Systolic Routing in an Optical Butterfly

Risto T. Honkanen

Report A/2003/3

ISBN 951-781-267-3

UNIVERSITY OF KUOPIO

Department of Computer Science

P.O.Box 1627, FIN-70211 Kuopio, FINLAND

Systolic Routing in an Optical Butterfly

Risto T. Honkanen
Department of Computer Science and Applied Mathematics
University of Kuopio
P.O.Box 1627
FIN-70211 Kuopio, FINLAND
Email: rthonkan@cs.uku.fi

Abstract. In this paper we present an all-optical network architecture and a systolic routing protocol for it. The r-dimensional optical butterfly ($\mathcal{OBF}$) network consists of $r2^r$ nodes and $r2^{r+1}$ edges. Processors are deployed at the level 0 (identical to level r) nodes of the network. Routing is based on the use of a cyclic control bit sequence and scheduling. The systolic routing protocol ensures that no electro-optical conversion is needed in the intermediate routing nodes and all the packets injected into the routing machinery will reach their target without collisions. A work-optimal routing of an h-relation is achieved with a reasonable size of $h \in \Omega(n \log n)$.

1 Introduction

Optics offers a possibility to increase the bandwidth of intercommunication networks. Optical communication offers several advantages in comparison with its electronic counterpart, for example, a possibility to use broader bandwidth and insensitivity to external interferences. These advantages have been covered, e.g., by Saleh and Teich in their book [12].

Our work is motivated by another kind of communication problem, namely the emulation of shared memory with distributed memory modules [6]. If a parallel computation has enough parallel *slackness*, the implementation of shared memory can be reduced to efficient routing of an h-relation [14]. An h-relation is a routing problem where each processor has at most h packets to send and it is the target of at most h packets [1]. An implementation of an h-relation is said to be *work-optimal* at *cost* c, if all the packets arrive at their targets in time ch. A precondition for work-optimality is that $h \in \Omega(\phi)$, where ϕ is the diameter of the network, and that the network can move $\Omega(n\phi)$ packets in each time steps, where n is the number of processors. Otherwise slackness cannot be used to "hide" latency influenced by the diameter [6]. For an r-dimensional optical butterfly ($\mathcal{OBF}$) having $n = 2^r$ processors the diameter $\phi = r$ fulfills this condition when $h \in \Omega(n \log n)$.

Butterfly networks are widely used in intercommunication machineries. There are several reasons to the popularity of butterfly networks. Firstly, they have a simple recursive structure. Secondly, in an r-dimensional butterfly any input p is linked to any output p' by a unique path of length r [7]. Most of implementations of butterfly based networks use packet switching as the routing strategy [7, 9, 13]. A drawback of packet switching is that routing decisions must be done in electronic form. Liu and Gu have presented an all-optical implementation based on wavelength-division multiplexing (WDM) [8]. An advantage of their implementation is that electro-optic conversions are avoided. A disadvantage is that a number of wavelengths and wavelength converters are needed to realize connections.

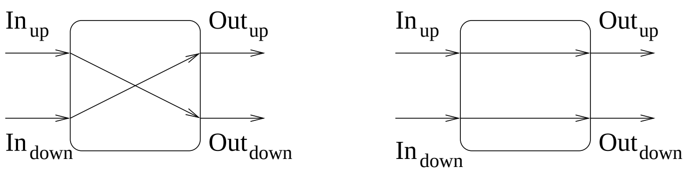
In this work we present an all-optical network architecture and a systolic routing protocol for it. The r-dimensional optical butterfly network consists of $r2^r$ nodes and $r2^{r+1}$ edges. Processors are deployed at the level 0 nodes of the network. Routing nodes are connected to each other by optical links. In this paper we present a novel packet routing protocol, called the *systolic routing protocol*. Additionally, when a packet is injected into the routing machinery, neither electro-optic conversions are needed during its path from source to target processor nor any collisions may happen between two distinct packets. An r-dimensional $\mathcal{OBF}$ can route an h-relation in $\Theta(h)$ time, if $h \in \Theta(n \log n)$. Section 2 presents the internal structure of routing nodes and the structure of an $\mathcal{OBF}$ network. In Section 3 we introduce the systolic routing protocol. Section 4 presents the analysis of our construction. Section 5 sketches conclusions and future work.

