Leaf Communications in Complete Trees*

Vassilios V. Dimakopoulos Nikitas J. Dimopoulos*

Department of Electrical and Computer Engineering,
University of Victoria
P.O. Box 3055, Victoria, B.C., CANADA, V8W 3P6.
Tel: (604) 721-8773, 721-8902, *Fax:* (604) 721-6052,
E-mail: {dimako, nikitas}@ece.uvic.ca

February 17, 1996

Abstract

One of the crucial issues associated with loosely-coupled multiprocessors based on interconnection networks is information dissemination. A number of regular communication patterns seem to cover a large portion of information dissemination requirements in practice: broadcasting, scattering/gathering, multinode broadcasting and total exchange. In this work we consider tree networks where the processing nodes are confined to the leaves of the tree. This type of hierarchical topology is also the basis of fat tree networks. We present and analyze algorithms and their timing requirements for the aforementioned problems under two models of communication capabilities.

For:

WORKSHOP W02

Routing and Communication in Interconnection Networks

*The second author is responsible for correspondence. This research was supported in part through grants from NSERC and the University of Victoria.

1 Introduction

Distributed memory multiprocessors are based on a collection of independent processing nodes integrated through an interconnection network. The presence of locally generated data spawns the need for information dissemination: a process through which "knowledgeable" nodes inform the rest of the network about the items of information they possess.

A number of information dissemination problems appear frequently in practice and their effective solution is of paramount importance to system performance. They include: *broadcasting* where a particular node needs to send the same message to all the other nodes in the network; *multinode broadcasting* where every node performs a broadcasting; *scattering (gathering)* where a particular node needs to send (receive) different messages to (from) all the other nodes; *total exchange* or *multiscattering* where every node performs scattering or gathering; that is, every node has distinct messages to send to the other nodes in the network. These communication problems are also known as *collective communications*. The need for their efficient solution was realized quite early, especially in the context of parallel numerical algorithms [2]. Broadcasting is an essential operation in almost every parallel algorithm. Multinode broadcasting arises naturally in iterative algorithms. Total exchange is a communication pattern occurring for example in FFT algorithms and is usually identified with matrix transposition.

Saad and Schultz [11], Johnson and Ho [8] and Bertsekas *et al* [1] are some of the researchers that studied the above communication problems for hypercubes. Since then there has been a considerable amount of work on communication algorithms for networks other than hypercubes and for a number of different models of communication; see for example [12]. Two excellent surveys on the subject were given by Hedetniemi, Hedetniemi and Liestman [7] and Fraigniaud and Lazard [6].

In this work we consider the complete tree topology where processors are confined to the leaf level. The well-known *fat trees*, introduced by Leiserson [9] and utilized in Thinking Machine's CM-5 multiprocessor [10], are also based on the complete tree topology. The tree is termed "fat" because the capacities of the branches do not remain constant but rather increase towards the root. Originally Leiserson studied binary trees but CM-5 is based on quaternary ones. Here we will consider k-ary trees where k is any integer greater than one. We will study the above communication problems in the context of such a topology and determine the time requirements for solving them in the optimum time. Related but not

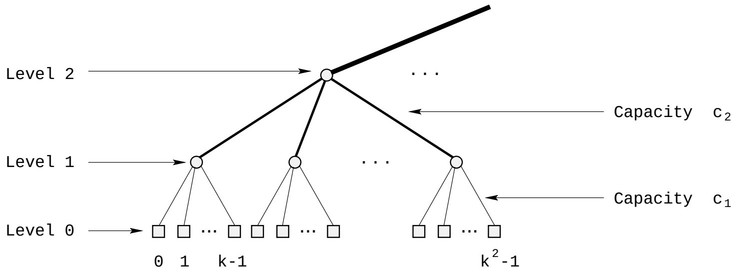


Figure 1: A k-ary tree

exactly applicable to fat trees are the works in [5, 3]. The authors in [5] studied among other problems the problem of broadcasting in complete trees, while in [3] scattering under the single-port model in arbitrary trees was considered. In both cases the network was assumed to be wormhole routed and, unlike fat trees, all nodes were assumed to generate and consume messages.

