

Cost and Performance Analysis of Scalable Parallel Architectures

SOTIRIOS G. ZIAVRAS, ECE Dept., New Jersey Institute of Technology, Newark, NJ 07102, USA, ziavras@oak.njit.edu

QIAN WANG, CIS Dept., New Jersey Institute of Technology

NIK ALEXANDRIDIS, ECE Dept., National Technical University, Athens, Greece

PARASKEVI PAPATHANASIOU, Data Line Computer Institute, Piraeus, Greece

Abstract

Interprocessor connection networks with very good topological properties are often impossible to build because of their prohibitively high wire complexity. Such a network is the generalized hypercube (GH) that supports full-connectivity of all its nodes in each dimension. We present here the class of HOW interprocessor connections which are capable of lower complexity than GHs, comparable performance, and better scalability. We analyze the hardware cost of HOWs, present communications algorithms for 2-D HOWs, and carry out performance comparisons with binary hypercubes and GHs.

Keywords: Parallel computer, interconnection network, cost analysis, communications operations.

1 Introduction

Developing low-complexity, high-bisection bandwidth, and low-latency networks to interconnect processors (PEs) in MPP systems is a Herculean task. Several message passing interconnection networks have been proposed for parallel computers [1]. To support scalability, current approaches most often use bounded-degree networks, such as meshes or k -ary n -cubes (i.e., tori), with low node degree. However, low-degree networks result in large diameter, large average internode distance, and small bisection width. The high VLSI(wire) complexity problem is unbearable for generalized hypercubes. Contrary to nearest-neighbor k -ary n -cubes that form rings with k nodes in each dimension, GHs implement fully-connected systems with k nodes in each dimension [3]. The n -D (symmetric) generalized hypercube $GH(k, n)$ contains k^n nodes. The address of a node is $x_{n-1}x_{n-2}\dots x_1x_0$, where x_i is a radix- k digit with $0 \leq x_i \leq k-1$. This node is a neighbor to the nodes with addresses $x_{n-1}x_{n-2}\dots x'_i\dots x_1x_0$, for all $0 \leq i \leq n-1$ and $x'_i \neq x_i$. Therefore, two nodes are neighbors if and only if their n -digit addresses differ in a single digit. Therefore, each node has $k-1$ neighbors in each dimension, for a total of $n \cdot (k-1)$ neighbors per node. The n -D $GH(k, n)$ has diameter equal to only n . The increased VLSI cost of

GHs results in outstanding performance that permits optimal emulation of hypercubes and k -ary n -cubes, and efficient implementation of complex communications patterns [2, 4].

2 HOW Architectures

We propose in this paper a new class of architectures, namely HOWs (Highly-Overlapping Windows), that can be implemented effectively with current technology, contain many highly-overlapping GHs, are scalable, and possess outstanding topological properties. HOWs require fewer and shorter interconnections between PEs than GHs, and therefore their construction may be feasible. Let us first introduce 1-D HOWs. $HOW(p, w, 1)$ denotes a 1-D HOW system with p nodes and window size w . Each node with unique address k , where $0 \leq k \leq (p-1)$, is directly connected to all nodes with addresses $0 \leq k \pm i \leq (p-1)$, for $i = 1, 2, 3, \dots, w$. Each PE has up to $2 \cdot w$ neighbors. Therefore, all connections could be short and rather local; the length of channel wires depends heavily on the value of w . The same configuration is also obtained by removing edges from the 1-D fully-connected system (a GH). The n -D $HOW(p, w, n)$ is constructed recursively, so that each node can have up to $2 \cdot w \cdot n$ neighbors. This system contains many GHs $GH(w+1, n)$. Also, it can be derived from the $GH(p, n)$ by uniformly removing some edges.

Table 1 compares the numbers of channels in several systems, all with the same number N of PEs. We assume bidirectional data channels for full-duplex communications, and that $N = k^n = 2^m$ (therefore, $k = N^{1/n} = 2^{m/n}$). We focus on 2-D HOWs because of their simplicity, high bisection width, and ease of implementation. HOWs have reduced pin-outs compared to GHs. In contrast, binary hypercubes are not scalable and are more difficult to build because of long wires. HOWs also improve dramatically the topological properties of k -ary n -cubes; the $HOW(k, 1, n)$ is the n -D mesh. The bisection widths of these systems are $\frac{N}{2}$, $2N^{1-\frac{1}{n}}$, $\frac{N}{4}N^{\frac{1}{n}}$, $w\frac{w+1}{2}N^{\frac{1}{2}}$, and $w\frac{w+1}{2}N^{1-\frac{1}{n}}$, respectively. These results show that HOWs can have good performance and reasonable VLSI cost when appropriate

values of w and n are chosen, for a given value of N .