2 Optical Butterfly with Systolic Routers

We study on the r-dimensional structure of $\mathcal{OBF}$ of diameter $\phi = r$ and having $n = 2^r$ processing nodes. We represent the structure of routing nodes in Section 2.1. Section 2.2 introduces the construction of $\mathcal{OBF}$. Section 2.3 discusses the feasibility of our construction.

2.1 Systolic Routers for $\mathcal{OBF}$

Each routing node of an $\mathcal{OBF}$ has two incoming and two outgoing links. A routing node can be in two states. When a routing node routes incoming packets from input links in_{up} and in_{down} to output links out_{down} and out_{up} respectively, it is said to be in the *invert* state and when it routes incoming signals from input links in_{up} and in_{down} to output links out_{up} and out_{down} respectively, it is said to be in the *push* state. The two possible states of routing nodes are presented in Figure 1.



(a) Invert state (b) Push state

Figure 1: Two possible states of routing nodes.

The basic component of routing nodes is the electrically controlled all-optical 2×2 switch. Switches can be implemented by $LiNbO_3$ technology [12]. The construction of routing nodes ensures that signals never collide and routing of the packets works correctly if we can arrange a situation that both incoming packets never prefer the same output link. We will show that this kind of situation is arrangeable.

2.2 Construction of Optical Butterfly

The r-dimensional butterfly consists of $r2^r$ nodes and $r2^{r+1}$ edges. The nodes correspond to pairs $\langle w, i\rangle$, where i is the level of the node $(0 \le i \le r)$ and w is an r-bit binary string denoting the row number of the node. Levels r and 0 are considered to be the same. Two nodes $\langle w, i\rangle$ and $\langle w', i'\rangle$ are connected by an edge (optical link) if and only if $i' = i + 1$, and either [7]

i. w and w' are identical (straight edges), or

ii. w and w' differ in precisely the i'th bit (cross edges).

The construction of 3-dimensional $\mathcal{OBF}$ out of two 2-dimensional $\mathcal{OBF}$'s is presented in Figure 2. In Figure 2, a circle indicates a processing node, a rounded square

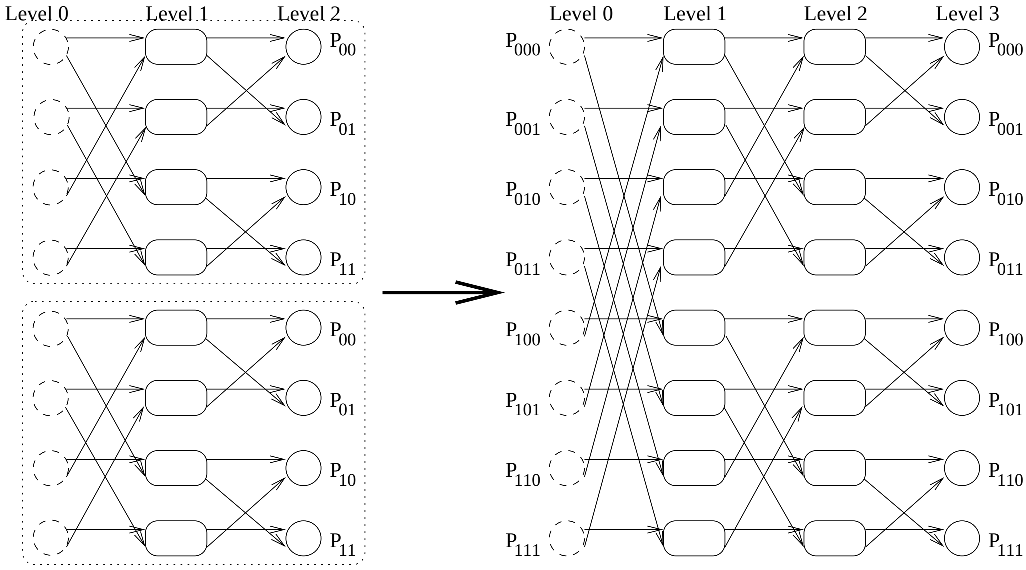


Figure 2: Construction of 3-dimensional $\mathcal{OBF}$ out of two 2-dimensional $\mathcal{OBF}$'s with relabeling of processing nodes and levels.

indicates a routing node, and an arrow between two nodes indicates a link between the nodes. Two attachable subnetworks are called *blocks* of the network. Straight edges are always connecting the output out_{up} from the i'th level router to the input in_{up} of the $i+1$'th level input (in the same block), and cross edges are always connected from the output out_{down} of the i'th level router to the input in_{down} of the $i+1$'th level input (adjacent block).

