Leaf Communications in Complete Trees *

Vassilios V. Dimakopoulos and Nikitas J. Dimopoulos

Department of Electrical and Computer Engineering,
University of Victoria
P.O. Box 3055, Victoria, B.C., CANADA, V8W 3P6.

Abstract. In this work we consider tree-based interconnects where the processing nodes are confined to the leaves of the tree. These types of interconnects include fat tree networks. We present and analyze algorithms for collective communications problems that include broadcasting, scattering/gathering, multinode broadcasting and total exchange.

1 Introduction

Distributed memory multiprocessors are based on a collection of independent processing nodes integrated through an interconnection network. *Collective communications* problems include: *broadcasting*, *multinode broadcasting*, *scattering (gathering)* and *total exchange*. The need for their efficient solution was realized quite early, especially in the context of parallel numerical algorithms [2].

Studies in collective communication problems for hypercube-networks appear in [11, 8, 1] and for non-hypercube ones in [12]. Two excellent surveys on the subject can be also found in [7, 6].

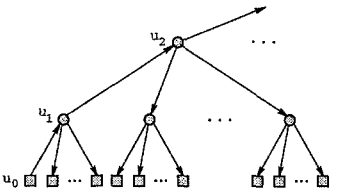
In this work we consider the complete tree topology where processors are confined to the leaf level. *Fat trees* [9, 10], are also complete trees but with branch capacities increasing towards the root. Here we consider k-ary trees where k is any integer greater than one. We shall study the above communication problems and determine the time requirements for solving them in the optimum time. Related but not exactly applicable to fat trees are the works in [5, 3].

We note that this paper provides only a short report of the results, avoiding formal proofs. A more detailed exposition of the material can be found in [4].

1.1 Preliminaries

In a complete k-ary tree each node has k children, except for the leaves. The tree consists of $\log_k n + 1$ levels. Level 0 is the leaf level and level $h = \log_k n$ is the level of the root; n denotes the number of leaves, which will be numbered from left to right as $0, 1, \ldots, n-1$. The leaves correspond to *processing* nodes while all the other levels include only *routing* nodes. A *fat tree* is based on the same topology only branches increase in capacity as one moves towards the root. Branches between levels $i-1$ and i have capacity $c_i \geq 1$ corresponding to the number of physical links included in the branch. Two capacity patterns are of interest: constant ($c_i = 1$; $i = 1, 2, \ldots, h$) and exponential ($c_i = k^{i-1}$).

* This research was supported in part through grants from NSERC and the University of Victoria.



(a)

Source Node:
1 Send message to parent;
Every Routing Node:
1 Send message to parent (if any);
2 Send message to each child in turn, except to the one the message came from;

(b)

Fig. 1. Broadcasting under the single-port model (a) labeling (b) algorithm

We assume packet-switching. Messages consist of a single packet. Transferring a message between two neighbors occurs in one time unit (or step). We consider two models of communication: in the first one, nodes will be able to utilize only one of their output links at a time (*single-port* model). In the *multiport* model, all links incident to a node can be utilized simultaneously.

2 Communications under the single-port model

In this section we shall derive lower bounds for the communication problems we consider and shall provide algorithms that achieve the lower bounds. Notice that we allow any node to receive messages from *all* its neighbors simultaneously but it can only send one message at a time; thus branch capacities have no effect.

2.1 Broadcasting

Let $B_r(\ell)$ denote the broadcast time from a node at level ℓ to the leaf nodes.

$$B_r(\ell) = B_r(\ell-1) + k \Rightarrow B_r(\ell) = k\ell. \tag{1}$$

Our goal though is to broadcast from a leaf of a complete k-ary tree. Let the leaf, labeled u_0 in Fig. 1(a), be the source node. Also, let $b(T_{u_i})$ be the broadcast time remaining after u_i receives the message. Node u_h is the root of the tree. When u_h receives the message it must inform its $k-1$ remaining children and for this it needs $B_r(h-1)+k-1$ steps. Consequently, $b(T_{u_h}) = B_r(h-1)+k-1$.