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

1.1 Preliminaries

Portion of a complete k-ary tree is shown in Fig. 1. Each node has k children, except for the leaves. The tree consists of $\log_k n + 1$ levels of nodes. Level 0 of the tree is the leaf level and level $h = \log_k n$ is the level of the root node; n is a power of k and denotes the number of leaves, which will be numbered from left to right as 0, 1, ..., $n-1$, as shown in the figure. The leaves correspond to *processing* nodes while all the other levels include only *routing* nodes. We will assume that the tree has at least three levels, i.e. that $h \geq 2$ or equivalently, $n \geq k^2$ (if $h = 1$ the graph degenerates to a star which consists of n vertices adjacent to an additional central vertex). A *fat tree* is based on the same topology only branches get thicker (i.e. increase in capacity) as one moves from the leaves to the root. Branches between levels $i-1$ and i have capacity $c_i \geq 1$ and if $j > i$ then $c_j \geq c_i$. The branch capacity corresponds to the number of physical links included in the branch.

Although the results presented here cover any capacity arrangement, two capacity patterns will be of more interest in later sections: constant and exponential. These two patterns represent two extremes in interconnection size. In the first case $c_i = 1$ for all levels $i = 1, 2, \ldots, h$ and the resulting tree is the simple k-ary tree. In the second case, $c_i = k^{i-1}$.

Every level of such a tree has n links in total, being able to carry messages from all leaves at the same time.

The network is assumed to be packet-switched. Messages are of the same size and consist of a single packet, so that transferring a message between two neighbors needs a constant amount of time which we take as one time unit (or step). The links are assumed to be bidirectional and fully duplex; half-duplex links where only one direction can be accommodated at a time cause a slowdown by at most a factor of two in some of the algorithms. Each of the routing nodes is equipped with packet queues (although this will not always be necessary for the algorithms we will discuss). Finally, we will consider two models of communication capability: 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

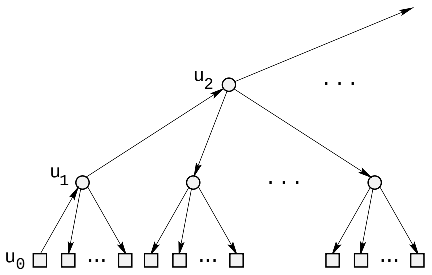
It should be clear that if every node is incapable of sending more than one message at every time unit, branch capacities have no effect at all. Any fat tree behaves exactly like the corresponding complete k-ary tree. In this section we will derive lower bounds for the communication problems we consider and will 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 each step.

2.1 Broadcasting

Let us first consider broadcasting from the root v of an arbitrary tree T, where v has d children v_i, $i = 1, 2, \ldots, d$. Let T_{v_i} be the subtree rooted at v_i and assume that broadcasting in T_{v_i} needs $b(T_{v_i})$ steps. Let us further assume for convenience that the trees have indices reversely to their broadcast times, i.e. if $i \leq j$ then $b(T_{v_i}) \geq b(T_{v_j})$. Since the root can send the message to only one child at a time it is clear that the best plan is to send the message to v_1 first, then to v_2, etc. As a result, the time needed to broadcast from v in T is

$$b(T) = \max_{1 \leq i \leq d} \{b(T_{v_i}) + i\}. \tag{1}$$

If T is a complete k-ary tree with height ℓ then the k subtrees rooted at the children of root v are identical: they are all complete k-ary trees with height $\ell - 1$. Let $B_r(\ell)$ be the



(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)

Figure 2: Broadcasting under the single-port model (a) labeling (b) algorithm

broadcast time from the root in T. Then the broadcast time for any of the k subtrees is $B_r(\ell-1)$, and from (1), setting $d=k$, $b(T)=B_r(\ell)$, $b(T_{v_i})=B_r(\ell-1)$, we obtain

$$B_r(\ell)=B_r(\ell-1)+k \Rightarrow B_r(\ell)=k\ell. \tag{2}$$

Our problem, however, is to broadcast from a leaf of a complete k-ary tree. Let, without loss of generality, the leftmost leaf be the source node, labeled u_0 in Fig. 2(a). Starting from v_0's only parent (labeled u_1) and following the upward edges we observe the following: below the ith node u_i in this path there are $k-1$ complete k-ary trees of height $i-1$. If u_i gets informed about the message then all the leaves in these $k-1$ subtrees can be informed in $B_r(i-1)+k-1$ steps, the earliest. Of course, u_i also has to inform its parent. In order to determine the exact broadcast time for v_0 we will start from the end of the upward path, calculate the minimum time needed for broadcasting from that node and then work our way down to u_0, using (1) at each step. We let $b(T_{u_i})$ be the broadcast time remaining after u_i receives the message.

