

The Polling Primitive for Computer Networks

A. Czygrinow*, M. Karoński† V. S. Sunderam‡

September 22, 1999

Abstract

We describe a distributed computing primitive termed polling that is both a means of synchronization and communication in distributed or concurrent systems. The polling operation involves the collection of messages from nodes in an interconnection network, in response to a query. We define the semantics of polling, and present algorithms for implementing the operation on complete and hypercube networks. Time and message lower bounds are presented, and it followed by an analysis of the number of operations performed at each node for every algorithm. We show that polling in a complete graph on 2^n vertices can be completed in $2n$ rounds using $2^n + 2^{n-3} + \lceil \frac{2^{n-3}+1}{3} \rceil - 2$ messages. In case of n -cube, we show that polling in $2n$ rounds requires $\lceil 2^n + \frac{1}{3}2^{n-1} + \frac{1}{4}\sqrt{2^n} - \frac{4}{3} \rceil$ messages and we present an algorithm that completes polling in $2n$ rounds and sends $2^n + 3 \cdot 2^{n-4} - 1$ messages.

1 Introduction

We define the *polling* operation on interconnection networks as follows: One processor in a network (termed the root) has a “question” that must be asked of all

*Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322

†Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, and Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Poznan, Poland. Research partially supported by grant KBN 2 PO3A 023 09 and by NSF grant INT-9406971.

‡Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322

other processors, each of which must respond with an “answer”. We wish to perform this operation in minimal time using a minimal number of messages, under the following assumptions:

(1) Processors communicate solely by message-passing; messages may contain the question, one or more answers, or both.

(2) Polling proceeds in “rounds”; a processor may either send or receive at most one message during any round. Processors other than the root may participate only after they have received at least one message (i.e. a message containing the question).

As with most distributed algorithms, the efficiency of polling is measured by the time required to accomplish the operation (number of rounds), and by the number of messages needed. Algorithm goals are to minimize the number of rounds necessary, and, given the minimal number of rounds, to minimize the number of messages.

The notion of polling is traditionally associated with terminals and controllers, and with data link protocols. However, polling has important applications in distributed systems, and on non-shared memory multiprocessor machines. Status or resource monitoring, fully replicated queries or updates, the computation of multiple-input functions, and certain synchronization primitives may all be implemented using polling. Polling inherently requires that one processor initiates the operation, that every processor participates, and that all outputs be returned to the initiator. When these are necessary conditions within an application, polling is an effective distributed computing primitive. It seems that until recently, the polling problem received much less attention than the classical broadcasting and gossiping problems for which many results were obtained and different models were studied (see for example [6] and references in it). The time complexity of polling was recently studied by A. Rescigno in [8] and [9]. The communication model considered is however different than ours. For example, in model in [8] and [9] a node can send a message to all of its neighbors in a single round but it is not possible to send many responses along a single edge. In our model, a vertex can communicate only with one of its neighbors in a single round. On the other hand, we assume that responses can be combined and send as one. Consequently, under the assumptions in [9] one can easily see that the polling in a complete graph on n vertices can be done in 2 rounds, in our model it requires about $2 \log n$ rounds. The algorithms of [9] are based on special kind of polling trees which are used to distribute the question and to gather the responses. In contrast, our communication graphs are not trees as it is easy to see that if we use any spanning tree as

communication graph then the number of messages sent will be greater than what is obtained using our graphs. The number of messages sent is not discussed in Resigno's papers.

The paper presents the lower bounds and algorithms for polling in networks with complete graph and hypercube topology. In the next section we show the lower bounds and we present an algorithm that performs polling in complete networks in the minimal number of rounds and using the minimal number of messages. Section 3 contains an analysis of polling in the hypercube network. We show a slightly better lower bound for the number of messages and propose a nearly optimal algorithm.

2 Preliminaries

We define a *path* in a network as a sequence of vertices $v_1v_2\dots v_n$ such that for all $1 \leq i \leq n - 1$ there is an edge v_i, v_{i+1} . We then say that the path *covers* n vertices or if v_1 and v_n are already nodes of a different path we say that the path covers $n - 2$ new vertices. *Cycle* is defined as a sequence $v_1v_2\dots v_n$ such that for $1 \leq i \leq n$ there is an edge $v_i, v_{(i+1) \pmod n}$. The *degree of a vertex* is the number of vertices incident to it. Also N will denote the number of nodes in a network, $\lg a$ will denote the logarithm of base 2 and $\ln a$ the logarithm of base e .