Network	Channels
$\log_2 N$ -cube	$\frac{N}{2} \log_2 N$
$N^{\frac{1}{n}}$ -ary n -cube	nN
$GH(N^{\frac{1}{n}}, n)$	$(N^{\frac{1}{n}} - 1)n\frac{N}{2}$
$HOW(\sqrt{N}, w, 2)$	$\sqrt{N}w(2\sqrt{N} - w - 1)$
$HOW(N^{\frac{1}{n}}, w, n)$	$\frac{n}{2}N^{1-\frac{1}{n}}w(2N^{\frac{1}{n}} - w - 1)$

Table 1: Networks with $N = p^n = 2^m$ nodes.

3 Further Cost Analysis

The *crossing number of a graph* is the minimum number of edge crossings needed to draw the graph in the plane. It is related to the area needed to lay out the graph for VLSI implementation. To eliminate all edge crossings, several printed-circuit layers may have to be implemented. Not only does the number of layers affect the VLSI cost, but the thickness also of each layer contributes to the cost measure. To determine the VLSI/wire cost, we measure the complexity of each system based on the minimum number of layers required in the colinear layout of the circuit for zero edge crossings and/or the width of each layer. In the colinear layout, all nodes in the 1-D system lie on the same straight line. The chosen rules of routing the wires for 1-D systems are:

- We consecutively number the PEs 0, 1, 2, \dots , $p - 1$, from left to right.
- Going from left to right, for even-numbered PEs the wires go to the top half.
- For odd-numbered PEs, the wires go to the bottom half.

These routing rules minimize the maximum collective width, *MCW* (expressed in number of wires), in the x dimension. The following theorems are pertinent.

Theorem 1. The *MCW* in the colinear layout of the 1-D *HOW*($p, w, 1$) with a single layer is

$$MCW = \begin{cases} \frac{w}{2}(\frac{w}{2} + 1) & \text{for even } w \\ (\frac{w+1}{2})^2 & \text{for odd } w \end{cases}$$

for practical cases with $w < \frac{p+1}{2}$. For the 1-D GH $GH(p, 1)$, the value of *MCW* is $(p - 3)\phi + p - 1 - 2\phi^2$ with $\phi = \lfloor \frac{p-1}{4} \rfloor$.

Therefore, HOWs have much smaller *MCW* than GHs for practical cases because this width is $O(w^2)$ and $O(p^2)$, respectively. Another routing technique, namely *restricted routing*, requires only two layers for the implementation of any system represented in 2-D; horizontal and vertical wire segments are laid on two different layers. Horizontal and vertical wires can then cross over each other without any

electrical connection. If a connection is needed, a contact is placed at the respective intersection; these contacts contribute to the VLSI cost. The wiring cost has four components:

- The number of wires. It is $O(wp^2)$ and $O(p^3)$ for 2-D HOWs and GHs, respectively.
- The *MCW* which affects the cost of the larger layer. It is $O(w^2)$ and $O(p^2)$ for HOWs and GHs, respectively.
- The length of wires. The maximum length is $O(w)$ and $O(p)$ for HOWs and GHs, respectively.
- The number of electrical contacts between the two layers. It is double the number of wires. That is, $O(wp^2)$ and $O(p^3)$ for HOWs and GHs, respectively.

HOWs are superior to GHs even with restricted routing.

4 Communications on 1-D HOWs

We first develop algorithms for 1-D HOWs. The *communication latency*, i.e. the time taken to communicate a message between two PEs in the system, depends on the following parameters. **Startup time** (t_s): the time consumed by the sender. **Per-word transfer time** (t_w): the time taken by a word to traverse a channel. **Combining time** (t_c): the time consumed by an intermediate node to switch a message from an input to an output port; it also includes the time to combine incoming messages, if needed, and send them to the appropriate output port. We calculate the time taken by a message to reach the receiver's input port.

