Computer Science
Technical Report

Colorado
State
University

Near-optimal Path-based Wormhole Broadcast in Hypercubes[*]

Shahram Latifi[†], **Myung Hoon Lee**[‡] **and Pradip K. Srimani**[§]

Technical Report CS-97-111

Computer Science Department
Colorado State University
Fort Collins, CO 80523-1873

Phone: (970) 491-5792 Fax: (970) 491-2466
WWW: http://www.cs.colostate.edu

[*] To appear in the Proceedings of **ISCA 10th INTERNATIONAL CONFERENCE ON PARALLEL & DISTRIBUTED COMPUTING SYSTEMS (PDCS),** October 1-3, 1997, New Orleans, Louisiana

[†] Department of Electrical Engineering, University of Nevada, Las Vegas, NV 89154-4026

[‡] Department of Electrical Engineering, University of Nevada, Las Vegas, NV 89154-4026

[§] Department of Computer Science, Colorado State University, Ft. Collins, CO 80523

Near-optimal Path-based Wormhole Broadcast in Hypercubes*

Shahram Latifi[†], **Myung Hoon Lee**[‡] **and Pradip K. Srimani**[§]

Abstract

We consider the problem of broadcasting a message in the n-cube, Q_n, equipped with wormhole switching. The communication model assumed is one-port, and the broadcasting scheme is *path-based* whereby, during broadcasting along a path by a node, all the nodes on that path will receive the message. The wormhole path length is m where $1 < m \leq n$, and thus this is a generalization of an earlier work which considered a path length of n. First, a method is proposed which is based on recursively partitioning the cube to subcubes of dimension m, and then calling the previously developed algorithm on such Q_m's concurrently (cube-based broadcast). The second method is based on the concept of *Gray codes (GCs)*, and at every given step, it forms the Hamiltonian path of appropriate size as the broadcast path (GC-based broadcast). It is shown that the steps required in GC-based broadcast is fewer than or equal to those needed by cube-based broadcast. Furthermore, comparison of time complexity of GC-based broadcast to the lower bound reveals that this algorithm is near-optimal, and in fact optimal in many cases. This work improves on the best algorithm developed for path-based broadcast in one-port hypercube both in complexity and in simplicity.

1 Introduction

Many applications running on multiprocessor networks call for casting information from one processor to others in minimal time. This communication primitive, normally referred to as broadcasting, is usually employed for control (synchronization, initialization, diagnosis) or algorithm execution (matrix operations).

Among the networks designed for parallel processing, hypercubes have acquired wide acceptance in academic and industrial communities. Owing to their node-symmetry, simplicity of routing, hierarchy and fault tolerance, hypercubes are employed as the underlying topology of commercial machines such as n-Cube, CM5, and iPSC/860 to name a few.

The first generation multiprocessors used $Store-and-Forward\ Routing$ scheme which involves routing packets of data from source to destination through intermediate nodes. No forwarding of a packet takes place by an intermediate node unless the packet is received in its entirety by the node under question. The latency in store-and-forward networks is directly proportional to the distance between source and destination. By dividing the packet into smaller units of data (called flits) and sending flits through intermediate nodes in a pipelined fashion, the communication latency can be made almost insensitive to distance between source and destination [1]. This scheme, known as wormhole routing, is implemented in modern multiprocessors. Reference [2] provides a good survey of this topic.

Common wormhole-based networks will route data only to their final destination. In other words, the routers attached to intermediate nodes can only forward data without copying it. It is, however, possible to add flit-copying feature to intermediate nodes whereby these nodes can copy the flits as they pass by. This model is called *path-based routing* and has been adopted by researchers to develop efficient multicast and broadcast algorithms for meshes and hypercubes [3, 4, 5].

Ho et. al. developed efficient broadcasting algorithm in hypercube based on all-port [6, 7], and one-port [4] communication whereby a path is extended from source to destination in one routing step, and during this step all the intermediate nodes along the path will receive the flits. This assumption clearly poses some burden on the complexity of the routers as they should

[*] To appear in the Proceedings of **ISCA 10th INTERNATIONAL CONFERENCE ON PARALLEL & DISTRIBUTED COMPUTING SYSTEMS (PDCS)**, October 1-3, 1997, New Orleans, Louisiana

[†] Department of Electrical Engineering, University of Nevada, Las Vegas, NV 89154-4026

[‡] Department of Electrical Engineering, University of Nevada, Las Vegas, NV 89154-4026

[§] Department of Computer Science, Colorado State University, Ft. Collins, CO 80523 perform copying as well as forwarding the flit.