We define *partial broadcast* as the delivery of a message originating at the root to a subset of nodes of a network. *Partial gather* is defined analogously as the collection of messages from a subset of nodes of a network.

Fact 1 For $0 \leq M \leq 2^n - 1$ *partial broadcast to* M (*gather from* M) *nodes requires* $\lceil \lg(M + 1) \rceil$ *rounds.*

Proof. During one round a node may either send or receive one message, so the number of nodes that have received the message can at most double in each round, which completes the proof.

□

3 The Algorithm for complete graphs

In this section we present the lower bound for the number of rounds and the number of messages. The technique used in the proof of second bound leads to an optimal polling algorithm in complete graphs which is described at the end of the section.

Proposition 2 Let $N = 2^n - k$ where $0 \leq k \leq 2^{n-1}$.

- (i) If $k = 0$ then the number of rounds is at least $2n$.
- (ii) If $0 < k \leq 2^{n-2}$ then the number of rounds is at least $2n - 1$.
- (iii) If $k \geq 2^{n-2} + 1$ then the number of rounds in is at least $2n - 2$.

Proof. (i) Let $N = 2^n$. From Fact 1 we know that immediately after $n - 1$ round at most $2^{n-1} - 1$ nodes other than root received the message originated at the root. Denote the set of these nodes by L and let $R = V(K_n) \setminus L$. To gather messages from R we need at least $\lceil \lg(|R| - 1) \rceil = \lceil \lg(2^{n-1} + 1) \rceil = n$ rounds. At least one more round is necessary to initiate the participation of nodes in R . This shows that to complete polling in K_N , at least $n - 1 + 1 + n = 2n$ rounds are necessary. (ii) When $k \leq 2^{n-2}$ then $N \geq 2^n - 2^{n-2} = 2^{n-1} + 2^{n-2}$ and Fact 1 implies that immediately after $n - 1$ rounds at most $2^{n-1} - 1$ nodes other than root will receive the message originated at the root. To gather from remaining 2^{n-2} nodes, we need $\lceil \lg(2^{n-2} + 1) \rceil$ rounds and at least one round to initiate the participation of remaining nodes. Therefore, at least $n - 1 + 1 + n - 1 = 2n - 1$ rounds are required to complete polling in K_N . (iii) It follows from (i) as $N \geq 2^n - 2^{n-1} = 2^{n-1}$.

□

Proposition 3 The polling in a complete graph K_{2^n} in $2n$ rounds requires $2^n + 2^{n-3} + \lceil \frac{2^{n-3} + 1}{3} \rceil - 2 \geq 2^n + \frac{1}{3}2^{n-1} - 1$ messages.

Proof. Consider the communication graph constructed by a polling algorithm. To minimize the number of messages the polling algorithm must construct the communication graph with as few cycles (or paths) as possible. Indeed, the optimal situation (minimal number of messages) would be in case of Hamiltonian cycle, but then we need 2^n rounds to complete the polling. Since communication must be completed in $2n$ rounds, the longest cycle in the graph contains $2n$ vertices. Every additional cycle or path in the graph increases the number of messages from the optimal case - cycle by one as it is illustrated in Figure 2. To minimize the number of cycles (paths) used in the communication graph the algorithm must use as many long cycles or paths as possible. We see that the longest cycle can cover $2n$ vertices and there can be only one cycle of this length. By induction, we can have at most