Our construction has two characteristics. Firstly, straight edges are always leading to the same block and cross edges are always leading to the adjacent block of the (sub)butterfly. Secondly, treatment of packets can be arranged uniformly at each router of the network because of uniform connections between routers.

Nodes $\langle w, 0\rangle$ and $\langle w, r\rangle$ are considered identical processing nodes. A useful property of the r-dimensional butterfly is that for any source/destination processing node pairs $\langle w_s, 0\rangle$ and $\langle w_d, r\rangle$ the packets can be routed by a unique path of length r.

2.3 Feasibility of $\mathcal{OBF}$ as a Systolic Router

The switching time of $LiNbO_3$ switches lies in the range of 10–15 ps [12]. The length of packet (l_p) can be evaluated by equation $l_p = \frac{N_p \times v_c}{B \times r}$, where N_p is the size of the packet in bits, $v_c = 0.3$ m/ns is the speed of light in vacuum, $r = 1.5$ is the refraction index of fiber

[12], and B is the link bandwidth. Assuming the bandwidth to be B=100 Gb/s, the length of a bit in a fiber is $\frac{v_c}{B \times r} = 2$ mm.

In order to estimate the feasibility of a 6-dimensional $\mathcal{OBF}$ (having 64 processing nodes) let us assume the link bandwidth to be $B = 100$ Gb/s, and the size of packets to be $N_p = 128$ b. The corresponding length of a packet in a fiber is $l_p \simeq 256$ mm, and the length of time slot is $t_p \simeq 1.3$ ns. Assuming the length of clock cycle of processing nodes to be $t_{cc} = 1$ ns (corresponding the frequency of 1 GHz), it will take 1.3 clock cycles for a packet to travel between two adjacent routing nodes. The overall amount of fibers is $L_f \simeq 200$ m, and the routing time of packet is $t_r \simeq 8$ clock cycles for each packet. We consider the requested parameters to be reasonable and the architecture to be feasible to construct in the near future.

3 Routing in Optical Butterfly

We have developed a routing algorithm for $\mathcal{OBF}$. In Section 3.1 we present properties of routing information and transitions between blocks. Section 3.2 introduces preprocessing phase. Preprocessing phase consists of determining of the control sequence and determining the routing table that will control the routing. Section 3.3 introduces the routing algorithm for the optical butterfly.

3.1 Properties of Routing

Determining the Routing Information.

Let $a_0 a_1 \ldots a_{r-1}$ ($a_i \in \{0, 1\}$) be a bit sequence indicating the edges used by a packet on its path from the source to the target in an r-dimensional $\mathcal{OBF}$. The value 1 in a bit position a_k indicates that at level k the packet should be routed from router on level k to level $k+1$ using a cross edge leading to the adjacent block. Correspondingly, if $a_k = 0$ the packet should be routed from router on level k to level $k+1$ using a straight edge leading to the same block. Clearly, we can construct an r-ary routing bit sequence for any source/destination pair so that it leads correctly the packet through the $\mathcal{OBF}$. To notice this, let us assume that in a bit sequence $a_0 a_1 \ldots a_k \ldots a_{r-1}$, the k'th bit stands for the edge leading to the wrong subnetwork. We just substitute the initial bit sequence by $a_0 a_1 \ldots \bar{a}_k \ldots a_{r-1}$, where $\bar{a}_k$ is the complement of a_k.