Now consider u_{h-1}. This node has to inform $k-1$ subtrees lying below plus its parent, u_h. For the subtrees it needs $B_r(h-2)+k-1$ steps, while broadcasting from u_h, needs $B_r(h-1)+k-1$ steps. Observe that the best plan is to first send the message to the subtree with the largest broadcast time, (i.e. to u_h first). Consequently, $b(T_{u_{h-1}}) = \max\{b(T_{u_h})+1, B_r(h-2)+k\} = b(T_{u_h})+1$.

Proceeding downwards in this manner it is seen that any node u_i must first inform its parent (u_{i+1}) and then its $k-1$ remaining children, and

$$b(T_{u_i}) = \max\{b(T_{u_{i+1}})+1, B_r(i-1)+k\} = b(T_{u_h})+h-i,$$

giving $b(T_{u_0}) = b(T_{u_h})+h = B_r(h-1)+k-1+h$. Since from (1) $B_r(h-1) = k(h-1)$, we have the following result (recall that $h = \log_k n$):

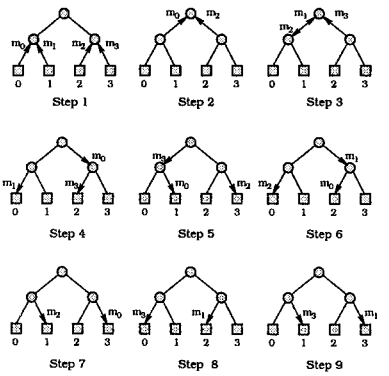
Theorem 1. *Broadcasting in fat trees under the single-port model requires* $(k+1)\log_k n - 1$ *steps.*

Every Leaf Node:
Send message to parent;

Every Routing Node: (except the root)
A Receive all messages from children, sending a copy of them upwards, one-by-one;
B Until the last message arrives from parent
Send one message to each child in turn;

Root Node:
B While receive all messages from children, keep sending one message to each child in turn;

(a)



(b)

Fig. 2. Multinode broadcasting under the single-port model (a) optimal algorithm (b) example in 4 leaves

2.2 Scattering

The observation that the source node has to send or receive $n-1$ different messages over its incident link leads to a lower bound of $S(n) = G(n) \geq n-1$ steps. In reality, the exact bound is n or $n+1$ steps. In gathering $n-1$ messages at processor 0, in the first two steps we can at most receive one message since the closest leaf is at distance two. Therefore, there will be at least one step with no message reception by node 0, i.e. $S(n) = G(n) \geq n$. If the tree is binary there will be one extra step of no reception since the next closest leaves (2 and 3) are both at distance four from node 0, i.e. $S(n) = G(n) \geq n+1$ if $k = 2$.

Gathering algorithms can be had from scattering ones by reversing the data paths. The lower bound is achieved using *furthest-first* scheduling, whereby the source node gives priority to messages that are destined the furthest.

Theorem 2. *The furthest-first discipline results in an optimal scattering algorithm.*

2.3 Multinode broadcasting

In multinode broadcasting every node broadcasts its own message. An optimal multinode algorithm schedules the traffic in each routing node so that all leaves receive all messages at the minimum possible time. The subsequent two theorems (the proofs of which can be found in [4]) establish the lower bound and the optimality of a multinode broadcasting algorithm.

Theorem 3. *Any multinode broadcasting algorithm under the single-port model requires at least* $kn + (k+1)(\log_k n - 2) + 1$ *steps.*

Theorem 4. *The algorithm presented in Fig. 2(a) is optimal.*

Every Leaf Node:
Send the $n-1$ messages in a furthest-first order;
Every Routing Node: (except the root)
A While there exist upward messages from children
Send one message to parent;
B Until the last message arrives from parent
Send one message to any (appropriate) child;
Root Node:
B While receiving all messages from children, keep sending one message to any (appropriate) child;

Fig. 3. Optimal total exchange algorithm under the single-port model

2.4 Total exchange

Under the single-port model we may easily determine the lower bound of total exchange algorithms, in a similar fashion to the proof of Theorem 3.