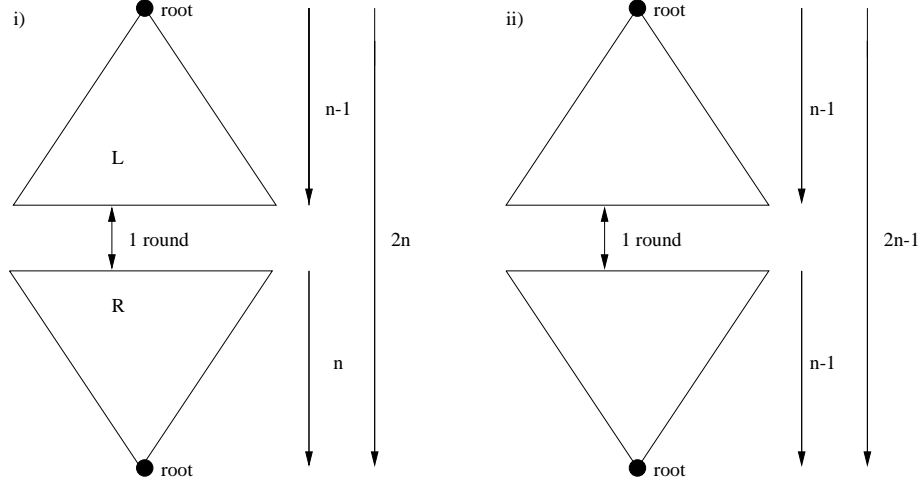


Figure 1: Proof of Proposition 2

2^i cycles or paths that cover $2n - (2i + 3)$ vertices. Therefore, the number of vertices covered by paths and cycles initiated in rounds 1 to $s + 1$ is at most

$$F(s) = 2n + \sum_{i=0}^s (2n - (2i + 3))2^i = 2^{s+1}(2n - 2s - 1) - 1.$$

For $s = n - 4$ we have $F(n - 4) = 2^n - 2^{n-3} - 1 < 2^n$ and so we need at least $r = \lceil \frac{2^{n-3} + 1}{3} \rceil$ additional paths that cover three vertices each. The total number of paths and cycles in the communication graph is at least $1 + r + \sum_{i=0}^{n-4} 2^i = r + 2^{n-3}$. We infer that the number of messages needed to complete polling in $2n$ rounds is at least $2^n + 2^{n-3} + r - 1$.

□

Next we present the algorithm that completes polling in K_{2n} in $2n$ rounds and that uses $2^n + 2^{n-3} + r - 1$ messages. The idea is as follows. In the first $n - 3$ rounds the greedy procedure is invoked that results in total of $2^n - 2^{n-3} - 1$ nodes covered. The remaining vertices are covered using $\lceil \frac{2^{n-3} + 1}{3} \rceil$ paths. More formally, let us define the broadcasting tree of height n , $B(n)$ as follows: $B(0)$ contains just one vertex- the root, for $n > 0$, $B(n)$ is obtained from $B(n - 1)$ by adding for each vertex $v \in V(B(n - 1))$ exactly one vertex v' and an edge vv' . Note that the vertices of the tree can be grouped into levels, where the i th level contains the

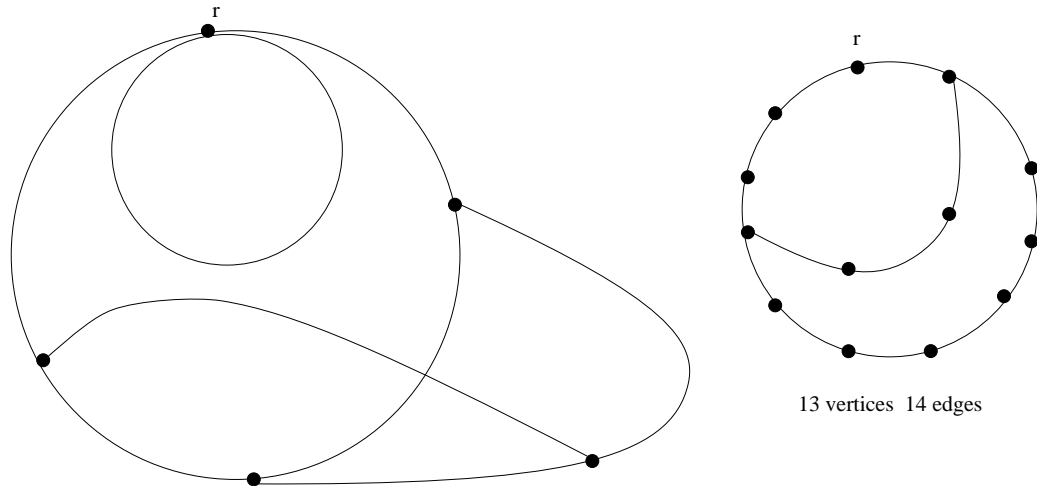


Figure 2: Proof of Proposition 3

vertices that are at distance i from the root. Then the communication graph can be constructed by the following procedure.