Node u_h in the path is the root node of our original tree. It can only be informed through its leftmost child. When u_h receives the message it has to inform its $k-1$ remaining children as we mentioned above and for this $B_r(h-1)+k-1$ steps will be needed. Consequently, $b(T_{u_h})=B_r(h-1)+k-1$.

Next, consider u_{h-1}. This node has to inform its $k-1$ subtrees lying below plus its parent, u_h. For the subtrees below $B_r(h-2)+k-1$ steps are needed, while for broadcasting from u_h, $B_r(h-1)+k-1$ steps are necessary as we just saw. According to (1), the best plan is to send the message to u_h first and to the subtrees below next. 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 (2) $B_r(h-1) = k(h-1)$, 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.*

Fig. 2(b) shows an algorithm that achieves this lower bound; it is directly derived from the preceding analysis.

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 simple lower bound of $S(n) = G(n) \geq n-1$ steps. As a matter of fact the exact bound is n or $n+1$ steps (depending on k) and can be seen as follows. 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. Consequently 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 ($k=2$) 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 $k=2$.

Since a gathering algorithm can be had from a scattering algorithm (and vice versa) by simply reversing the data paths, we will only consider scattering here. The lower bound of n or $n+1$ steps can be achieved using the *furthest-first* scheduling discipline whereby the source node gives priority to messages that have to travel the furthest.

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

Proof outline: Assuming that leaf node 0 is the source node, let R_i be the routing node at level i which is the root of the subtree containing nodes 0 to k^i-1, as shown in Fig. 3. Let S_j be the subtree rooted at the jth child of R_i from the left (the source node belongs to S_0). The last message destined for any of the subtrees $S_1, S_2, \ldots, S_{k-1}$ will leave node 0 before any message destined for a leaf within S_0 does, according to the furthest-first regimen; that

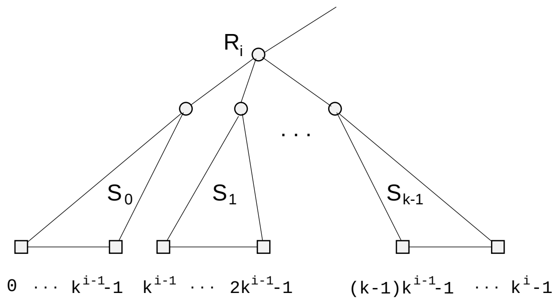


Figure 3:

is, it will leave at time $n - k^{i-1}$ and will arrive at its destination $2i - 1$ steps later, at time $T_i = n - k^{i-1} + 2i - 1$. The algorithm will thus finish at time $T = \max_{1 \le i \le h}\{T_i\}$. It can be seen that T_i is a decreasing function of i for $i \ge 2$. The maximum value of T_i for integer i can thus occur only for $i = 1$ or $i = 2$. Since $T_1 = n$ and $T_2 = n - k + 3$, we see that $T = n$ if $k \ge 3$ and $T = n + 1$ if $k = 2$, as claimed. □

2.3 Multinode broadcasting

In multinode broadcasting every node broadcasts its own message. If all nodes start broadcasting at the same time contention will be observed sooner or later at the routing nodes. An optimal multinode algorithm schedules the traffic in each routing node so that all leaves receive all messages at the minimum possible time. The algorithm we present next will be proven to be optimal. First, however, we derive the lower bound for multinode broadcasting algorithms.

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

Proof outline: Consider the k children v_i, $i = 1, 2, \ldots, k$, of the root node. Node v_1 is at level $h - 1$ and consequently has k^{h-1} leaves lying below. Each of the leaves will generate a broadcast message and this message must be sent through a child of v_1 to v_1; v_1 will send it to its remaining $k - 1$ children and to the root of the tree. Thus v_1 must make $kk^{h-1} = k^h$ transmissions. In addition, v_1 will receive all messages coming (through the root) from v_2, v_3, …, v_k. Each of these messages must be broadcast by v_1 to all its k children. This will result in v_1 making an extra $k(k-1)k^{h-1}$ transmissions. In total, the

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;

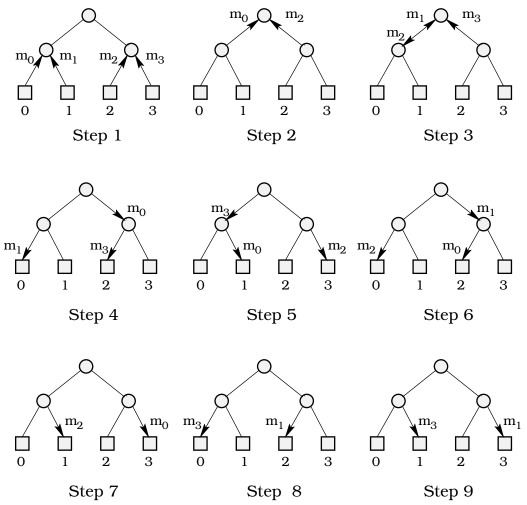
B Until the last message arrives from parent