In **store-and-forward (SF) routing**, each intermediate PE forwards the message to the next PE after it has received the entire message. **Wormhole routing** divides a message into **flits**. Normally, the flit size coincides with the channel width. The combining time t_c is ignored in wormhole routing. We develop algorithms under *three communication models*. For all of the models, each PE can receive more than one message at a time in different input ports.

- **Model-1:** Each PE can use only one output port at a time.
- **Model-2:** Each PE can use multiple output ports simultaneously, as long as all output ports contain the same value.
- **Model-3:** Each PE can use multiple output ports simultaneously, and different output ports can have different values.

The analysis is done each time for SF and wormhole routing, in this order.

4.1 One-to-All Broadcasting

Model-1. Assume messages of size m words for all operations. Since there is only one output port “available” for each PE at each transfer step, we consider two different stages. We assume that the leftmost PE is the source, for worst case timing. In the first stage, we copy the data to all PEs in the source’s window of size w . In the second stage, the data in the leftmost window is propagated to the right, one window size at a time. We introduce two parameters here: s_1 represents the number of transfer steps needed to fill the first window and s_2 represents the number of transfer steps needed in the second stage to copy the values in the first window into the remaining windows. In the first stage, the propagation doubles each time the number of PEs that receive the message, and therefore the PEs within the window are assumed to form a binary tree. We have the following relations: $s_1 = \lceil \log(w + 1) \rceil$ and $s_2 = \lceil (p - 2^{s_1})/w \rceil$. All logarithms are to the base 2. The communication time $T_{1_to_all,1}$ for one-to-all broadcasting under model-1 and SF routing has the *upper bound* of $t_s + mt_w \lceil \log p \rceil + t_c(\lceil \log p \rceil - 1) = O(m \log p)$ if $(p - 1) \leq w$ and $t_s + mt_w(s_1 + s_2) + t_c(s_1 + s_2 - 1) = O(m \log w + m \frac{p}{w})$ if $(p - 1) > w$. This asymptotic time is optimal.

With wormhole routing, the *upper bound* of $T(WR)_{1_to_all,1}$ is $t_s + t_w \lceil \log p \rceil + (m - 1)t_w = O(m + \log p)$ if $(p - 1) \leq w$ and $t_s + t_w(s_1 + s_2) + (m - 1)t_w = O(m + \log w + \frac{p}{w})$ if $(p - 1) > w$, assuming that incoming data can simultaneously be stored locally and also be transferred to the next PE in the path.

Model-2 and Model-3. Assume the leftmost PE as the source. Model-2 is not inferior to model-3 because up to w output ports are “available” to the right of each PE at each transfer step as long as these ports transfer the same value, which is the case here. The first stage now consumes one transfer step and the transfer steps are $\lceil (p-1)/w \rceil$. The communication time $T_{1_to_all,2}$ has the *upper bound* of $t_s + mt_w \lceil \frac{p-1}{w} \rceil + t_c(\lceil \frac{p-1}{w} \rceil - 1) = O(m \frac{p}{w})$. This asymptotic time is optimal.

With wormhole routing, for $T(WR)_{1_to_all,2}$ the *upper bound* is $t_s + t_w \lceil \frac{p-1}{w} \rceil + (m - 1)t_w = O(m + \frac{p}{w})$.

4.2 All-to-All Broadcasting

Model-1. We choose channels that form a ring. If communication is performed circularly in a single direction, then each PE receives all $(p - 1)$ pieces of information from all other PEs in $(p - 1)$ steps. The time taken by the entire operation is $t_s + mt_w(p - 1) + t_c(p - 2) = O(mp)$. This asymptotic time is optimal because each PE can use only one output port at a time, and therefore each message must make $p - 1 = O(p)$ hops.

With wormhole routing, the communication time is $t_s + mt_w(p - 1) = O(mp)$ because the header of each message is blocked at each intermediate node until the previous message has completely departed.

