Systolic Routing in an Optical Fat Tree

Risto T. Honkanen

Report A/2003/2

ISBN 951-781-266-3

UNIVERSITY OF KUOPIO
Department of Computer Science

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

Systolic Routing in an Optical Fat Tree

Risto T. Honkanen
Department of Computer Science
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. An r-dimensional optical fat tree network ($\mathcal{OFT}$) consists of $2^r - 1$ routing nodes and $n = 2^r$ processing nodes deployed at the leaf nodes of the network. In our construction packets injected into the $\mathcal{OFT}$ carry no routing information. 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 $h \in \Omega(n \log n)$.

Key Words. Optical fat tree, systolic routing, work-optimal routing.

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 [5]. If a parallel algorithm has enough parallel *slackness*, the implementation of shared memory can be reduced to efficient routing of an h-relation [13]. An h-relation is a routing problem where each processing node 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 step, where n is the number of processors. Otherwise slackness cannot be used to "hide" latency influenced by the diameter [5]. For an r-dimensional optical fat tree $\mathcal{OFT}$ having $n = 2^r$ processing nodes, the diameter $\phi = r$ fulfills this condition when $h \in \Omega(n \log n)$.

Using the fat tree topology as an intercommunication network is a cost-effective way to connect a large number of processors. Congestion problems are avoided by providing more bandwidth in the higher levels of the tree. A number of intercommunication networks has been implemented by using fat tree topology. For example, the basic architecture of the Thinking Machine CM-5 data network is a fat tree [8]. High performance clusters typically use fat tree networks as well. For instance the InfiniBand architecture utilizes fat tree [9]. Recent implementations use packet switching as the routing strategy. A drawback of packet switching is that routing decisions must be done in an electronic form. For now we do not know any all-optical implementation of the fat tree network architecture.

In this work we present an all-optical fat tree network architecture and a systolic routing protocol for it. The r-dimensional optical fat tree consists of $2^r - 1$ routing nodes and $n = 2^r$ processing nodes deployed at the leaf 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 its source to the target processing node nor any collisions happen between two distinct packets. An r-dimensional $\mathcal{OFT}$ can route an h-relation in $\Theta(h)$ time, if $h \in \Omega(n \log n)$. Section 2 presents the structure of routing nodes and the structure of an $\mathcal{OFT}$ 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 Fat Tree with Systolic Routers

We study the r-dimensional structure of an $\mathcal{OFT}$ of diameter $\phi = r$ and having $n = 2^r$ processing nodes. Section 2.1 represents the structure of routing nodes. In section 2.2 we introduce the construction of an $\mathcal{OFT}$. Section 2.3 discusses the feasibility of our

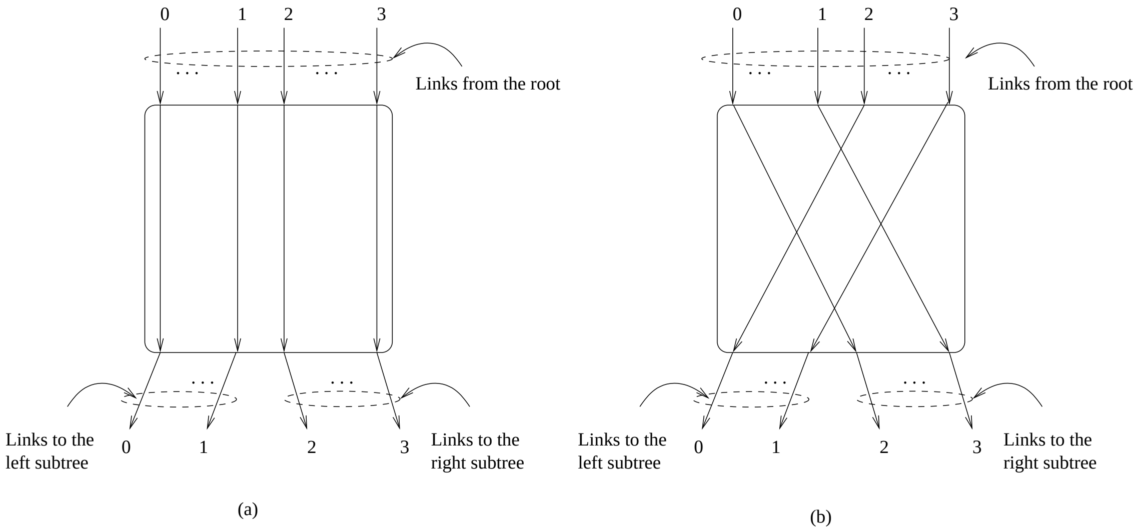

Figure 1: A routing node at level 2: (a) in the drop state, and (b) in the turn state.

construction.