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)

Figure 4: Multinode broadcasting under the single-port model (a) optimal algorithm (b) example in 4 leaves

number of transmissions by v_1 will be $k(k-1)k^{h-1} + k^h = k^{h+1} = kn$ and since under the single-port model it can only send one message at a time, node v_1 will be busy for at least kn steps.

The *first* message will arrive at v_i at time $h-1$ the earliest since the closest leaf is at distance $h-1$. The *last* message to leave v_i will be sent to one of its children. This child will have to broadcast this message to the subtree it roots, which is a complete k-ary tree with height $h-2$. Consequently, any multinode broadcasting algorithm needs time $MB(n) \geq kn + h - 1 + B_r(h-2) = kn + (k+1)(\log_k n - 2) + 1$. □

In Fig. 4(a) we give an algorithm for the multinode broadcasting problem and in Fig. 4(b) we demonstrate its operation for $n = 4$ leaves. The proof of the following theorem is given in [4].

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

2.4 Total exchange

Total exchange causes the densest form of contention on the network. Consider the subtree rooted at a routing node R_i in level i. A leaf u in this subtree will generate $n-1$ messages in total, out of which $n - k^{i-1}$ will pass through R_i as follows. There will be $n - k^i$ "upward"

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;

Figure 5: Optimal total exchange algorithm under the single-port model

messages, that is messages to be sent through the parent of R_i to a subtree other than the one R_i roots. There will also exist messages to be given from the subtree u lies in, to the other $k-1$ subtrees of R_i. Those messages will be $(k-1)k^{i-1}$ in number, giving a total of $n-k^{i-1}$ messages of u that go through R_i. 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.*

We now present an algorithm that achieves the lower bound of Theorem 5. Every leaf node transmits its $n-1$ messages under the *furthest-first* discipline as in the case of scattering and every routing node sends all possible messages to its parent before it starts sending any messages to its children. The algorithm is given in Fig. 5. Since leaves follow a furthest-first rule, "upward" messages arrive in routing nodes first, followed by messages destined to within the subtrees rooted at the routing nodes. (For the proof of the following see [4].)

Theorem 6 *The algorithm presented in Fig. 5 is optimal.*

Table 1: Time requirements for communications under the single-port model

Problem	Time
Broadcasting	$(k+1)\log_k n - 1$
Scattering/Gathering	n (if $k \geq 3$) , $n+1$ (if $k=2$)
Multinode Broadcasting	$kn + (k+1)(\log_k n - 2) + 1$
Total Exchange	$n^2(2k+1)(k-1)/k^3 + 2\log_k n - 3$

2.5 Summary of the single-port model

We summarized the results of the previous analyses in Table 1. It is interesting to compare the relative performance of different trees given that the number of leaves (n) is fixed. The broadcasting time is a decreasing function of k for $2 \leq k \leq 4$ and increases with k for $k \geq 4$; quaternary trees need thus the minimum time. For multinode broadcasting though, binary trees give the best performance as $MB(n)$ is an increasing function of k for $k \geq 2$. Finally, for the total exchange problem, ignoring the logarithmic term, it is seen that larger values of k are needed; at the extreme where $k = \sqrt{n}$ the network consists of only three levels and total exchange can be performed in $\Theta(n\sqrt{n})$ steps.

3 Communications under the multiport model

We will 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 since now the number of messages allowed to cross a link can be greater than one. In order to compare the performance of fat trees with that of the simple complete k-ary tree, we will assume that the processor (leaf) connections to/from the routing network are fixed, and consist of exactly one link. In other words, the capacity of level-1 branches will be $c_1 = 1$.

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, since the farthest node from a source is a leaf at distance $2h$, 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 to a routing node, forcing only one message to be send or received at a time by a leaf. Consequently, $S(n) = G(n) \geq n$ (or $n+1$ if $k=2$). Had we allowed $c_1 > 1$, other lower bounds could have been derived in an obvious way.

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 $k=2$. It is seen from this bound (and from the fact that the bound will be shown to be tight) that capacities within the routing network make no difference at least in terms of speed.