Model-2. In the first stage, each PE sends its message to all of its neighbors. In the remaining stages, assume the stage i , where $i = 1, 2, \dots, \lceil \frac{p-1}{w} \rceil - 1$. In one direction, beginning from position iw and also involving all its successors, send the messages from the PEs $0, 1, \dots, (p - 1 - iw - 1)$ through all possible channels. In the other direction, beginning from position $(p - 1 - iw)$ and also involving all its predecessors, send the messages from the PEs $p - 1, p - 2, \dots, (iw + 1)$. If there is an overlap between these two directions, then split this stage into two steps in order to make sure that every PE sends just one value at a time. From all the messages it contains, each time a PE sends out the message received earlier from its most distant PE. The time is $t_s + mt_w(\lceil \frac{p-1}{w} \rceil + x) + t_c(\lceil \frac{p-1}{w} \rceil + x - 1)$ where x is the number of stages needed to be split into two steps, and x should satisfy the condition $xw < p - 1 - xw$. So x is the largest integer less than $\frac{p-1}{2w}$. Therefore, the complexity is $O(m \frac{p}{w})$. This asymptotic time is optimal because the diameter of the system is $O(\frac{p}{w})$.

With wormhole routing , the time is $t_s + mt_w(\lceil \frac{p-1}{w} \rceil + x) = O(m \frac{p}{w})$ because of message blocking on reused channels.

Model-3. This procedure is very similar to that for model-2. Since each individual PE can send different messages at the same time, we do not need to split any step. The time is optimal and given by $t_s + mt_w \lceil \frac{p-1}{w} \rceil + t_c(\lceil \frac{p-1}{w} \rceil - 1) = O(m \frac{p}{w})$.

With wormhole routing, the time is $t_s + mt_w \lceil \frac{p-1}{w} \rceil = O(m \frac{p}{w})$.

4.3 One-to-All Personalized Communication

Model-1 and Model-2. No matter what the window size is, it will take $(p - 1)$ transfer steps. A ring is used to communicate values. Messages going farther have higher priority of transmission. The total time taken by this operation is $t_s + mt_w(p - 1) + t_c(\lceil \frac{p-1}{2} \rceil - 1) = O(mp)$. This is similar to the asymptotic time consumed by the source, and therefore it is optimal. The shortest paths in the ring are chosen to reach respective destinations. For the sake of simplicity, assume that the source is p_0 . To reach the PE p_x , where $1 \leq x \leq (p - 1)$, the message makes $\lceil \frac{x}{2} \rceil$ hops. Assume that the source first sends out the messages destined for the odd-numbered PEs. It then transmits messages to the even-numbered PEs. Assume for the second case the PE p_x with $x = 2y$. This PE will receive its message with delay $t_c(y - 1) + mt_w(y - 1)$ after it was transmitted by the source. The time left for the source to complete the entire operation is $mt_w(y - 1)$, because $(y - 1)$ is the number of

messages still to be transmitted. Therefore, the “combining time” term used in the equation is for the worst case, where $y = \lceil \frac{p-1}{2} \rceil$.

With wormhole routing, the total number of flits to be transferred by the source is $(p-1)m$. Messages going farther have higher priority of transmission. The time is $t_s + mt_w(p-1) = O(mp)$.

Model-3. For the worst case, we assume p_0 to be the source. First, the PE p_0 passes the w most distant messages to its w neighbors, so that a destination PE with higher address gets a message for a higher-addressed PE. Second, the PE p_0 similarly passes the next w most distant messages to its window, while all PEs that received an intermediate message earlier pass that message to their neighbor at distance w in the next window. The second step repeats until all PEs receive their own message. The time is $t_s + mt_w \lceil \frac{p-1}{w} \rceil + t_c(\lceil \frac{p-1}{w} \rceil - 1) = O(m \frac{p}{w})$ which has the same asymptotic complexity with the time consumed by the source, and therefore it is optimal. This is the lower bound because the PE has to communicate $p-1$ values and can send out up to w values in each cycle.

With wormhole routing, all PEs receive their messages simultaneously in time $t_s + mt_w \lceil \frac{p-1}{w} \rceil$, because of message pipelining and message blocking resulting from the m -flit messages. Therefore, the time is $t_s + m \lceil \frac{p-1}{w} \rceil t_w = O(m \frac{p}{w})$ which is again optimal because it is identical to the time consumed by the source with peak utilization of its communication ports and no data duplication.