The routing information for packets can be evaluated by the bitwise XOR-operation $\oplus$. For example, if processor P_{011} has a packet destined to processor P_{111}, the routing

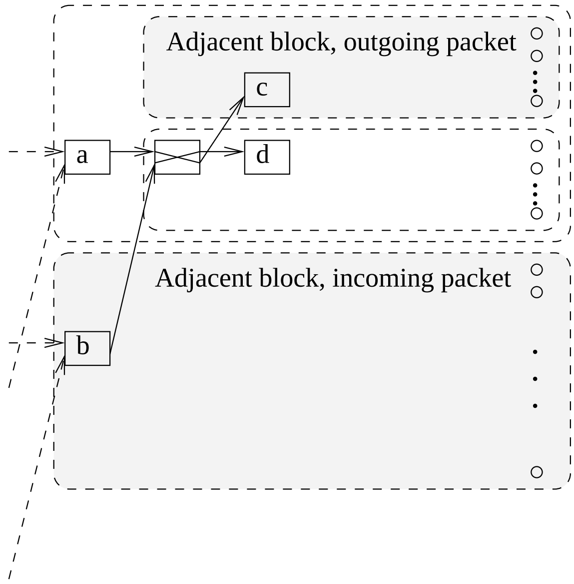


Figure 3: Example of transitions between blocks.

information can be expressed as $011 \oplus 111 = 100$. The meaning of the information is that the packet from P_{011} to P_{111} must be routed from the sender to the first level router using a cross edge, from the first level router to the second level router using a straight edge, and from the second level router to the destination using a straight edge.

Determining Transitions Between Blocks.

An r-dimensional $\mathcal{OBF}$ has $r-1$ levels of routers. According to the recursive contruction of $\mathcal{OBF}$ and our definition every routing node at level r' is connected to two subnetworks of dimension $r - r' - 1$ by two outgoing links. Additionally it is a target of two incoming links from two subnetworks (blocks) whose dimension is $r - r' + 1$, except routing nodes at level 1 that are targets of processors. Figure 3 clarifies the idea of blocks.

Routers can be considered to be an interface between blocks. Let us assume that a packet has bits $\ldots 10 \ldots$ in its $(i-1)$'th and i'th bit positions of routing information. The router responsible to route this packet (at the i'th level) receives the packet from adjacent block into its in_{down} input and it should route the packet to the same block of the $\mathcal{OBF}$. According to our construction the router should be in the invert state. Correspondence between two-bit routing information, transitions between blocks, and required state of router is presented in Table 1.

Because all the routers have two incoming and two outgoing links, each router can

Table 1: Correspondence between two-bit routing information, transitions between blocks, and required state of router.

Routing information	Transition	Required state
00	Same $\rightarrow$ Same	Push
01	Same $\rightarrow$ Adjacent	Invert
10	Adjacent $\rightarrow$ Same	Invert
11	Adjacent $\rightarrow$ Adjacent	Push

route two packets at the same time, if the packets do not prefer the same outgoing link. According to Table 1 this is fulfilled if the incoming packets have either 00 and 11 or 01 and 10 in their $(i-1)$'th and i'th bit position of routing information when they reach a router at level i. Clearly we can see that using these two states it is possible to route any combination of transitions between blocks. Precondition of correct routing is that arrival of packets and the state of router are synchronized correctly.

The transition information for packets can be evaluated by the bitwise XOR-operation $\oplus$. Let $w = s \oplus d$ denote the routing information of a packet from processor P_s to P_d and w_j denote the value at the bit position j of the routing information. We are able to determine a unique transition bit sequence $\vec{\tau}$ by $\vec{\tau_j} = w_j \oplus w_{j+1}, j = 0 \ldots r-1$, where τ_j indicates push/invert state at level $j+1$.

3.2 Initialization Phase

In our contruction injected packets have no routing information. When a packet arrives a routing node it is routed into an adjacent or the same block according to the state of the router. Anyway we are able to arrange a control system so that every packet injected into the $\mathcal{OBF}$ reaches its target. We will use a cyclic control bit sequence and timing of injections of packets.

Determining the Control Sequence.

An r-dimensional $\mathcal{OBF}$ has $r-1$ levels of routing nodes. Packet routing in an r-dimensional $\mathcal{OBF}$ can be implemented by constructing a long control bit sequence $s_0 s_1 s_2 \ldots$, applying at time step t the state corresponding to the value of bit position s_t to all the routing nodes of the $\mathcal{OBF}$, and synchronizing injections of packets so that they reach every routing node in the correct state. Precondition of all-to-all routing is that the bit sequence includes (cyclically) all bit sequences of $l = r - 1$ bits. A naive solution would be to construct the control bit sequence of all l-ary bit combinations. The length of control cycle would be $l2^l$. The control sequence can be reduced to $T = 2^l$ by using *de Bruijn sequences* [3].