To limit the cumulative delay incurred at each intermediate node during the reception process, a maximum should be posed on the length of the wormhole path (w-path for short). The maximum w-path length is typically a function of such parameters as: start up latency at the source and destination, the delay on the intermediate routers encountered on the path (due to copying and forwarding), the length of the message, and the per-flit transmission time. Loosely speaking, this maximum length corresponds to a case where transmission time can still be regarded as "insensitive" to distance. We assume the one-port communication model whereby each node can communicate with only one of its neighbors at any given step of the broadcast algorithm.

In their study on hypercubes, Ho et. al [4] have chosen the diameter as the limit. The diameter limit combined with e-cube routing (routing dimensions in ascending order) avoids any deadlock. We generalize this assumption to the case where a w-path can be of length m, $1 < m \leq n$. This assumption will make the router complexity dependent on some fixed value m, and independent of n, the network size.

The rest of the paper is organized as follows. Section 2 describes the communication model and presents the lower bound on broadcasting. In Section 3, two algorithms namely cube-based and Gray code-based are presented and compared. Section 4 summarizes the work and offers some intriguing remarks.

2 Communication Model

In order to describe the wormhole broadcast algorithm in a binary n-cube Q_n, and to evaluate its performance we need to specify the communication model.

Consider that a source node in the network wants to send an m-flit message to another node y which is Δ hops away in the network (Δ is the number of links to be traversed by the message as determined by the routing algorithm chosen. If wormhole routing is chosen, the total routing time or the so-called communication complexity will be given by $s + s'(\Delta - 1) + m\tau$, where s denotes the start-up latency at the source node, s' denotes the start-up latency at each of the intermediate nodes before the message reaches its destination, and τ denotes the transmission time per flit [2]. This is under the assumption that there is no *link congestion*, i.e., no two messages contend for the same communication channel at the same time (note that when there is link congestion modeling communication complexity for wormhole routing becomes the same as in circuit switched routing). The efficiency of the wormhole routing derives from the fact that $s \gg s'$ and that when m is large and Δ is small, the second term $s'(\Delta - 1)$ becomes negligiblly small – in other words, if the message travels a relatively small distance, the communication delay in wormhole routing becomes *insensitive* to the distance [8]. Experimental results are available [9] to verify this in case of hypercube networks; for an Intel iPSC/2 hypercube, it was measured that $s = 0.7\ ms$, $s' = 60\mu s$, and $\tau = 0.36\mu s$. There are several factors that need be carefully considered to decide on the assumptions in the communication model and thereafter to evaluate the performance of the communication primitives.

2.1 Single Port vs. Multi Port

The communication latency in wormhole broadcast depends on the capability of the nodes to send/receive messages. In the single-port model, each node can send or receive a message along only one port whereas in the all-port model the infomration exchange can take place along all the available ports concurrently.

2.2 Distance Insensitivity in Wormhole Routing

The insensitivity of point to point wormhole routing in a direct network has an immediate consequence: any source node can send a message to another destination node (say, Δ hops away) in one routing step (in the process it can also inform all the nodes on the path if intermediate reception capability is assumed at the nodes). This statement need be *qualified*. The distance insensitivity depends on two factors: the message size is relatively large and the distance between source and destination is relatively small. So, there is a practical limit on the length of the path between the source and destination beyond which the communication latency would not be distance insensitive and this limit does depend on various network parameters in addition to the network size[1].

In their study on wormhole broadcast in hypercubes, Ho et. al [7] have chosen n to be the size of the wormhole path for a hypercube with $N = 2^n$ nodes. Our

[1] Without such a limit, one can trivially grow a Hamiltonian path from the source and inform all the nodes in the network in one routing step.

purpose in this paper is to generalize this approach for wormhole broadcast in hypercube networks; we will follow the same guidelines and use m to be the limit on the distance between the source and destination along a w-path, where m is an arbitrary integer, $1 < m \leq n$. The proposed algorithms are more versatile since they can adapt to varying network parameters thus maintaining the distance insensitivity of wormhole routing.