4.4 All-to-All Personalized Communication

Model-1 and Model-2. We form a ring. In each transfer step every PE transfers the m -word message destined for its farthest remaining PE. If only one direction in the ring is used for all transfers, then the total number of transfer steps is equal to $\sum_{i=1}^{p-1} (p-i) = \sum_{i=1}^{p-1} i = \frac{(p-1)p}{2}$. The total time is $t_s + \sum_{i=1}^{p-1} mt_w(p-i) + \sum_{i=1}^{p-1} t_c(p-i-1) = t_s + mt_w \frac{(p-1)p}{2} + t_c \frac{(p-1)(p-2)}{2} = O(mp^2)$. However, for the shortest paths both directions in the ring should be used. In this case, there are $\lceil \frac{p-1}{2} \rceil$ “large” communications stages. In the i -th “large” stage, where $i = 1, 2, \dots, \lceil \frac{p-1}{2} \rceil$, each PE transmits the respective messages to the PEs at the same distance i to its left and to its right, exclusively in this order. If p is even, then the $\lceil \frac{p-1}{2} \rceil$ -th “large” stage implements transmissions in only one of the two directions in the ring. Therefore, the total number of transfer steps to neighbors is equal to $2 \sum_{i=1}^{\lceil \frac{p-1}{2} \rceil} i - (\lceil \frac{p-1}{2} \rceil - \lfloor \frac{p-1}{2} \rfloor) = 2 \frac{1}{2} \lceil \frac{p-1}{2} \rceil (\lceil \frac{p-1}{2} \rceil + 1) - (\lceil \frac{p-1}{2} \rceil - \lfloor \frac{p-1}{2} \rfloor) = \lceil \frac{p-1}{2} \rceil^2 + \lfloor \frac{p-1}{2} \rfloor$. The total time is $t_s + mt_w(\lceil \frac{p-1}{2} \rceil^2 + \lfloor \frac{p-1}{2} \rfloor) + 2 \sum_{i=0}^{\lceil \frac{p-1}{2} \rceil - 1} t_c i - (\lceil \frac{p-1}{2} \rceil - \lfloor \frac{p-1}{2} \rfloor) t_c = t_s + mt_w(\lceil \frac{p-1}{2} \rceil^2 + \lfloor \frac{p-1}{2} \rfloor) + t_c(\lceil \frac{p-1}{2} \rceil^2 - 2\lceil \frac{p-1}{2} \rceil + \lfloor \frac{p-1}{2} \rfloor) = O(mp^2)$ which is asymptotically optimal because each PE sends out $O(p)$ mes-

sages of m words each, and the average distance travelled is $O(p)$.

With wormhole routing, the time is $t_s + 2 \sum_{i=1}^{\lceil \frac{p-1}{2} \rceil} m t_w i - mt_w(\lceil \frac{p-1}{2} \rceil - \lfloor \frac{p-1}{2} \rfloor) = t_s + mt_w(\lceil \frac{p-1}{2} \rceil^2 + \lfloor \frac{p-1}{2} \rfloor) = O(mp^2)$.

Model-3. We present a simple procedure that comprises two stages. The basic idea is to use the largest possible number of linear arrays for pipelined message transfers, with the smallest possible number of nodes per such array. **First stage:** this is the initialization stage where local transfers are employed to move messages to PEs that belong to the aforementioned linear arrays. Every PE passes all relative messages to neighbors in its window(s). For a given destination message, it passes that message to its neighbor that belongs to a linear array containing that destination; if two such neighbors exist, the one closer to the destination is chosen. It takes up to $s_1 = \lceil \frac{p-1}{w} \rceil$ cycles to finish the initialization, which is the same as the maximum number of values to be sent from a PE to another one. **Second stage:** the linear arrays are used to transfer the values. There are w linear arrays to be used. We need up to $s_2 = \lceil \frac{p-1}{w} \rceil - 1$ cycles to finish the broadcasting along the linear arrays, which is the same as the maximum number of values a PE has to send in a single dimension; messages going farther have higher priority. The time is $t_s + m(s_1 + s_2)t_w + m(s_1 + s_2 - 1)t_c = t_s + (2\lceil \frac{p-1}{w} \rceil - 1)mt_w + mt_c(2(\lceil \frac{p-1}{w} \rceil - 1)) = O(m \frac{p}{w})$.

With wormhole routing, the time is $t_s + (2\lceil \frac{p-1}{w} \rceil - 1)mt_w = O(m \frac{p}{w})$.

5 Communication Operations on 2-D HOWs

Assume symmetric 2-D HOW systems with p PEs. Also, p_{ij} denotes the PE on row i and column j , where $i, j = 0, 1, \dots, \sqrt{p} - 1$.