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

Proof outline: 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). Due to space limitations the reader is referred to [4] for the optimality proof of this method. □

This flooding algorithm achieves the lower bound but bares the cost of excessive queueing requirements, even for trees with increased capacities. In [4] we take a closer look at the queue sizes induced at each node; we also give an algorithm which is suboptimal by $2\log_k n - 2$ steps and which completely eliminates the queues.

3.3 Total exchange

Lower bounds for the total exchange problem can be derived by considering the messages that go through a level $h-1$ routing node. A particular node at this level will receive $k^{h-1}(n - k^{h-1}) = n^2(k-1)/k^2$ "upward" messages from the leaves of the subtree it roots; those messages will go through the other nodes in level $h-1$. If the branch capacities are c_h then only c_h messages can be transferred at a time from our node to the root of the tree, so that

$$TE(n) \geq n^2 \frac{k-1}{k^2 c_h} + 2\log_k n - 1. \tag{3}$$

For all $i = 0$ to $h - 1$

  /* Phase i */

  Do in parallel for all level $h - i$ nodes

    Transfer all messages from each of the k subtrees

    to the other $k - 1$ subtrees;

Figure 6: A total exchange algorithm with no contention

A simple total exchange algorithm that works for any capacity pattern and induces no queueing is derived 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. Iteratively, the algorithm can be stated as in Fig. 6.

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

$$T = \sum_{i=1}^{h} \left\lceil \frac{(k-1)k^{2i-2}}{c_i} \right\rceil + 2h - 1. \tag{4}$$

It is seen from (3) that the time needed for our algorithm is slightly suboptimal by a small constant factor. For the exponential capacities case in particular, the bound of (3) is not tight since it predicts $TE(n) \geq n - n/k + 2h - 2$. It should be clear that $TE(n) \geq n$ (or $TE(n) \geq n + 1$ if the tree is binary) is a tighter bound, since multinode broadcasting can be performed in at most as many steps as total exchange [2]. Using $c_i = k^{i-1}$, from (4) we see 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. In the next section we show that actually there exists a tighter lower bound for trees with exponential capacities.

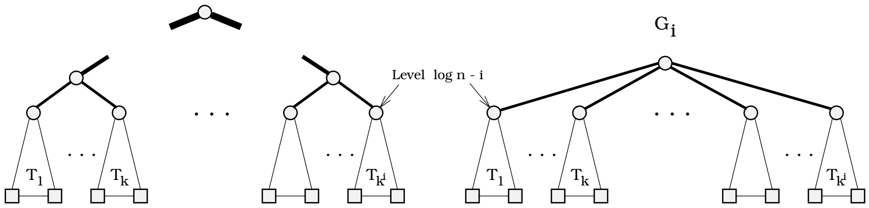


Figure 7: Collapsing the last i levels of the tree gives graph G_i

3.3.1 A tighter bound for exponential capacities trees

We show here 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.

If at any given step of a total exchange algorithm a leaf did not receive a message we say that a *hole* occurred in its receptions. Notice that since each leaf must receive $n-1$ messages, if the average number of holes per leaf node was w then the algorithm takes time $T \geq n-1+w$; the equality holds if all leaves had the same number of holes (w).

Consider the graph G_i which is derived from our original tree by collapsing its last i levels into a single vertex (see Fig. 7). We then have a collection of k^i subtrees $T_1, T_2, \ldots, T_{k^i}$ each having n/k^i leaves and their roots have a single common parent. The height of the new graph is $h-i+1$. In addition we introduce new edges between every pair of leaves within each of the k^i subtrees so that messages within a subtree can be transferred in a single step. In order to avoid those edges begin used simultaneously, we also impose the restriction that *only one message may be received by a leaf at any time* as in the original tree. It is seen that any total exchange algorithm for the original tree is a total exchange algorithm for G_i (but not vice versa) — the message transfers within the last i levels of our tree are substituted by no-ops in G_i and the additional edges in G_i are not utilized. As a result, the graph G_i can be used to derive a lower bound on total exchange algorithms for our original tree.

A message will be called *internal* to subtree T_j if both the source and the destination lie in T_j otherwise the message is *external*. The crucial observation is that holes will start appearing in G_i after the first external message is dispatched. Assume that T_1 is the one to send the first e external messages at time t. Then at time $t+1$ there will be e holes in the leaves of T_0. In fact, due to the $2(h-i+1)$-step delay external messages suffer before reaching their destinations, no messages will have arrived from any other subtree before time $t + 2(h - i + 1)$. Consequently the number of holes in T_0 between times t and $t + 2(h - i + 1)$ will be equal to the total number of external messages it sent during this period.