2.3 Reception Capability at Intermediate Nodes

When the processors are connected by a direct point to point network and wormhole routing is used for communication, communication latency depends heavily on whether the nodes on the w-path (between the source and destination nodes) can receive the message for their own use. This very much depends on the architecture of the node: routing controller, its internal channels and its local buffer in the routing controller. Usually, in addition to the switch, the routing controller has a very small buffer, i.e., if the node wants to store the passing message the controller must transfer it to the local memory of the node through internal channels. The routing controller is normally connected to the node's local memory by one pair of input/output internal channel and hence an intermediate node on a w-path cannot, in general, store a copy of the passing message. However, as noted by Ho et. al. in [7], it is technologically possible to add such reception capability at intermediate nodes without any degradation in communication efficiency; real-life examples include HARTS machine [5] and 2D mesh computers [2]. So, it makes sense to talk about implementation of communication primitives assuming the reception capability at intermediate nodes on a w-path.

3 Generalized Wormhole Routing in Hypercube Q_n

The network under study is the binary n-cube denoted by Q_n. A Q_n can be partitioned into 2^{n-k} disjoint Q_ks, $0 \leq k \leq n$. An n-dimensional hypercube Q_n is denoted by X^n implying that all n dimensions need to be spanned to form such a cube. A Q_k (as a subcube of Q_n, where $k < n$), however, is represented by an n-tuple with $(n-k)$ positions fixed (0 or 1) and k X's. For instance, $01XX1$ denotes a Q_2 in Q_5, spanning dimensions 1 and 2 of the original cube.

Communication takes place along one port at a time, i.e., we consider single port communication only in this paper. In the single-port model, each node can only activate one of its ports at a time; and also according to our communication model, nodes are assumed to have intermediate reception capability. We are given a hypercube Q_n and a source node s which wants to broadcast a message to all the nodes in Q_n. , and the maximum length of the w-path is assumed to be m, where $1 < m \leq n$.

Here assuming the maximum w-path length of m, we derive the lower bound on $T(n,m)$, the number of steps required to broadcast a message in Q_n.

Theorem 1 *Under one-port communication model and path length of m in Q_n, $T(n,m) \geq \lceil \frac{n}{log_2(m+1)} \rceil$.*

Proof : The number of informed nodes will increase at most by a factor of $(m+1)$. After step 1, there are $(m+1)$ such nodes and after step i, there $(m+1)^i$. In step $T(n)$, the number of informed nodes should be at least 2^n. It follows: $(m+1)^{T(n)} \geq 2^n$ and thus the result. □

Algorithms for Generalized Wormhole Routing

In the following, two algorithms are developed to broadcast a message in Q_n with w-path length of m. The first algorithm employs Ho's algorithm as a subroutine and hence we adopt the following notation. $Basic(SN, DS)$, where SN stands for the source node and DS stands for dimensions spanned, refers to Ho's broadcasting algorithm [4] which is applied to a subcube with the specified origin and dimensions. For example $Basic(00001; 12)$ involves broadcasting from 00001 within the subcube $00XX1$. At times, broadcasting must be done on a set of subcubes concurrently in which case the address of source nodes take the form of $a_{j-1} \dots a_0$ to imply all the binary j-tuples as source nodes, each residing in one disjoint Q_j. For instance, $Basic(a_1a_0001; 12)$ refers to $Basic(00001; 12)$, $Basic(01001; 12)$, $Basic(10001; 12)$, and $Basic(11001; 12)$, i.e. concurrent broadcasting in 4 disjoint Q_2s.