2.1 Systolic Routers for $\mathcal{OFT}$

Routing nodes of an $\mathcal{OFT}$ at the level r' have $\frac{l}{2} = 2^{r'-1}$ outgoing links both to its left and right subtrees and $l = 2^{r'}$ incoming links from its parent node. The in-degree of a routing node equals to the out-degree. Let $\mathcal{I} = \{i_0, i_1, \ldots, i_{l-1}\}$ denote the set of incoming links of a router at level r', and $\mathcal{O} = \{o_0, o_1, \ldots, o_{\frac{l}{2}-1}, o_{\frac{l}{2}}, \ldots, o_{l-1}\}$ denote the set of outgoing links of the router. A routing node can be in two states. When a routing node routes signals from inputs to outputs using mapping $i_s \to o_s$ for all $0 \leq s < l$ it is said to be in *drop* state. Respectively, when a routing node routes signals from inputs to outputs using mapping $i_s \to o_{(s+\frac{l}{2}) \bmod l}$ for all $0 \leq s < l$ it is called to be in *turn* state. An example of a routing node at level 2 in its two possible states is presented in Figure 1.

The basic component of routing nodes is the electrically controlled all-optical 2×2 switch. Switches can be implemented by $LiNbO_3$ technology [12]. We can construct a routing node of any level with edge-disjoint paths, e.g., by using the Beneš network structure [7]. The construction ensures that whatever state is applied, signals never collide.

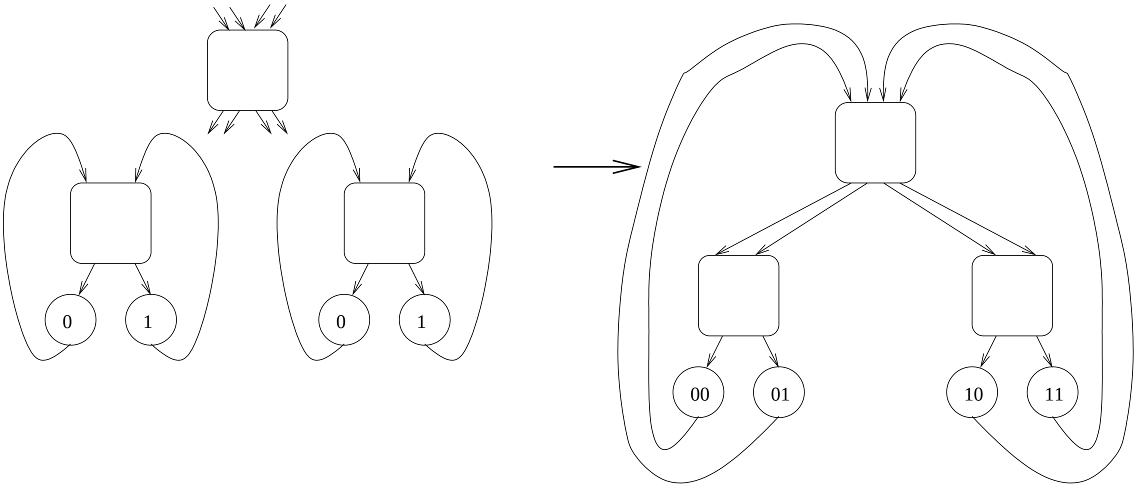

Figure 2: Construction of 2-dimensional $\mathcal{OFT}$ out of two 1-dimensional $\mathcal{OFT}$'s and one routing node of level 2 with relabeling of processing nodes.