Theorem 5. *Any total exchange algorithm under the single-port model requires at least* $n^2(2k+1)(k-1)/k^3 + 2\log_k n - 3$ *steps.*

An algorithm that achieves the lower bound of Theorem 5 is given in Fig. 3. (For the proof of its optimality see [4].)

Theorem 6. *The algorithm presented in Fig. 3 is optimal.*

3 Communications under the multiport model

We shall now assume that a node is able to utilize all its incident links simultaneously. This model of communication is affected by the capacity arrangement on the branches of the tree. We shall further assume that the capacity of level-1 branches is $c_1 = 1$.

3.1 Single-source communications

In any network, broadcasting from a node under the multiport model takes time equal to d, where d is the distance between the source node and a node farthest from the source. In our case, broadcasting will require $B(n) = 2h = 2\log_k n$ steps. It is easily accomplished by setting the routing nodes to a broadcast mode whereby the received message is replicated towards all directions.

Scattering and gathering, under our assumptions, is governed by the same bounds as in the single-port case; there is only one link available from a leaf, forcing only one message to be sent or received at a time. Consequently, $S(n) = G(n) \geq n$ (or $n+1$ if $k=2$). Had we allowed $c_1 > 1$, different lower bounds would have been derived.

3.2 Multinode broadcasting

The same argument used for deriving bounds for scattering/gathering algorithms can be used to determine lower bounds for multinode broadcasting in the multiport model since every leaf has to receive $n-1$ different broadcast messages. The exact bounds are $MB(n) \geq n$ if $k \geq 3$ or $MB(n) \geq n+1$ if $k=2$.

```
1   For all i = 0 to h − 1
        /* Phase i */
2       Do in parallel for all level h − i nodes
3           Transfer all messages from each of the k subtrees
            to the other k − 1 subtrees;
```

Fig. 4. A total exchange algorithm with no contention

Theorem 7. *Multinode broadcasting can be performed in time equal to the lower bound.*

Proof. The lower bound can be achieved by following a "flooding" procedure: each node replicates every received message to all possible directions (expect the one the massage came from). The proof can be found in [4].

This flooding algorithm achieves the lower bound but it requires large queues. In [4] we give an algorithm which is suboptimal by $2\log_k n - 2$ steps but it eliminates the queues.

3.3 Total exchange

A simple lower bound for the total exchange problem is $TE(n) \geq n$ (or $TE(n) \geq n+1$ if the tree is binary), since multinode broadcasting can be performed in at most as many steps as total exchange [2]

A total exchange algorithm that works for any capacity pattern and induces no queueing is as follows. Initially, the k subtrees of the root node exchange their $n(n-k^{h-1})$ messages meant for each other. Then, the k subtrees perform internally a total exchange in parallel. The algorithm is stated in Fig. 4.

During the ith phase a node at level $h-i$ has to pass $k^{h-i}(k^{h-i}-k^{h-i-1})$ messages over its k incident branches which have capacity c_{h-i}. This means that a maximum of kc_{h-i} messages can cross towards the node at a time. To avoid contention while maintaining maximum speed, exactly kc_{h_i} leaves should dispatch messages at a single step, and the messages should be appropriately chosen so that their destinations are distinct. By calculating the time needed for each phase, it can be seen that the algorithm needs time

$$T = \sum_{i=1}^{h} \left\lceil \frac{(k-1)k^{2i-2}}{c_i} \right\rceil + h^2.$$

Instead of executing the phases serially, it is possible to pipeline the phases appropriately and save in total h^2-2h+1 steps (see [4]). The pipelined algorithm has a final number of steps

$$T = \sum_{i=1}^{h} \left\lceil \frac{(k-1)k^{2i-2}}{c_i} \right\rceil + 2h - 1. \tag{2}$$