3.1 Cube-based Broadcasting Algorithm

In this approach, Q_n is partitioned into 2^{n-m} mutually disjoint Q_m's. If one node in each of these Q_m's can be made to have the message, Ho's algorithm *Basic(SN, DS)* can be invoked by the node to complete the broadcasting in Q_m using a w-path of length m. To get the message to at least one node of each Q_m, one can easily construct a Q_{n-m} composed of one node of every Q_m. At this point broadcasting in Q_n has reduced to that of broadcasting in Q_{n-m}. So the algorithm is called recursively until the size of the cube is reduced to m or less. Let $k = \lfloor n/m \rfloor$ and r be an integer such that $n = m * k + r$. These two integers can be computed from given values of n and m. The algorithm uses only one primitive *Basic(SN, DS)* which broadcasts in a Q_m using Ho's algorithm as discussed earlier. The formal description of the algorithm follows.

```
Procedure Cube-based (n,m)
(1)   $k = \lfloor n/m \rfloor$
(2)   $r = remainder(n/m)$
(3)   for $i = 0$ to $k-1$ do
(4)       Call Basic ( $\overbrace{0\cdots0}^{n-i*m \text{ zeros}}$ $a_{im-1}\dots a_0$,
        $i*m\ i*m+1\dots(i+1)*m-1$)
(5)   end for
(6)   If $R = 0$ then Exit
(7)       else Call Basic ( $\overbrace{0\cdots0}^{r \text{ zeros}}$
          $a_{k*m-1}\cdots a_1a_0, n-r\cdots n-1$).
```

Lemma 1 *The time complexity of Ho's basic broadcasting algorithm, i.e., $Basic(0..0, 1\cdots n)$ in a Q_n with w-path length of $m = n$ is given as follows. $T(1) = 1, T(2) = T(3) = 2, T(4) = T(5) = T(6) = 3$, and $T(n) \leq T(n-3)+1$ for $n \geq 7$. [4]*

Theorem 2 *The complexity of the proposed Cube-based broadcast algorithm in a Q_n with path length m is given by $T_c(n,m) = k{*}T(m)+T(r)$ where $n = k{*}m+r$, and $T(m)$ and $T(r)$ are determined from the previous lemma.*

Proof : Obvious from the construction of the algorithm. Note that each iteration of the **for** loop in line 3 takes time $T(m)$ (due to invocation of "Basic" on a Q_m). □

Example 1 *Consider a hypercube Q_8 with the w-path length of three, i.e., $n = 8$ and $m = 3$. Then $k = r = 2$. The steps taken by the algorithm are as follows.*

1. *Call Basic* $[00000000; 0, 1, 2]$ *(the active cube: $00000XXX$); this takes $T(3)$ routing steps.*
2. *Call Basic* $[00000a_2a_1a_0; 3, 4, 5]$ *(the active cubes: $00XXXa_2a_1a_0$); this takes $T(3)$ routing steps.*
3. *Call Basic* $[00a_6a_5\dots a_0; 6, 7]$ *(the active cubes: $XXa_6a_5\cdots a_0$); this takes $T(2)$ routing steps.*

Note that in the first step one cube, in the second step 8 cubes, and in the last step 128 cubes are involved in broadcasting. Since $T(3) = T(2) = 2$, total number of routing steps taken by our algorithm is $T(8,3) = 6$.

3.2 Gray-code based Broadcasting Algorithm

Our second algorithm for broadcasting in Q_n with wormhole path length of $m < n$ is based on Gray codes (GCs) [10]. An n-bit GC contains 2^n binary codewords, each of length n bits, in which two adjacent codewords differ in exactly one bit. The GC sequence for a given n is not unique, and any arbitrary GC sequence can be used for wormhole broadcasting in our algorithm. We consider the *binary reflected gray code $GC(n)$* defined as follows: $GC(1) = \{0,1\}$ and $GC(i)$ is constructed from $GC(i-1)$ by concatenating $GC(i-1)$ and its mirror (reflected) image, and appending a "0" to the left of the codewords in the first half, and a "1" to the left of the codewords in the second half. Symbolically, $GC(i) = 0GC(i-1), 1GC^R(i-1)$ where $GC^R(i-1)$ is the mirror (or reflected) image of $GC(i-1)$. For example, $GC(3) = \{000, 001, 011, 010, 110, 111, 101, 100\}$. If a starting codeword is fixed, one can specify the dimension sequence instead of the codewords where a dimension between two consecutive codewords is the bit positions in which they differ (with the rightmost bit position being 0 and dimensions increasing from right to left). For example, the dimension sequence $dGC(3) = \{0, 1, 0, 2, 0, 1, 0\}$ denotes the same Gray code assuming the starting node 000 is known from the context. We may also need to use only a part of a gray code. A gray code $GC(n)$ has 2^n codewords; suppose we need to use only only $2^{n'}$ of those (note that the codewords are still of length n). We need to add two more parameters: the integer n' and DS which stands for dimensions spanned. For example, $GC(n, n', DS) = GC(4, 2, 03) = \{0010, 0011, 1011, 1010\}$ (this assumes that the starting node is 0010). A similar designation can be arranged for dimension sequence representation of partical gray codes. In the running example we then have: $dGC(4, 2, 03) = \{0, 3, 0\}$