A de Bruijn sequence (in alphabet $\mathcal{A} = \{0, 1\}$) of length 2^l is a sequence of 2^l bits in which every subsequence of $l = r - 1$ bits appears once, including wraparound [7]. For $l = 4$, for example, $\vec{\xi} = 0000111101100101$ is a de Bruijn sequence applicable for our purpose. All sixteen 4-bit sequences occur exactly once as subsequence of $\vec{\xi}$.

Fredricksen has presented an algorithm to construct a de Bruijn sequence [2]. The algorithm is *Prefer one* and it can be presented as follows:

Algorithm `Prefer one`

1: `Write` $l = r$ `zeros;`

2: **for** `the` k^{th} `bit of the sequence,` $k > l$`, write a one;`
  if `the newly formed` l`-tuple has not previously appeared in the sequence` **then** $k := k + 1$
  else

3: **for** `the` k^{th} `bit of the sequence, write a zero;`
  if `the newly formed` l`-tuple has not previously appeared in the sequence` **then** $k := k + 1$ `and go to step 2`
  else `stop;`

Bit positions of $\vec{\xi}$ present states of routers of $\mathcal{OBF}$. Let $\vec{\xi}_m$ denote the m^{th} bit of $\vec{\xi}$. At each time step t all the routers are set in push state if $\vec{\xi}_{t \bmod ||\vec{\xi}||} = 0$, where $||\vec{\xi}||$ is the length of de Bruijn sequence, and in invert states otherwise. Determining of the control sequence is necessary to do only once at the initialization phase of the $\mathcal{OBF}$.

Determining the Routing Table.

The optical butterfly has a number of properties. Firstly, structure of routers and connections between them are uniform. Secondly, it is possible to determine a unique routing bit sequence for any packet from a source P_s to the destination P_d for any pair (s, d). Thirdly, determination of unique transitions between blocks is possible as well because of uniformness of the construction of the $\mathcal{OBF}$ and uniqueness of the routing bit sequences. Fourthly, the $\mathcal{OBF}$ is controlled by the static control bit sequence $\vec{\xi}$. For these reasons we are able to determine a routing table for every connection at the initialization phase.

Let us consider an r-dimensional $\mathcal{OBF}$ having $p = 2^r$ processors. For this construction the length of routing bit sequence is $\|w\| = r$, the length of transition bit sequence is $\|\vec{\tau}\| = \|w\| - 1 = r - 1$, and the length of the control sequence is $\|\vec{\xi}\| = 2^{r-1}$. A packet having $w_0 = 0$ is routed correctly if it is injected at time step t into output link out_{up} leading to the same block and during the next $r-1$ time steps we have $\vec{\tau}_{t+i} = \vec{\xi}_{(t+i) \bmod ||\xi||}$, for each $i = 1 \ldots r-1$. At the same time the sending processor can inject another packet into the output link out_{down} for which $w_0 = 1$ and during the next $r-1$ time steps we have $\vec{\tau}_{t+i} = \vec{\xi}_{(t+i) \bmod ||\xi||}$, for each $i = 1 \ldots r-1$. For these two packets destined to processors d and d' stand $d_i = \bar{d}'_i, j = 0 \ldots r-1$, i.e., d' is the complement of d.

At the initialization phase every processor P_i determines a routing table R having $\|\vec{\xi}\| = 2^{r-1}$ rows. Let R_i denote the value of i'th row of the routing table. The algorithm determining routing table is *Routing table* and it can be presented as follows:

Algorithm `Routing table`

`1:` **for** $i = 0$ **to** $i = 2^{r-1}$;

`In the` i`'th row of the routing table` R `write the index value of destination processor for which` $w_0 = 0$ `and` $\vec{\tau}_t = \vec{\xi}_{i+t+1 \bmod ||\xi||}, t = 0 \ldots r-2$`;`

Algorithm Routing table is necessary to execute only once at the initialization phase of the $\mathcal{OBF}$.