Algorithm

1. Take two copies of the broadcast tree $B(n-2)$ of height $n-2$ and match the corresponding leaves with paths of length four. The resulting graph looks as in Figure 3.
2. Let $r = \lceil \frac{2^{n-3}+1}{3} \rceil$. To simplify the exposition we assume that $r = \frac{2^{n-3}+1}{3}$. Add r paths that cover three vertices each to the graph constructed in the first step. Each path should start at the node on any level smaller than $n-1$, and should end in the corresponding vertex of the second copy of $B(n-2)$.
3. Map the constructed graph such that the roots of both copies of $B(n-2)$ are mapped to the single vertex- the root of the network.

We observe that the graph constructed in the first step of the algorithm contains exactly $F(n-4)$ vertices. We add r paths in the second step which gives total

$$F(n-4) + 3 \cdot r = 7 \cdot 2^{n-3} + 3 \frac{2^{n-3} + 1}{3} = 2^n + 1$$

It is easy to see that we can add r paths to the communication graph constructed in the first step such that the requirements of the communication are satisfied. Indeed, since there are 2^{n-3} vertices on levels 1 through $n-2$, it is possible to add r paths not using the vertices on the $(n-1)$ th level.

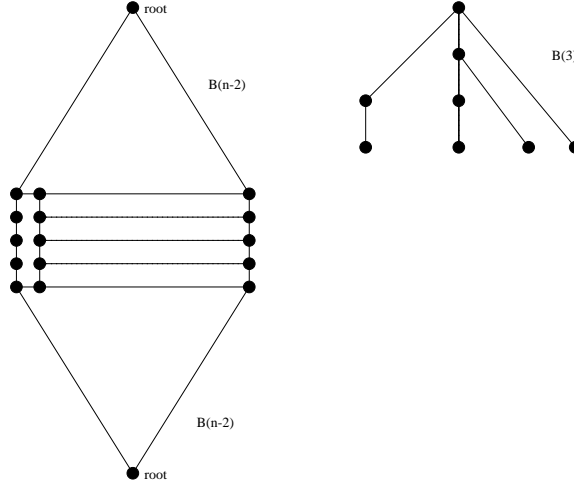


Figure 3: Construction of communication graph

The total number of vertices in the constructed communication graph is $F(n-4)+3r-1 = 2^n$, the number of edges is equal to $2(2^{n-2}-1)+4\cdot 2^{n-3}+4r = 2^n+4r-2$.

Example 4 If $N = 2^5$ then $r = 2$ and the communication graph is presented in Figure 4.

4 Analysis for the n -cube

In this section we discuss polling primitive in hypercube networks. The n -cube Q_n is the graph (V, E) such that $V = \{(i_1, i_2, \dots, i_n) : i_k \in \{0, 1\}\}$ and two vertices (i_1, i_2, \dots, i_n) and (j_1, j_2, \dots, j_n) are joined by an edge if and only if there is exactly one k such that i_k and j_k are different. The number of vertices in the n -cube is 2^n and the number of edges is $n2^{n-1}$. To improve the lower bound from previous section, we consider the following graph. Take $K_{2^{n-1}}$ where $V(K_{2^{n-1}}) = \{1, \dots, 2^n - 1\}$ and let $(x + K_{2^{n-1}})_n$ be the graph obtained from $K_{2^{n-1}}$ by adding vertex $root$ and edges between x and i for $i = 1, \dots, n$, i.e. with vertex set $V(K_{2^{n-1}}) \cup \{x\}$ and edge set $E(K_{2^{n-1}}) \cup \{\{x, 1\}, \{x, 2\}, \dots, \{x, n\}\}$.

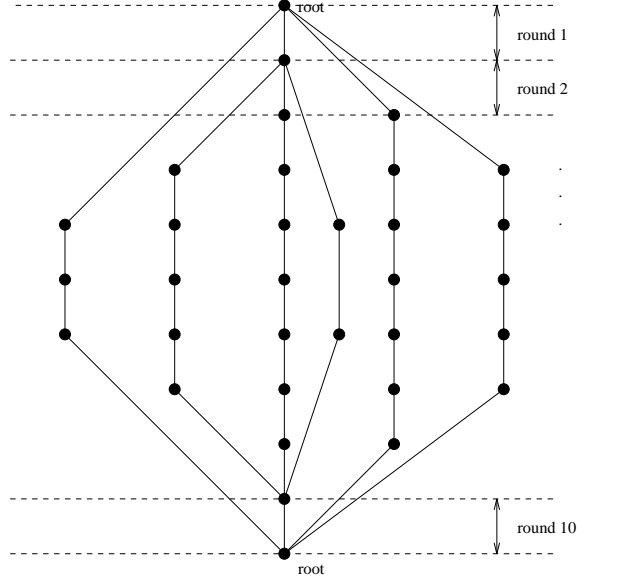


Figure 4: Polling in the K_5

Proposition 5 *Let n be an even number. Polling in $2n$ rounds in $(x + K_{2^n-1})_n$ with the root x requires $\lceil 2^n + \frac{1}{3}2^{n-1} + \frac{1}{4}\sqrt{2^n} - \frac{4}{3} \rceil$ messages.*