The requirement of differing in one bit for adjacent codewords in GCs resembles the node adjacency relationship of hypercube nodes; and that is why the full n-bit GC specifies a Hamiltonian path in Q_n. Similarly, any subsequence of the n-bit GC corresponds to a path in Q_n. If the origin in a Q_n is known, a Hamiltonian path may be defined by $dGC(n)$ which constitutes the link labels along the path in the order of traversal. Denote by $P(sourcenode, dGC(s))$ the path that originates from *sorucenode* along the dimension sequence specified by $dGC(s)$. Let s be such that $2^s - 1 \leq m$. Furthermore, suppose $n = k' * s + r'$. Then Q_n is partitioned into 2^s disjoint Q_{n-s}'s. Using a GC of length $2^s - 1$ one can broadcast the message to exactly one node in each Q_{n-s}. The algorithm repeats for each Q_{n-s} recursively until the dimension of disjoint subcubes becomes equal to r'. The broadcast will then be completed by routing the message in each $Q_{r'}$ using a GC sequence of length $r' - 1 \leq 2^s - 1$. This algorithm is formally presented below.

```
Procedure GC-based (n,m)
    $s = \lfloor log_2(m+1) \rfloor$;
    $k' = \lfloor n/s \rfloor$;
    $r' = remainder(n/s)$
    for $i = 0$ to $k' - 1$ do
        Call Gray $(n, s, \overbrace{00\cdots00}^{n-i*s \text{ zeros}}\ a_{i*s-1}\cdots a_0;$
            $i*s\ i*s+1 \ldots (i+1)*s-1)$
    end for
    if $r' = 0$ then Exit
    else Call Gray $(n, s, \overbrace{00\cdots00}^{r' \text{ zeros}}$
            $a_{k'*s-1}\cdots a_1 a_0; n-1 \ldots n-r')$.
```

```
Function Gray (n,n', source node, DS)
    Form the $dGC(n, n', DS)$.
    Construct the path $P(source\ node, dGC(n, n', DS))$.
    Broadcast the message along path $P$ in one step.
```

Theorem 3 *The number of routing steps taken by the GC-based algorithm for a given hypercube Q_n and a path length m is*

$$T_g(n,m) = k' + \alpha, \quad \textit{where } n = k' * \lfloor log_2(m+1) \rfloor + r',$$

$$\textit{and } \alpha = \begin{cases} 0 & \textit{if } r' = 0 \\ 1 & \textit{otherwise} \end{cases}$$

Proof : The proof is obvious from the construction of the algorithm. Note that each iteration of the **for** loop involves one routing step and the additional invocation of the function "Gray" is needed only if $r' \neq 0$. □

Example 2 *Let $n = 5$ and $m = 3$. Then $s = 2$, $k' = 2$, $r' = 1$, and $\alpha = 1$. The broadcasting is done in 3 routing steps as follows.*

1. *Call Gray* $[5, 2, 0^5; 0, 1]$ *(the active cube: 0^5XX)*
2. *Call Gray* $[5, 2, 0^3a_1a_0; 2, 3]$ *(the active cubes: $0XXa_1a_0$)*
3. *Call Gray* $[5, 2, 0a_3a_2a_1a_0; 4]$ *(the active cubes: $Xa_3a_2a_1a_0$)*

Note that in the first step one cube, in the second step 4 cubes, and in the last step 16 cubes are involved in broadcasting.

3.3 Comparison of the Two Approaches

In this section we compare the performance of the two approaches to wormhole broadcasting in hypercubes when the w-path length m is restricted and can be arbitrary. Table 1 shows the values of $T_c(n,m)$, $T_g(n,m)$ and $T(n)$ for different combinations of values of n and m. We make the following observations:

- Our cube-based algorithm becomes Ho's algorithm [4] when $m = n$ and when $m < N$, the cube-based algorithm performs worse than Ho's algorithm, but can accommodate arbitrary restriction on the w-path length m.
- The GC based algorithm almost always outperforms the existing Ho's algorithm and at the same time can accommodate any w-path length of m.
- The GC based algorithm is optimal in one-port communication model when $m = n$ and is near-optimal for $m < n$.
- The GC based algorithm is much simpler and easier to implement; does not need any complex decomposition of the hypercubes.