2.2 Construction of an Optical Fat Tree

Construction of an $\mathcal{OFT}$ is recursive. A 1-dimensional $\mathcal{OFT}$ consists of $2^1 = 2$ processing nodes and a routing node of level 1 (R_{11}). Processing nodes are connected to the router as its left and right leaves by one outgoing link. Outputs of processing nodes are connected to inputs of the router. A 2-dimensional $\mathcal{OFT}$ can be constructed out of two 1-dimensional $\mathcal{OFT}$'s and a routing node of level 2 (R_{22}). Each processing node P_w is relabeled by a unique r-bit binary string w. Two 1-dimensional $\mathcal{OFT}$'s are connected as left and right subtrees of the routing node R_{22} by using mapping $o_w \rightarrow P_w$ for all $0 \leq w < 2^r$. Outputs of processing nodes are connected to inputs of the routing node R_{22} by using mapping $P_w \rightarrow i_w$ for all $0 \leq w < 2^r$. An example of constructing of 2-dimensional $\mathcal{OFT}$ is presented in Figure 2. In Figure 2, a rounded square indicates a routing node, a circle indicates a processing node, and an arrows between objects indicate unidirectional links.

Respectively, an r-dimensional $\mathcal{OFT}$ can be constructed by two $(r-1)$-dimensional $\mathcal{OFT}$'s and a routing node of level r. The bandwidth of the systems is divided in time slots, whose length t_p equals to the bypass time of a packet via a link between two consecutive routing nodes. We call the length of time slot t_p the *packet cycle*. A packet consists of data bits so that the overall length of the time slot measured in time units is t_p. Each processing node P_i is uniquely labeled by a bit sequence $x_0x_1 \ldots x_{r-1}$, and it has $n = 2^r$ sending buffers $(b_{(i,0)}, b_{(i,1)}, \ldots, b_{(i,n-1)})$ that have an important role in routing.

The number of routing nodes at each level s' is $2^{(r-s')}$, where r is the dimension of $\mathcal{OFT}$. Respectively, the number of links can be calculated by $(r+1)2^r$. Because the number of processing nodes of an $\mathcal{OFT}$ is 2^r and the diameter r, we can conclude that the precondition for work-optimality is satisfied if $h \in \Omega(r)$.

2.3 Feasibility of $\mathcal{OFT}$ with Systolic Routers

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 $\mathcal{OFT}$ (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 $L_f \simeq 115$ m, and the routing time of packets is $t_r \simeq 8$ clock cycles for each packets. We consider the requested parameters to be reasonable and the architecture to be feasible to construct in the near future. A drawback of our construction is that the complexity of routers increases with respect to the dimension of $\mathcal{OFT}$.

3 Routing in Optical Fat Tree

We develop a routing algorithm for $\mathcal{OFT}$. The algorithm can be divided in two phases. During the initialization phase we first construct a control bit sequence that controls the system. Then the routing table is determined. The initialization phase must be executed only once when the system is set up. During the utilization of the $\mathcal{OFT}$ packets are injected into the network so that they are routed level by level to the destination. In section 3.1 we present properties of routing and transitions between subtrees. Section 3.2 introduces the initialization phase of the system. Section 3.3 introduces the routing algorithm for the optical fat tree.

3.1 Properties of Routing

According to our construction an r-dimensional $\mathcal{OFT}$ consists of 2^r processing nodes and r levels of routing nodes. Each routing node has an equal number of incoming and outgoing links. Let us consider a routing node at level s'. It has $2^{r-s'}$ incoming links from its parent node, $2^{r-s'-1}$ links leading to its left subtree, and $2^{r-s'-1}$ links leading to its right subtree. The incoming links can be divided in two groups. Let us denote g_l to be the group of $2^{r-s'-1}$ leftmost incoming links and g_r to be the group of $2^{r-s'-1}$ rightmost incoming links of the routing node.