Proof. In the process of polling the algorithm constructs a layered graph with the k th level containing the vertices initiated in the k th round. It is convenient for our discussion to view the root x as a pair of vertices (x_1, x_2) , where x_1 sends the messages, x_2 receives the messages. Denote by $G_1(l)$ a graph with vertex set consisting of levels 0 through l and all the edges in the communication graph between these levels, by $G_2(l)$ graph with vertex set consisting of levels l through $2n$ and all the edges in the communication graph between these levels. It is easy to see that in the scheme in which the minimal number of messages is sent, we can find an l such that both graphs $G_1(l)$ $G_2(l)$ are trees (see Figure 5).

Denote by h_1 (h_2) the height of G_1 (G_2) by d_1 , (d_2) the degree of x_1 (x_2). To send the minimal number of messages the algorithm must enforce as many long cycles or paths as possible. Thus we can assume that x sends and receives a total of n messages, i.e. $d_1 + d_2 = n$. The number of paths or cycles initiated in rounds $1, \dots, s_1 + 1$ is at most

$$f = 1 + \sum_{i=0}^{s_1} 2^i - 1 - \sum_{i=d_1}^{s_1} 2^{i-d_1} = 2^{s_1+1}(1 - 2^{-d_1}).$$

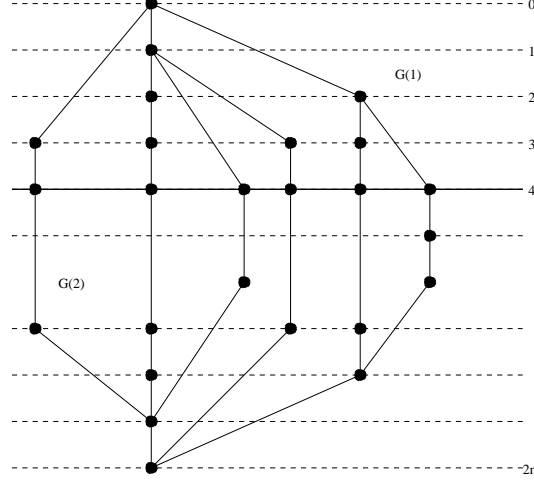


Figure 5: Construction of communication graph

Note that f is equal to the number of leaves of $G_1(l)$ and so f is equal to the number of leaves of $G_2(l)$. Since the communication in G_2 can be viewed as reversed polling procedure, we also have

$$f = 2^{s_2+1}(1 - 2^{-d_2}).$$

The number of vertices of G_1 is at most

$$(h_1 + 1) + \sum_{i=0}^{s_1} (h_1 - i - 1)2^i - (h_1 - d_1) - \sum_{i=d_1}^{s_1} 2^{i-d_1} =$$

$$d_1 + 1 + (h_1 - 1)(2^{s_1+1} - 1) - 2(s_1 2^{s_1+1} - (s_1 + 1)2^{s_1} + 1) - (h_1 - d_1 - 1)(2^{s_1-d_1+1} - 1) +$$

$$2((s_1 - d_1)2^{s_1-d_1+1} - (s_1 - d_1 + 1)2^{s_1-d_1} + 1) = 1 + (h_1 - s_1)(2^{s_1+1} - 2^{s_1-d_1+1})$$

On the other hand, by a similar argument, the number of vertices of G_2 is at most $1 + (h_2 - s_2)(2^{s_2+1} - 2^{s_2-d_2+1})$. Therefore, the number of vertices in $G_1(l)$ and $G_2(l)$ is at most

$$2 + (h_1 - s_1)f + (h_2 - s_2)f - f = 2 + (2n - (s_1 + s_2) - 1)f$$

On the other hand the total number of vertices in $G_1(l)$ and $G_2(l)$ is $|V(G_1(l))| + |V(G_2(l))| = 2^n + 1$. Since $s_1 = \lg f - \lg(1 - 2^{-d_1}) - 1$ and $s_2 = \lg f - \lg(1 - 2^{-d_2}) - 1$ we see that

$$2^n - 1 = (2n - 2 \lg f + \lg(1 - 2^{-d_1}) + \lg(1 - 2^{-d_2}) + 1)f \quad (1)$$