n	m	k	r	k'	r'	$T_c(n,m)$	$T_g(n,m)$	$T(n)$
8	2	4	0	4	0	8	4	
8	4	2	0	2	2	4	3	
8	6	1	2	2	2	5	3	
8	8	1	0	2	0	4	2	4
12	2	6	0	12	0	12	12	
12	4	3	0	6	0	9	6	
12	6	2	0	6	0	6	6	
12	8	1	4	4	0	7	4	
12	10	1	2	4	0	7	4	
12	12	1	0	4	0	5	4	5
16	2	8	0	16	0	16	16	
16	4	4	0	8	0	12	8	
16	6	2	4	8	0	9	8	
16	8	2	0	5	1	8	6	
16	10	1	6	5	1	8	6	
16	12	1	4	5	1	8	6	
16	14	1	2	5	1	8	6	
16	16	1	0	4	0	7	4	6
20	2	10	0	20	0	20	20	
20	4	5	0	10	0	15	10	
20	6	3	2	10	0	11	10	
20	8	2	4	6	2	11	7	
20	10	2	0	6	2	10	7	
20	12	1	8	6	2	9	7	
20	14	1	6	6	2	9	7	
20	16	1	4	5	0	10	5	
20	18	1	2	5	0	9	5	
20	20	1	0	5	0	8	5	7

Table 1: Comparison of the two algorithms

4 Conclusion

Using the one-port communication and path-based wormhole switching, broadcasting in n-cube Q_n was addressed. First, an earlier work is generalized to the case where the path length m varies between 1 and n. A new algorithm based on Gray code is then introduced. This algorithm, while being the best so far, is very simple and easy to implement. Observe that using this approach, for a fixed n, there is a range for m in which the number of broadcasting steps does not change. This range is characterized by $2^s - 1 \leq m < 2^{s+1} - 1$. This is true since for the specified range, $log_2(m+1) = s$ is constant and therefore k' and r' are constant as well. Similarly, for a fixed m, the dimension of the cube can vary in the range $k's \leq n < (k'+1)s$ without affecting the total number of steps for broadcasting. For larger n, the only difference is that a larger portion of the GC sequence can be utilized to cover all the nodes with the same number of steps.

References

[1] W. J. Dally and C. l. Seitz. The Torus routing chip. *Journal of Parallel and Distributed Computing*, 1(3):187–196, 1986.

[2] L. M. Ni and P. K. McKinley. A survey of wormhole routing techniques in direct networks. *IEEE Computer*, 26(2):92–96, February 1993.

[3] P. K. McKinley, Y. J. Tsai, and D. Robinson. Collective communication in wormhole routed massively parallel computers. *IEEE Computer*, 28(12):39–50, December 1995.

[4] C. T. Ho and M. Y. Kao. Optimal broadcast on hypercube with wormhole and e-cube routing. In *Proceedings of International Conference on Parallel and Distributed Systems*, pages 694–697, 1993.

[5] D. D. Kandlur and K. G. Shin. Reliable broadcast algorithms for HARTS. Technical Report CSE-TR-69-90, University of Michigan, Ann Arbor, 1990.

[6] D. F. Robinson and P. K. McKinley. Efficient multicast in all-port wormhole routed hypercubes. *Journal of Parallel and Distributed Computing*, 31:126–140, 1995.

[7] C. T. Ho and M Kao. Optimal broadcast in all-port wormhole routed hypercubes. *IEEE Transactions on Parallel and Distributed Systems*, 6(2):200–204, February 1995.

[8] P. K. Mckinley, H. Xu, A. H. Esfahanian, and L. M. Ni. Unicast based multicast communication in wormhole routed networks. *IEEE Transactions on Parallel and Distributed Systems*, 5(12):1252–1264, December 1994.

[9] C. T. Ho and M. T. Raghunath. Efficient communication primitives on hypercubes. *Journal of Concurrency: Practice and Experience*, 4(6):427–457, September 1992.

[10] Y. Saad and M. H. Shultz. Topological properties of hypercubes. *IEEE Transactions on Computers*, 37(7):867–872, July 1988.