Let $a_0a_1 \ldots a_{r-1}$ ($a_i \in \{0, 1\}$) be a bit sequence indicating the states of routing nodes used by a packet on its path from the source to the target in an r-dimensional $\mathcal{OFT}$. The value 1 in a bit position $\ldots a_{s'} \ldots$ indicates that at level $r - s'$ the packet using incoming link group g_l or g_r should be routed to the right or left subtree respectively. Correspondingly, the value 0 in a bit position $a_{s'}$ indicates that the packet using incoming link group g_l or g_r should be routed to the left or right subtree respectively. It is obvious that we can construct an r-ary routing bit sequence for any source/destination pair so that it leads the packet correctly through the $\mathcal{OFT}$. To notice this, let us assume that in a bit sequence $a_0a_1 \ldots a_k \ldots a_{r-1}$, the k'th bit stands for the state leading to the wrong subtree. We just substitute the initial bit sequence by $a_0a_1 \ldots \bar{a}_k \ldots a_{r-1}$, 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} (belonging to the leftmost group g_l of the root routing node of a 3-dimensional $\mathcal{OFT}$) has a packet destined to processor P_{111}, the routing information can be expressed as $011 \oplus 111 = 100$. The meaning of this is that the packet from P_{011} to P_{111} must be routed from the leftmost incoming link group to the right subtree at the level 3 routing node, from the rightmost incoming link group to the right subtree at the level 2 routing node, and from the rightmost link to the right subtree at the level 1 routing node. Example of the routing is presented in Figure 3.

Routers can be considered to be an interface between incoming link groups and subtrees. Let us assume that a packet has the the bit $\ldots 1 \ldots$ in its i^{th} bit position. The router responsible to route this packet (at the level $r - i$) receives the packet from the leftmost link group g_l or from the rightmost link group g_r. Regardless of the link group used the router node should be set in turn state. Correspondence between routing bit information, transitions between link groups and subtrees, and required states is presented in Table 1.

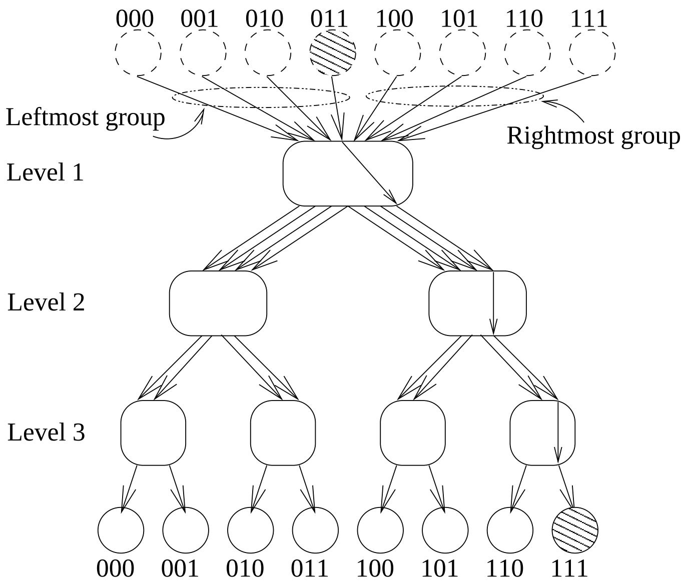

Figure 3: Example of routing of a packet in an $\mathcal{OFT}$.

Table 1: Correspondence between routing bit information, transitions between link groups and subtrees, and the required state of router.

Routing information	Transition	Required state
0	$g_l \to$ Left	Drop
0	$g_r \to$ Right	Drop
1	$g_l \to$ Right	Turn
1	$g_r \to$ Left	Turn

3.2 Initialization Phase

In our construction injected packets carry no routing information. When a packet reaches a routing node it is routed into the left or right subtree according to the state of the router. Anyway we are able to arrange a control system so that every packet injected into the $\mathcal{OFT}$ reaches its target. We will use a cyclic control bit sequence and timing of injections of packets.

Determining the Control Bit Sequence.

An r-dimensional $\mathcal{OFT}$ has r levels of routing nodes. Packet routing in an r-dimensional $\mathcal{OFT}$ 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 $\mathcal{OFT}$, 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 r bits. A naive solution would be to construct the control bit sequence of all r-ary bit combinations. The length of control cycle would be $r2^r$. The control sequence can be reduced to $T = 2^r$ by using *de Bruijn sequences* [3].