To satisfy equation (1) with the minimum number of messages sent (with minimum f) we must maximize $\lg(1 - 2^{-d_1}) + \lg(1 - 2^{-d_2})$, subjected to $d_1 + d_2 = n$. This is maximized for $d_1 = d_2 = \frac{n}{2}$ which implies $s_1 = s_2$. Thus, the total number of vertices covered by paths and cycles initiated in first $s + 1$ rounds is at most

$$F(s) = 1 + (2n - 2s - 1)(2^{s+1} - 2^{s-\frac{n}{2}+1})$$

We see that $F(n - 4) = 1 + 7(2^{n-3} - 2^{\frac{n}{2}-3}) < 2^n$, and so we need at least $r = \lceil \frac{1}{3}(2^{n-3} + 7 \cdot 2^{\frac{n}{2}-3} - 1) \rceil$ additional paths, each one covering three vertices. Therefore, the number of leaves in $G_1(l)$ is at least $r + f$ which shows that the number of messages used to complete polling is at least

$$2^n - 1 + f + r \geq 2^n + \frac{1}{3}2^{n-1} + \frac{1}{4}\sqrt{2^n} - \frac{4}{3}$$

□

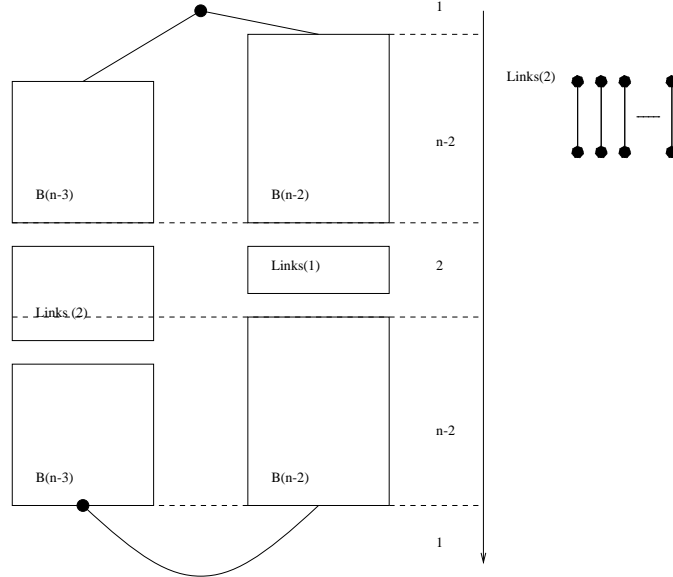


Figure 6: The communication graph for the n -cube

Since Q_n is a subgraph of $(x + K_{2^n-1})_n$ we have the following

Corollary 6 *Let n be an even number. Polling in $2n$ rounds in n -dimensional hypercube requires $\lceil 2^n + \frac{1}{3}2^{n-1} + \frac{1}{4}\sqrt{2^n} - \frac{4}{3} \rceil$ messages.*

Next, we describe an algorithm that can be used to perform polling in an n -cube. In the communication graph we will again make use of $B(n)$ trees described in Section 2. The algorithm uses the communication graph from Figure 6. The n -cube embedding of this graph is illustrated in Figure 7. One can easily see that all of groups A, B, C, L, F, G can be linked together if the following rules are observed:

- F consists of two layers: $F1$ contains $x \dots x1110$, $F2$ contains $x \dots x1010$.
- L is a broadcasting tree of height $n - 3$ with leaves being $x \dots x0110$, which can be easily constructed.
- G is a broadcasting tree of height $n - 3$ with leaves from $x \dots x1000$.
- A is a broadcasting tree of height $n - 2$ with leaves from $x \dots x011$.
- B has only one layer: $x \dots x111$.
- C is a broadcasting tree of height $n - 2$ with leaves from $x \dots x101$

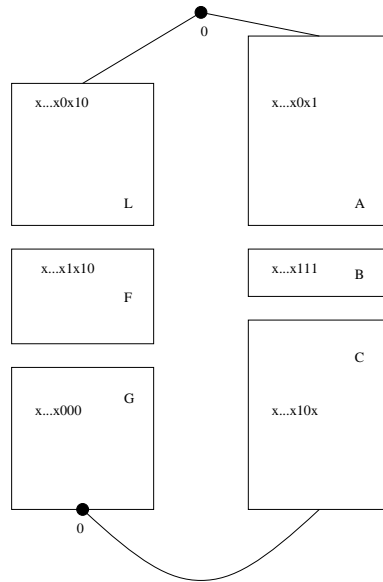


Figure 7: Embedding

Example 7 In case $n = 5$, the communication scheme is as in Figure 8.

