorithm 1. Clustering algorithm on Q_n

```
        Input  : source node $u_0$, multicast set $M = \{u_1, u_2, \cdots, u_k\}$
        Output: cluster set $C$
step 1: $M_0 \leftarrow \{u_0\}, M_1 \leftarrow M_2 \leftarrow \cdots \leftarrow M_n \leftarrow \emptyset$
        for $i \in [1, n]$ do
            $M_i = \{u_j | H(u_j, u_0) = i\}$
        end
step 2: $\forall i \in [1, n]$;  for $u \in M_i$ do
            if $\exists v \in M_{i-1}$ && $H(u, v) == 1$ then
                $M_{i-1} = M_{i-1} \cup \{u\}$
                $M_i = M_i - \{u\}$
            end
        end
        $C = M_1 \cup M_2 \cup \cdots \cup M_n$
step 3: $\forall i, j \in [1, n]$; if $expan(c_i) \subseteq expan(c_j)$ then
            $c_j \leftarrow c_j \cup c_i$
            $c_i \leftarrow \emptyset$
        end
```

3.3 Clustering Multicast

In this subsection, we present the clustering multicast algorithm (algorithm 2). At step 1, if local node is contained in the multicast set, it deletes itself from the multicast set. At step 2, for each cluster, we find out the head that is nearest to local node in the cluster. Then compute the relative address of each head to local node. Now we can use the relative addresses as the message routing direction and the weight of clusters as the priority. At step 3, for each cluster and dimension i, we counts the product of wight and ith bit of relative address. Similar to LEN's algorithm, we choose a particular dimension j with the maximum count value. All clusters whose relative address of the head has 1 on jth bit are sent to the neighbor of local node on jth dimension. This procedure is repeated until the cluster set C becomes empty.

The following theorem shows the correctness of clustering multicast Algorithm.

Algorithm 2. Message routing phase

```
         Input   : Local node address $w$, cluster set $C$
         Output: None
step 1: if $\exists c \in C, w \in c$ then
            send the message to local processor
            delete $w$ from $c$
         end
step 2: foreach $c_i \in C$ do
            $u_i \leftarrow min(H(u_j, w)); (u_j \in c_i)$
            $u_i \leftarrow \text{bitxor}(u_i, w)$
            $w_i \leftarrow \|c_i\|$
         end
step 3: while $C \neq \emptyset$ do
            $t \leftarrow u_i * w_i;\ (t \triangleq t_1 t_2 \cdots t_n)$
            $t_j \leftarrow \max_{1 \leq i \leq n}(t_i)$
            $C' \leftarrow \{c_i | c_i \in C, u_i$ and $w$ diff on dimension $j\}$
            if $\exists c \in C, H(\{w\}, c) \geqslant H(c, C')$ then
                $C' \leftarrow C' \cup \{c\}$
                $C \leftarrow C - c$
            end
            Transmit $C'$ and the message to node $(w \oplus 2^j)$
            $C \leftarrow C - C'$
         end
```

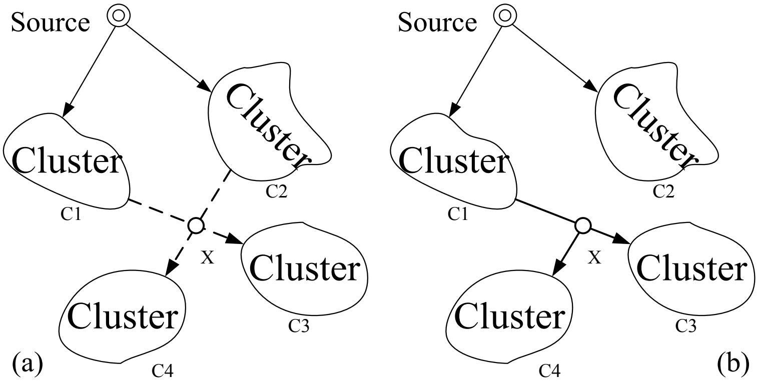

Fig. 3. Routing between clusters

Theorem 1. *Given a cluster set C in Q_n, the edges selected by clustering multicast algorithm form a multicast tree.*

Proof. (Proof by Contradiction.) Assume to the contrary that there is a cycle in the result from clustering multicast algorithm. It means that paths between clusters intersect at a node. We may suppose that the path from cluster c_1 to cluster c_3 and the path from cluster c_2 to cluster c_4 intersect at node x (see figure 3(a)). The distance between each cluster and node x is denoted as d_1,d_2,d_3 and d_4. There are two cases to consider. (1) $d_1 == d_2$. Because the algorithm 2 executed in sequence, cluster c_3 and c_4 can only be both appeared in the branch from cluster c_1(or c_2), contradicting our assumption. (2) $d_1 \neq d_2$. We may assume $d_1 < d_2$. Then we get $d(c_1, c_4) < (c_2, c_4)$. It means that cluster c_4 is closer to c_1 than c_2. According to clustering multicast algorithm, cluster c_3 and c_4 are both appeared in the branch from cluster c_1 (shown in figure 3(b)), again a contradiction. This completes the proof. □