A de Bruijn sequence (in alphabet $\mathcal{A} = \{0, 1\}$) of length 2^r is a cyclic sequence of 2^r bits in which every subsequence of r bits appears once and the first bit is considered to follow the last [7]. For $r = 4$, 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 $\mathcal{OFT}$. That is, let $\vec{\xi}_m$ denote the value the m^{th} bit of de Bruijn sequence $\vec{\xi}$. At each time step t all the routing nodes are set in turn state if $\vec{\xi}_{t \bmod ||\vec{\xi}||}$ = 1, where $||\vec{\xi}||$ is the length of $\vec{\xi}$, and in drop state otherwise. The control bit sequence needs to be constructed only once at the initialization phase of the $\mathcal{OFT}$.

Determining the Routing Table.

The optical fat tree has a number of properties. Firstly, the structure of routing nodes and connections at each router level 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 link groups and subtrees is possible as well because of uniformness of the construction of the $\mathcal{OFT}$ and uniqueness of the routing bit sequences. Fourthly, the $\mathcal{OFT}$ 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 $\mathcal{OFT}$ having $p = 2^r$ processors. For this construction the length of routing bit sequence is $\|w\| = r$ and the length of control sequence is $\|\vec{\xi}\| = 2^r$. A packet is routed correctly if it is injected into the network so that during the next r time steps we have $\vec{\tau}_t = \vec{\xi}_{t \bmod \|\xi\|}, t = 0 \ldots r-1$, where $\vec{\tau}$ is the address bit sequence of the destination.

At the initialization phase every processor P_i determines a routing table R having $\|\vec{\xi}\| = 2^r$ rows. Let R_j denote the value of j'th row of the routing table. Each row $R_{t \bmod \|\xi\|}$ contains the valid address bit sequence of the destination processor at time t. The algorithm determining routing table is *Routing table* and it can be presented as follows:

```
**Algorithm** Routing table
  {Assuming $s$ and $d$ are the source and the destination
  processors, and $\vec{\xi}$ is the control sequence};
  **for** $i = 0$ **to** $i = 2^r - 1$
      In the $i$'th row of routing table $R$ write the index
      value of destination processor for which
      $\vec{\tau}_t = \vec{\xi}_{i+t+1 \bmod \|\xi\|}, t = 0 \ldots r-1$;
```

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

3.3 Routing Algorithm for the Optical Fat Tree

At the initialization phase each processor determines the control sequence $\vec{\xi}$ and the routing table. This must be done when the system is set up. At the beginning of routing each processor of the $\mathcal{OFT}$ 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 row in the routing table and inject it into the outgoing link. The r-tuple of bits starting at $\vec{\xi}_t$ then indicates the successive drop and turn states that correctly route the packet to the target processor d'.

At each time step each processor sends at most one packet to the root router node along a distinct link. Since the routers realize a one-to-one mapping of the incoming links to the outgoing links, there are no collisions in the outgoing links either. The same holds inductively through all levels of the $\mathcal{OFT}$, which means that there are no collisions in the entire network.

4 Analysis of Systolic Routing

In the preprocessing phase, each of the h packets of a processing node P_s was inserted into sending buffer $b_{(s,d)}$, where P_d is the target of the packet. Clearly, all of the packets have been routed after time $O(Tn)$, where T is the maximum size of all buffers and $n = 2^r$ is the number of processing nodes. The result is poor if the packets have an odd distribution over targets. In this presentation we assume that packets have an even distribution over targets.

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 (*whp*)[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 $n \log n / \log \log n + \Theta(1)$ that 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 $\Theta(h/n)$ [11]. Assuming that $h = n \log n$ the maximum load is $\Theta(\log n)$, 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. Intuitively it is clear that the cost approaches to 1, when h/n grows.