5.1 One-to-All Broadcasting

Model-1. We first have to determine which of the row or column window the source belongs to is closer to the center of that row or column, respectively. If it is the row window, then the source broadcasts within that row, and this is followed by broadcasting from those row PEs into all columns. Otherwise, we begin with column broadcasting. However, here we assume the worst case, where the source PE is in the first window of the corresponding 1-D HOW row and column subsystems. Using the same notations as for the 1-D HOW system, s_1 represents the number of transfer steps needed to fill the first window in this row and s_2 represents the number of transfer steps needed in the second stage to copy the values from the first window into the remaining windows of this row. We already know the

following relations among s_1, s_2 , and w : $s_1 = \lceil \log(w + 1) \rceil$ and $s_2 = \lceil (\sqrt{p} - 2^{s_1})/w \rceil$. This operation is done by first broadcasting within the aforementioned row and then from that row within all the columns. The communication time under model-1 with SF routing has the *upper bound* $t_s + 2mt_w \lceil \log \sqrt{p} \rceil + t_c(2 \lceil \log \sqrt{p} \rceil - 1) = O(m \log p)$ if $(\sqrt{p} - 1) \leq w$ and $t_s + 2mt_w (s_1 + s_2) + t_c(2(s_1 + s_2) - 1) = O(m \log w + m \frac{\sqrt{p}}{w})$ if $(\sqrt{p} - 1) > w$. We need time $O(m \log p)$ for the $GH(\sqrt{p}, 2)$.

With wormhole routing, the *upper bound* is $t_s + 2t_w \lceil \log \sqrt{p} \rceil + (m - 1)t_w = O(m + \log \sqrt{p})$ if $(\sqrt{p} - 1) \leq w$ and $t_s + 2t_w (s_1 + s_2) + (m - 1)t_w = O(m + \log w + \frac{\sqrt{p}}{w})$ if $(\sqrt{p} - 1) > w$, assuming that incoming data can be stored locally and can simultaneously be transferred to the next PE in the path. We need time $O(m + \log p)$ for the $GH(\sqrt{p}, 2)$.

Models 2 and 3. This method first broadcasts within the row and then within all columns. The *upper bound* is $t_s + 2mt_w \lceil \frac{\sqrt{p}-1}{w} \rceil + t_c(2 \lceil \frac{\sqrt{p}-1}{w} \rceil - 1) = O(m \frac{\sqrt{p}}{w})$.

With wormhole routing, the *upper bound* is $t_s + 2t_w \lceil \frac{\sqrt{p}-1}{w} \rceil + (m - 1)t_w = O(m + \frac{\sqrt{p}}{w})$ assuming that the dimension to be traversed is changed just after the first flit is received. Both times are $O(m)$ for the $GH(\sqrt{p}, 2)$.

5.2 All-to-All Broadcasting

The procedure repeats many times the corresponding procedure for the 1-D HOW system. That is, PEs first exchange messages along rows, so that each PE has \sqrt{p} messages at the end for the PEs on its own column. Then, PEs exchange their \sqrt{p} messages along columns by repeating the same procedure \sqrt{p} times within the columns.

Model-1. For model-1, there is only one output port of each PE we can use at a time. In order to let every PE pass some information to a neighbor, we deliberately choose some channels to form a ring in each row/column. We assume pipelining of messages along rows and columns. We start with all-to-all row broadcasting that takes time $t_s + \tau = t_s + (\sqrt{p} - 1)mt_w + t_c(\sqrt{p} - 2)$, as derived for the 1-D HOW system. The \sqrt{p} column broadcasts then take time $\sqrt{p}\tau$, because all-to-all 1-D HOW broadcasting is repeated \sqrt{p} times. The time is $t_s + (1 + \sqrt{p})mt_w(\sqrt{p} - 1) + (1 + \sqrt{p})t_c(\sqrt{p} - 2) + t_c = t_s + (p - 1)mt_w + (p - \sqrt{p} - 1)t_c = O(mp)$. The last t_c term is for switching from row broadcasting into column broadcasting. This asymptotic time is optimal because each PE can use only one output port at a time, and therefore each message will make $O(p)$ hops to visit all $O(p)$ PEs.