3.3 Routing Algorithm for the Optical Butterfly

At the initialization phase each processor determines the control sequence and the routing table. This must be done when the system is set up. At the beginning of routing each processor of the $\mathcal{OBF}$ has a number of packets to send. In the preprocessing phase each processor P_s inserts packets destined to processor P_d into sending buffer $B_{(s,d)}$.

At each time step t each processor s picks up a packet from sending buffer $B_{(s,d')}$, where $d' = R_{t \bmod ||\xi||}$ is the value of ($t \bmod \|\xi\|$)'th position in the routing table. The packet is injected into the outgoing link out_{up} leading to the same block. A packet from sending buffer $\bar{d}'$ is picked up as well and injected into the outgoing link out_{down}.

4 Analysis of Systolic Routing

In the preprocessing phase, each of the h packets of a processor P_i are inserted into sending buffers according to their target. Clearly, all of the packets have been routed after time $O(Tn)$, where T is the maximum size of all buffers. According to Mitzenmacher et al. [10], supposing that we throw n balls into n bins with each ball choosing a bin independently and uniformly at random, then the *maximum load* is approximately $\log n / \log\log n$ with high probability[1]. Maximum load means the largest number of balls in any bin. Correspondingly, if we have n packets to send and n sending buffers during a simulation step, then the maximum load of sending buffers is approximately $\log n / \log\log n$ *whp*. The overall routing time of those packets is $n \log n / \log\log n + \Theta(1)$, which is not work-optimal according to the definition of work-optimality.

If the size of h-relation is enlarged to $h \geq n \log n$, the maximum load is $\Theta(h/n)$ *whp* [11]. Assuming that $h = n \log n$ the maximum load is $\Theta(\log n)$ and the corresponding routing time is $\Theta(n \log n)$. A work-optimal result is achieved according to the definition of work-optimality.

Routing h packets in time $\Theta(h)$ implies work-optimality. We ran some experiments to get an idea about the cost. We ran 5 simulation rounds for each occurrence using a visualizator programmed with Java [4]. Packets were randomly put into output buffers and the average value of the routing time over all the 5 simulation rounds were evaluated. The average cost was evaluated using equation $c_{ave} = \frac{t_r}{h}$, where t_r is the average routing time. Figure 4 gives support to the idea that h does not need to be extremely high to get a reasonable routing cost.

5 Conclusions and Future Work

We have presented a systolic routing protocol for optical butterfly. No electro-optical conversion is needed during the transfer and all the packets injected into the routing machinery are guaranteed to reach their destination. The simple structure presented and the systolic routing protocol are useful and realistic and offer work-optimal routing of h-relation if $h \in \Omega(n \log n)$.

An advantage of our construction is that the overall number of links is $\Theta(n \log n)$. Honkanen presented the systolic routing protocol for Sparse Optical Torus ($\mathcal{SOT}$) is his paper [5]. For $\mathcal{SOT}$, the number of links is $\Theta(n^2)$.

[1] We use *whp, with high probability* to mean with probability at least $1 - O(1/n^\alpha)$ for some constant α.