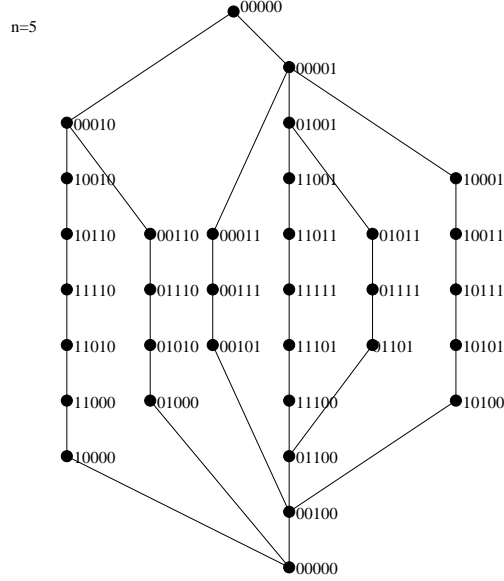


Figure 8: The communication scheme for 5-cube

Proposition 8 *The algorithm terminates in $2n$ rounds and sends $2^n + 3 \cdot 2^{n-4} - 1$ messages.*

Proof. Tree $B(k)$ contains $2^k - 1$ edges and 2^{k-1} leaves, and so the total number of edges used is

$$2(2^{n-2} - 1) + 2 \cdot 2^{n-3} + 2(2^{n-3} - 1) + 3 \cdot 2^{n-4} = 2^n + 3 \cdot 2^{n-4} - 1.$$

□

5 Summary

We studied the polling problem in networks with complete graph and hypercube topologies. In both cases, we proved lower bounds for the number of rounds and messages needed to complete the polling operation. We also presented algorithms that complete polling in the minimal number of rounds.

References

- [1] D. Agarwal, E. Abbadi, “An Efficient Solution to the Distributed Mutual Exclusion Problem”, Proc. 9th Symposium on the Principles of Distributed Computing, 1989.
- [2] L. Bomans, D. Roose, “Benchmarking the iPSC/2 Hypercube Multiprocessor”, Concurrency: Practice and Experience, Vol. 1, pp. 3-18, 1989.
- [3] T. Chan and Y. Saad, “Multigrid Algorithms on the Hypercube Multiprocessors”, IEEE Transactions on Computers, Vol. 35, pp. 969-977, 1986.
- [4] A. Czygrinow, M. Karoński, V. S. Sunderam, “The Polling Primitive for Hypercube Networks”, Proc. 7th IEEE Symposium on Parallel and Distributed Processing, 1995.
- [5] T. H. Dunigan, “Performance of the Intel iP-SC/860 Hypercube”, Oak Ridge National Laboratory, Technical Report TM-11491, 1990.
- [6] J. Hromkovic, R. Klasing, B. Monien, R. Peine, “Dissemination of information in interconnection networks (Broadcasting & Gossiping)”, *Combinatorial Network Theory* (D. Du and D. Hsu Eds.), pp. 125-212, Kluwer Academic Publishers 1990.
- [7] Y. Lan, A. Esfahanian, and L. M. Ni, “Multicasting in Hypercube Multiprocessors”, Journal of Parallel and Distributed Computing, Vol 8, pp. 30-41, 1990.
- [8] A. Rescigno, “On the communication complexity of polling”, Information Processing Letters 59, pp. 317-323, 1996.
- [9] A. Rescigno, “Optimal Polling in Communication Networks”, IEEE Transactions on Parallel and Distributed Systems, Vol. 8, No. 5, pp. 449- 461, 1997.
- [10] Y. Saad and M. Schultz, “Topological Properties of Hypercubes”, IEEE Transactions on Computers, Vol. 37, pp. 867-872. 1988.