With wormhole routing, within each row, the entire time is $t_s + m(\sqrt{p} - 1)t_w$, assuming the formation of a ring. This is because each PE starts receiving flits with the first data transfer, pipelining of messages is applied, and

the total number of flits each PE receives is $m(\sqrt{p} - 1)$. Similarly, for columns the time is $m\sqrt{p}(\sqrt{p} - 1)t_w$. The time is $t_s + m(1 + \sqrt{p})(\sqrt{p} - 1)t_w = t_s + m(p - 1)t_w = O(mp)$. We need the same times for the $GH(\sqrt{p}, 2)$.

Model-2. Based on the algorithm proposed for the 1-D HOW system, the time taken by this operation is $t_s + (1 + \sqrt{p})mt_w(\lceil \frac{\sqrt{p}-1}{w} \rceil + x) + t_c(1 + \sqrt{p})(\lceil \frac{\sqrt{p}-1}{w} \rceil + x - 1) = O(m \frac{p}{w})$ where x is the largest integer less than $\frac{\sqrt{p}-1}{2w}$. The algorithm for the 1-D HOW system is used $(1 + \sqrt{p})$ times, once for the rows and \sqrt{p} times for the columns.

With wormhole routing, the time is $t_s + 2mt_w(\lceil \frac{\sqrt{p}-1}{w} \rceil + x)(1 + \sqrt{p}) = O(m \frac{p}{w})$. Both times are $O(m\sqrt{p})$ for the $GH(\sqrt{p}, 2)$.

Model-3. It is very similar to the procedure for model-2. Since each PE can send different messages at the same time, we do not need to split any stage. The time is $t_s + (1 + \sqrt{p})mt_w \lceil \frac{\sqrt{p}-1}{w} \rceil + t_c(1 + \sqrt{p})(\lceil \frac{\sqrt{p}-1}{w} \rceil - 1) = O(m \frac{p}{w})$.

With wormhole routing, the time is $t_s + mt_w \lceil \frac{\sqrt{p}-1}{w} \rceil (1 + \sqrt{p}) = O(m \frac{p}{w})$. Both times are $O(m\sqrt{p})$ for the $GH(\sqrt{p}, 2)$.

5.3 One-to-All Personalized Communication

We will describe only the procedure for SF routing under model-3. Unlike one-to-all broadcasting, this operation does not involve any duplication of data. We give higher priority to messages that must travel longer distances and all column and row connections for a PE are used simultaneously. It will take up to $\lceil \frac{\sqrt{p}-1}{w} \rceil m(\sqrt{p} - 1)$ steps to send all messages for the last column on the respective row PE, and up to $\lceil \frac{\sqrt{p}-1}{w} \rceil m$ steps on that column. The *upper bound* on the time is $t_s + \lceil \frac{\sqrt{p}-1}{w} \rceil \sqrt{p}mt_w + (2 \lceil \frac{\sqrt{p}-1}{w} \rceil - 1)t_c = O(m \frac{p}{w})$ which is optimal (it is the same as the time consumed by the source PE).

5.4 All-to-All Personalized Communication

Models 1 and 2. We form rings on rows and columns. In each transfer step the message size is m words and every PE tries to transfer the message(s) destined for its farthest PE. We start with row transfers and continue with \sqrt{p} all-to-all personalized communications within columns. Based on the implementation of $(\sqrt{p} + 1)$ all-to-all personalized 1-D HOW operations, we get time $t_s + (\sqrt{p} + 1)mt_w(\lceil \frac{\sqrt{p}-1}{2} \rceil^2 + \lfloor \frac{\sqrt{p}-1}{2} \rfloor) + (\sqrt{p} + 1)t_c(\lceil \frac{\sqrt{p}-1}{2} \rceil^2 - 2 \lceil \frac{\sqrt{p}-1}{2} \rceil + \lfloor \frac{\sqrt{p}-1}{2} \rfloor) = O(mp^3/2)$.

With wormhole routing, the communication time is $t_s + (\sqrt{p} + 1)mt_w(\lceil \frac{\sqrt{p}-1}{2} \rceil^2 + \lfloor \frac{\sqrt{p}-1}{2} \rfloor) = O(mp^3/2)$.