Routing Costs in Systolic Routing

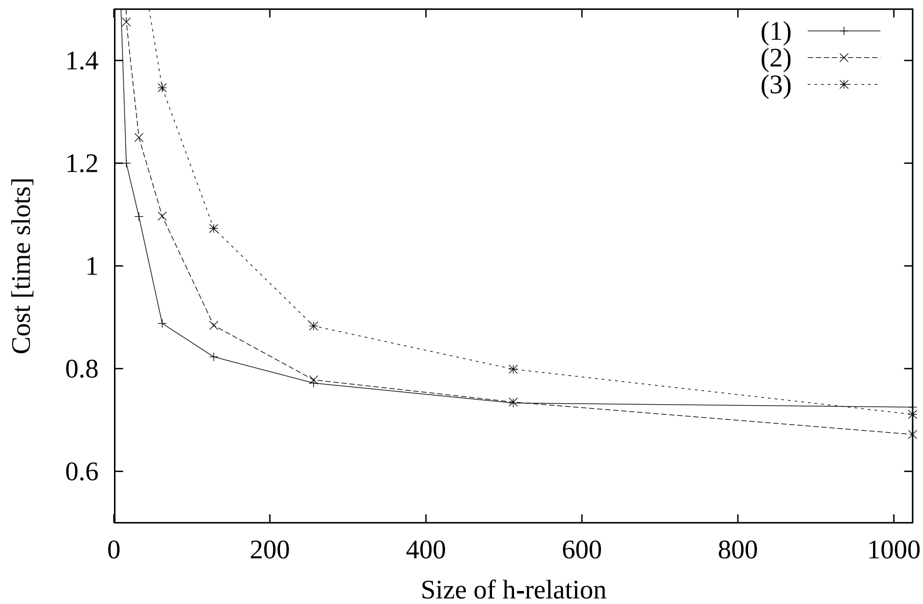


Figure 4: Routing costs, when the size of h-relation varies. (1) $n = 4$, (2) $n = 8$, and (3) $n = 16$.

However, a drawback arise, when the systems are scaled up. Putting M elements in the physical space requires at least a volume of size $\Theta(\sqrt[3]{M})$ [15, 16]. The length of wires between routing nodes increase with respect to the physical space required.

References

[1] Adler, M., Byers, J.W., Karp, R.M.: Scheduling Parallel Communication: The h-relation Problem. *Proceedings of Mathematical Foundations of Computer Science* (MFCS). Prague Czech Republic (1995) 1–20.

[2] Fredricksen, H.: A Survey of Full Length Nonlinear Shift Register Cycle Algorithms. *SIAM Review* **24**,2 (1982) 195–221.

[3] Golomb, S.W.: *Shift Register Sequences*. Aegean Park Press, Laguna Hills California (1982).

[4] Haikarainen, T.: Visualisator for $\mathcal{OBF}$ network. Special project, University of Kuopio, Kuopio. URL: http://www.cs.uku.fi/∼rthonkan/OBF/ (March 30, 2005).

[5] Honkanen, R.T.: Systolic Routing in Sparse Optical Torus. *Proceedings of the 8th Symposium on Programming Languages and Programming Tools* (SPLST'03). Kuopio Finland (2003) 14–20.

[6] Honkanen, R., Leppänen, V., Penttonen M.: Hot-Potato routing Algorithms for Sparse Optical Torus. *Proceedings of the* 2001 *ICPP Workshops*. Valencia Spain (2001) 302–307.

[7] Leighton, F.T.: *Introduction to parallel algorithms and architectures: arrays, trees, hypercubes*. Morgan Kaufmann Publishers Inc., California (1992).

[8] Liu, X., Gu, Q.-P.: Multicasts on WDM All-Optical Butterfly Networks. *Journal of Information Science and Engineering* **18** (2002) 1049–1058.

[9] Maggs, B.M., Sitaraman, R.K.: Simple Algorithms for Routing on Butterfly Networks with Bounded Queues. *SIAM J. Comput.* **28**,3 (1999) 984–1003.

[10] Mitzenmacher, M., Richa, A.W., Sitaraman, R.: The power of randomized choices: A survey of techniques and results. In: *Handbook of Randomized Computing*. Eds. Kluwer (2002).

[11] Raab, M., Steger, A.: "Balls into Bins"—A Simple and Tight Analysis. *Proceedings of 2nd Workshop on Randomize and Approximation Techniques in Computer Science* (RANDOM'98). Barcelona Spain (1998) 159–170.

[12] Saleh, B.E.A., Teich, M.C.: *Fundamentals of Photonics*. John Wiley & Sons Inc., New York (1991).

[13] Upfal, E., Felperin, S., Snir, M.: Randomized Routing with Shorter Paths. *IEEE Transactions on Parallel and Distributed Systems* **7**,4 (1996) 356–362.

[14] Valiant L.G.: General Purpose Parallel Architectures. In Algorithms and Complexity. *Handbook of Theoretical Computer Science* vol. A. Elsevier and MIT press (1990) 943–971.

[15] Vitányi P.B.M., 1988: Locality, Communication, and Interconnect Length in Multicomputers. *SIAM Journal of Computing* **17**,4 (1988) 659–672.

[16] Vitányi P.B.M., 1994: Multiprocessor Architectures and Physical Law. *Proceedings of 2nd Workshop on Physics and Computation* (PhysComp'94). Dallas Texas (1994) 24–29.