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 [6]. Packets were randomly created and put into output buffers and the average value of the routing time over all the 5 simulation rounds were evaluated. The results are only speculative because of a

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

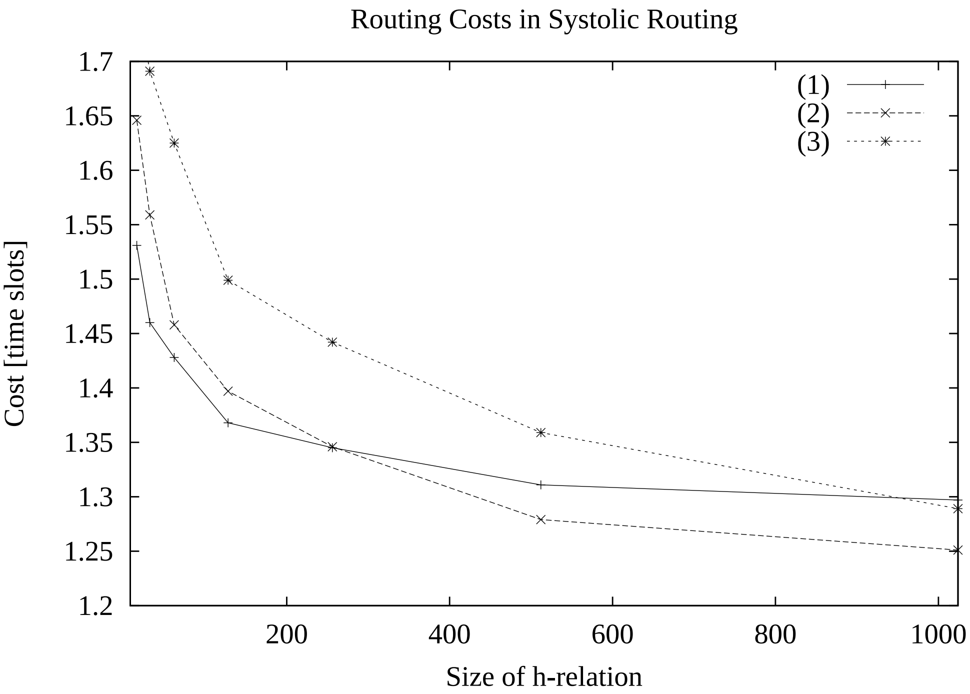

Figure 4: Routing costs, when the size of h-relation varies. (1) $n = 4$, (2) $n = 8$, and (3) $n = 16$.

small number of evaluation rounds executed. 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 the systolic routing protocol for optical fat tree. 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 $\Theta(n \log n)$. We presented the systolic routing protocol for sparse optical torus ($\mathcal{SOT}$) in paper [4]. For $\mathcal{SOT}$, the number of links is $\Theta(n^2)$.

However, a couple of drawbacks arise, when the systems are scaled up. Firstly, the degree of root node the $\mathcal{OFT}$ increases with respect to the size of network. Secondly, putting M elements in the physical space requires at least a volume of size $\Omega(\sqrt[3]{M})$ [14, 15]. The length of wires between routing nodes increases with respect to the physical space required.

Acknowledgments

I express my gratitude to professor Martti Penttonen for his valuable discussions on the topic and his guidance on the structure of the text.

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] 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.

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

[6] Koivistoinen, A., Pietarinen, K., Rantonen, A., Valo T.: Visualisator for $\mathcal{OFT}$ network. Programming project, University of Kuopio. Kuopio Finland URL: http://www.cs.uku.fi/∼rthonkan/OFT/Laski.htm (March 30, 2005).

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

[8] Leiserson, C.E., et al.: The Network Architecture of the Connection Machine CM-5. *Proc.* 4th *Ann. Symp. Parallel Algorithms and Architectures*. New York USA (1992) 272–285.

[9] Mellanox Technologies Inc.: InfiniBand Clustering — Delivering Better Price/Performance than Ethernet. White paper, Mellanox Technologies Inc., Santa Clara California (2005).

[10] Mitzenmacher, M., Richa, A.W., Sitaraman, R.: The power of randomized choices: A survey of techniques and results. To appear in: *Handbook of Randomized Algorithms*. URL: http://www.eecs.harvard.edu/∼michaelm/ (June 24, 2002).

[11] Raab, M, Steger, A.: "Balls into Bins"—A Simple and Tight Analysis. *Proceedings of* 2nd *Workshop on Randomize and Approximation Techniques on 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 USA (1991).

[13] Valiant, L.G.: General Purpose Parallel Architectures. In: *Algorithms and Complexity, Handbook of Theoretical Computer Science* volume A (1990) 943–971.

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

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