Model-3. The implementation requires the steps:

- Each PE transmits \sqrt{p} values to each of the other $\sqrt{p} - 1$ PEs on its row, to be later distributed on the corresponding columns. At the end of this step, each PE has received $(\sqrt{p} - 1) * \sqrt{p}$ messages. This operation is equivalent to \sqrt{p} all-to-all personalized communications on an 1-D HOW (row).
- In this step, each PE transmits the values it received earlier and its own $\sqrt{p} - 1$ values to the other PEs on its column. Since $\sqrt{p} - 1$ of the messages received in the first step were destined for this particular PE, the number of messages to be transmitted is $(\sqrt{p} - 1) * \sqrt{p} - (\sqrt{p} - 1) + (\sqrt{p} - 1) = (\sqrt{p} - 1) * \sqrt{p}$.

So the total number of all-to-all personalized 1-D HOW communications is $\sqrt{p}(\sqrt{p} - 1) + \sqrt{p} = p$. Therefore, the time is $t_s + p (2 \lceil \frac{\sqrt{p}-1}{w} \rceil - 1)mt_w + pmt_c (2 \lceil \frac{\sqrt{p}-1}{w} \rceil - 1) = O(m \frac{p^{3/2}}{w})$.

With wormhole routing, the time is $t_s + p mt_w 2 \lceil \frac{\sqrt{p}-1}{w} \rceil = O(m \frac{p^{3/2}}{w})$.

6 Performance Comparisons

We compare the communications capabilities of 2-D HOWs, binary hypercubes, and 2-D GHs, all with the same number p of PEs. We consider communications under model-3 for store-and-forward routing. Table 2 summarizes their performance and compares them using as the cost measure the product of the communication time and the node pin-out; systems with lower cost are preferable. The pin-out for a network is the number of wires per node; it is the product of the node degree and the channel width. It is a very widely used measure of the VLSI cost. The cost of implementing these communications operations is asymptotically identical for HOWs and GHs; this is very important as HOWs are much easier to implement than GHs. Therefore, HOWs are proven viable networks in the field of very high performance computing.

7 Conclusions

We introduced a class of scalable architectures and relevant communications algorithms for popular communications models. Our results prove the viability of our scalable architectures for very high performance computing.

Acknowledgment: The work presented in this research was supported in part jointly by NSF and DARPA under the New Millennium Computing Point Design Grant ASC-9634775.

Network	HOW	BH	GH
Pin-out	w	$\log p$	\sqrt{p}
$T_{1_to_1,3}$	$m \frac{\sqrt{p}}{w}$	$m \log p$	m
$T_{1_to_all,3}$	$m \frac{\sqrt{p}}{w}$	$m \log p$	m
$T_{all_to_all,3}$	$m \frac{p}{w}$	mp	$m\sqrt{p}$
$T_{1_to_all_pers,3}$	$m \frac{p}{w}$	mp	$m\sqrt{p}$
$T_{all_to_all_pers,3}$	$m \frac{p^{3/2}}{w}$	$mp \log p$	mp
$C_{1_to_1,3}$	$m\sqrt{p}$	$m \log^2 p$	$m\sqrt{p}$
$C_{1_to_all,3}$	$m\sqrt{p}$	$m \log^2 p$	$m\sqrt{p}$
$C_{all_to_all,3}$	mp	$mp \log p$	mp
$C_{1_to_all_pers,3}$	mp	$mp \log p$	mp
$C_{all_to_all_pers,3}$	$mp^{3/2}$	$mp \log^2 p$	$mp^{3/2}$

Table 2: Cost comparison of networks using the $O(\cdot)$ notation. Cost= (communication time) * (node pin-out). BH and C stand for binary hypercube and cost, respectively.

References

- [1] S. G. Ziavras, "RH: A Versatile Family of Reduced Hypercube Interconnection Networks," *IEEE Trans. Paral. Distr. Systems* 5(11), 1210-1220.
- [2] P. Fragopoulou, S. G. Akl, and H. Meijer, "Optimal Communication Primitives on the Generalized Hypercube Network," *Journ. Paral. Distr. Comput.* 32, 1996, 173-187.
- [3] L.N. Bhuyan and D.P. Agrawal, "Generalized Hypercube and Hyperbus Structures for a Computer Network," *IEEE Trans. Comput.* 33 (4), 1984, pp. 323-333.
- [4] S.G. Ziavras and S. Krishnamurthy, "Evaluating the Communications Capabilities of the Generalized Hypercube Interconnection Network," *Conc. Prac. Exp.*, 11(3), 1999.