4 Performance Analysis

4.1 Complexity Compare

Consider the 1-to-k multicast on Q_n. The complexities of step 1 and step 2 in clustering algorithm (algorithm 1) are both $O(nk)$. Since the elements of C are far less than M, the complexity of step 3 is $O(step3) < O(nk)$. Hence the complexity of clustering algorithm is $O(nk)$. Similarly in message routing algorithm, $O(step1) = O(k)$, $O(stpe2) = O(step3) < O(nk)$, the complexity of clustering multicast is $O(nk)$. So the complexity to finish multicast is $O(nk)$. And the complexity of LEN's algorithm is also $O(nk)$[1,2].

And the transfer latency of LEN's algorithm and our algorithm are approximately equal. The latency of multicast was composed of transferring time and routing time which are independent with each other. The transferring time is dependent on the electric mechanism. Since the LEN's algorithm and our algorithm are both with the complexity of $O(nk)$, the routing time of two algorithm are approximately equal. Then so do the total latency.

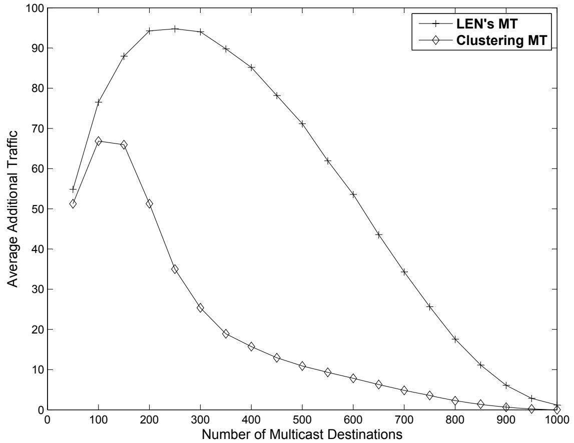

Fig. 4. Average additional traffic generated by LEN's MT algorithm and clustering multicast

4.2 Simulation

In this subsection, we analysis the performance of our multicast algorithms on Q_{10} by simulation experiments. In general, the distribution of destination nodes has a great effect on the total traffic of multicast communication. But, similar to the performance studies in existing works [1,2,3], we assume that the routing distribution is *uniform*. For a 1-to-k multicast, it requires at least *k units of traffic*, where a *unit of traffic* is measured as one message transmitted over one link. As in the existing studies [1,2,3], we use *average additional traffic* to evaluate the performance of a multicast communication, where the *average additional traffic* is defined as the average amount of total traffic minus k. The number of destination nodes k is ranged from 50 to 1000 by the step of 50. For each k, we perform the simulation 500 times and the amount of traffic generated for a given k is averaged over the 500 runs. Figure 4 shows the comparison of average additional traffic generated by LEN's MT algorithm and our multicast algorithm.

5 Conclusions

In this paper, we propose a clustering model and a clustering algorithm to achieve a better balance between the global-info and locality which are two essential properties of the multicast set. Based on the clustering model, an efficient heuristic multicast algorithm is presented. By simulation experiments, our clustering multicast algorithm has significant improvements compared to the existing algorithms. And the generation of resulting multicast tree is fully distributed.

References

1. X. Lin and L. M. Ni: Multicast communication in multicomputer networks. IEEE Trans. Parallel Distrib. Systems **4** (1993) 1105-1117
2. Y. Lan, A. H. Esfahanian, and L. M. Ni: Multicast in hypercube multiprocessors. J. Parallel Distrib. Comput. **8** (1990) 30-41
3. Shih-Hsien Sheu and Chang-Biau Yang: Multicast Algorithms for Hypercube Multiprocessors. Journal of Parallel and Distributed Computing **61** (2001) 137-149
4. Y. Lan, A. H. Esfahanian, and L. M. Ni: Distributed multi-destination routing in hypercube multiprocessors. Proceedings of the Third Conference on Hypercube Concurrent Computers and Applications 1988 631-639
5. Choi Y., Esfahanian A. H., and Ni L. M.: One-to-k communication in distributed-memory multiprocessors. Proc. 25th Annual Allerton Conference on Communication, Control, and Computing 1987 268-270
6. William James Dally and Brian Patrick Towles: Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers 2003
7. J. Duato, S. Yalamanchili, and L. M. Ni: Interconnection Networks: An Engineering Approach. Morgan Kaufmann Publishers 2002

8. Junming Xu: Topological structure and Analysis of Interconnection Networks. Kluwer Academic Publishers 2001
9. R. L. Graham and L. R. Foulds: Unlikelihood that minimal phylogenies for realistic biological study can be constructed in reasonable computational time. Math. Biosci. **60** (1982) 133-142
10. Jianer Chen, Guojun Wang and Songqiao Chen: Locally subcube-connected hypercube networks: theoretical analysis and experimental results. IEEE Transactions on Computers **5** (2002) 530-540