For trees with exponential capacities, $c_i = k^{i-1}$, we see, from (2), that our algorithm requires $T = k^h+2h-2 = n+2h-2$ steps. Consequently, the algorithm

we presented above is suboptimal by $2\log_k n - 2$ steps. It is interesting to see whether the last bound is tight or not. We have shown [4] that for the case of trees with exponential capacities arrangement, a bound tighter than $TE(n) \geq n$ exists which guarantees that the total exchange algorithm we presented is very close to optimal (within $2\log_k \log_k n$ steps) for such trees.

Theorem 8. *An optimal total exchange algorithm for exponential capacity trees needs at least* $n + 2\log_k n - 2\log_k \log_k n - 2$ *steps.*

4 Conclusion

We studied the implementation and performance of communication operations in complete k-ary trees where the processing nodes are confined to the leaf level. The results can be easily generalized to the case where nodes in level i have k_i children, where $n = k_1 k_2 \cdots k_h$. However, there are still a number of issues to be considered. Multinode broadcasting has excessive queueing requirements if it is to be performed in the minimum number of steps. It would be interesting to see what is the lower time bound under the constraint of no queueing. Another issue for consideration is total exchange under the multiport model. The algorithm we presented is close to optimal especially in the case of exponential capacities. Improved bounds though for the total exchange problem need to be found as the straightforward one does not seem to be tight.

A detailed exposition of this material is available in [4] and can be obtained through the World Wide Web at `http://www-lapis.uvic.ca`.

References

1. D.P. Bertsekas, et. al. "Optimal communication algorithms for hypercubes," *J. Parallel Distrib. Comput.*, Vol. 11, pp. 263–275, 1991.
2. D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewoods Cliffs, N.J.: Prentice - Hall, 1989.
3. S. N. Bhatt, et. al. "Scattering and gathering messages in networks of processors," *IEEE Trans. Comput.*, Vol. 42, No. 8, pp. 938–949, Aug. 1993.
4. V. V. Dimakopoulos and N. J. Dimopoulos, "Leaf communications in complete trees," Technical Report ECE-95-6, University of Victoria, Oct. 1995.
5. R. Feldmann, et. al. "Optimal algorithms for dissemination of information in generalized communication modes," in *Proc. 4th PARLE, Parallel Architectures and Languages Europe*, Paris, France, June 1992, pp. 115–130.
6. P. Fraigniaud and E. Lazard, "Methods and problems of communication in usual networks," *Discrete Appl. Math.*, Vol. 53, pp. 79–133, 1994.
7. S. M. Hedetniemi, et. al. "A survey of gossiping and broadcasting in communication networks," *Networks*, Vol. 18, pp. 319–349, 1988.
8. S. L. Johnsson and C. - T. Ho, "Optimum broadcasting and personalized communication in hypercubes," *IEEE Trans. Comput.*, Vol. 38, pp. 1249–1268, 1989.
9. C. E. Leiserson, "Fat-trees: universal networks for hardware-efficient supercomputing," *IEEE Trans. Comput.*, Vol. C-34, No. 10, pp. 892–901, Oct. 1985.
10. C. E. Leiserson and et al, "The network architecture of the Connection Machine CM-5," in *Proc. 4th ACM Symp. Parall. Algor. Arch.*, June 1992, pp. 272–285.
11. Y. Saad and M. H. Schultz, "Data communications in hypercubes," *J. Parallel Distrib. Comput.*, Vol. 6, pp. 115–135, 1989.
12. Y. Saad and M. H. Schultz, "Data communications in parallel architectures," *Parallel Comput.*, Vol. 11, pp. 131–150, 1989.