We will now find the minimum number of holes that will occur in every leaf in G_i. Consider any total exchange algorithm for G_i and partition time in $2(h-i+1)$-step periods. If there are p such periods (plus possibly a last, shorter one), the algorithm needs time $T \geq 2p(h - i + 1)$. Assume that at the sth period there were $w_s^{(j)}$ holes in T_j. Hence in the first period the total number of holes was $w_1 = w_1^{(1)} + w_1^{(2)} + \cdots + w_1^{(k^i)}$. This is exactly the number of external messages sent (in total) during the first period. During the second period the total number of holes was $w_2 = w_2^{(1)} + w_2^{(2)} + \cdots + w_2^{(k^i)}$ and as a result the total number of external messages sent during the second period was *at most* equal to $w_1 + w_2$. In general, during the jth period the maximum number of external messages was $w_1 + w_2 + \cdots + w_j$. Hence after p periods the total number of external messages observed was at most

$$\sum_{j=1}^{p}(w_1 + \cdots + w_j) = \sum_{j=1}^{p}(p - j + 1)w_j = p\sum_{j=1}^{p} w_j - \sum_{j=1}^{p}(j-1)w_j = pW - Q$$

where $W = \sum w_j$ is the total number of holes and Q is a non-negative term.

Each leaf node must sent in total $n - 1$ messages out of which the $n - n/k^i$ will be external, so that the total number of external messages will be $n(n - n/k^i)$. This means that $pW \geq n(n - n/k^i)$. Notice that the average number of holes per node is $w = W/n$ and since $T \geq 2p(h - i + 1)$, the last inequality yields

$$wT \geq 2n(1 - \frac{1}{k^i})(h - i + 1). \qquad (5)$$

From our earlier discussion, if w is the average number of holes per leaf then

$$T \geq n - 1 + w, \qquad (6)$$

the equality holding if all leaves have exactly w holes. Inequalities (5) and (6) can be combined to determine the minimum value of T. Since (5) holds for any value of i, we can choose i so that we obtain the best bound on T. For example if $i = 1$, it can be seen that T cannot be less than $n + h - 1$. A tighter bound can be had if we select $i = \log_k h + 1$; after a bit of algebra, we obtain from (5):

$$wT \geq 2n(1 - \frac{1}{kh})(h - \log_k h) \geq 2nh - 2n\log_k h - n. \qquad (7)$$

Table 2: Time requirements for communications under the multiport model

Problem	Time
Broadcasting	$2\log_k n - 1$
Scattering/Gathering	n (if $k \geq 3$) , $n+1$ (if $k=2$)
Multinode Broadcasting	n (if $k \geq 3$) , $n+1$ (if $k=2$)
Total Exchange*	$\geq n^2(k-1)/(k^2 c_h) + 2\log_k n - 1$

*see text

If $w < q = 2h - 2\log_k h - 1$, then (7) gives

$$T \geq \frac{2nh - 2n\log_k h - n}{2h - 2\log_k h - 2} = n + \frac{n}{2h - 2\log_k h - 2}.$$

Noting that $h = \log_k n$, and after a bit of algebra we obtain $T > n - 1 + q$. On the other hand, if $w > q$ then from (6) $T \geq n - 1 + w > n - 1 + q$. Consequently, the minimum time can be had only for $w = q$ and in that case

$$T = n - 1 + q = n + 2h - 2\log_k h - 2 \text{ steps.} \tag{8}$$

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.*

3.4 Summary of the multiport model

The results for the multiport model are summarized in Table 2. The preceding analyses were based on the assumption that $c_1 = 1$. It is expected that some of the problems will require less steps if the first level branches consisted of more than one links; it would be interesting to see what is the exact effect of c_1 in this case. For the total exchange problem, the bound given in Table 2 is general but provably not tight in some cases. What we have shown here is that for trees with exponentially growing capacities, total exchange can be performed in time $n + 2\log_k n - 2\log_k \log_k n - 2 \leq TE(n) \leq n + 2\log_k n - 2$.

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, C. Ozveren, G.D. Stamoulis, P. Tseng and J.N. Tsitsiklis, "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, G. Pucci, A. Ranade and A. L. Rosenberg, "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, J. Hromkovic, S. Madhavapeddy, B. Monien and P. Mysliwietz, "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, S. T. Hedetniemi and A. L. Liestman, "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, No. 9, 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.