Adaptive Load Balanced Routing for 2-Dilated Flattened Butterfly Switching Network

Ajithkumar Thamarakuzhi   John A. Chandy

Department of Electrical & Computer Engineering
University of Connecticut, Storrs, CT USA 06269–2157
{ajt06010,chandy}@engr.uconn.edu

***Abstract*—High-radix networks such as folded-Clos outperform other low radix networks in terms of cost and latency. The 2-dilated flattened butterfly (2DFB) network is a nonblocking high-radix network with better path diversity and reduced diameter compared to the folded-Clos network. In this paper, we introduce an adaptive load balanced routing algorithm that is designed to exploit all the positive topological properties of a 2DFB network. The proposed algorithm achieves load balance by allowing one non minimal forwarding in each dimension in case of network congestion. This algorithm provides high throughput on adversarial traffic patterns and provides better latency on benign traffic patterns. We have compared the performance of our algorithm on a 2DFB network with an Adaptive Clos algorithm on a folded-Clos network and a Minimal routing algorithm on a 2DFB network for different traffic patterns. We observed that 2DFB network with the proposed algorithm provides the same throughput with reduced latency compared to the folded-Clos network with an Adaptive Clos algorithm for all the traffic patterns.**

***Keywords*-Routing; adaptive; switching architecture;**

I. Introduction

High performance computing on distributed memory parallel processing systems such as clusters are very dependent on communication between processing nodes. As a result, the interconnection network that connects these nodes is a critical part of the performance of the system. For the past few decades, we have seen improving performance of processors and memory systems. In order to keep up with this, the network switch performance must also improve. The study of interconnection networks has a long history and a large number of network topologies and routing algorithms have been studied by researchers. Among these networks, hypercube [1] and Clos [2] (or its derivatives) are the most popular networks.

The technological progress in modern ASICs has led to the availability of routers with high bandwidth in the range of Tb/s. The improved pin bandwidth of these routers can be efficiently used to construct high-radix network topologies. Recent work has shown that high-radix network outperforms corresponding low-radix network in terms of cost and latency. Folded-Clos and flattened butterfly [3] are two topologies which can take advantage of the high-radix routers. The 2-dilated flattened butterfly (2DFB) is a nonblocking version of a flattened butterfly network. In previous work [4], [5], we have introduced the 2DFB network and proved its nonblocking behavior and shown the implementation of a 2DFB network switch using the NetFPGA platform. In this paper we propose an adaptive load balanced algorithm for 2DFB and observe its performance for different traffic patterns.

A routing algorithm can be considered as optimal if it provides low latency on local traffic and high throughput on adversarial traffic. Most algorithms must compromise one goal in order to achieve the other. Minimal routing, which always chooses the shortest path for each packet, provides minimum latency for local and benign traffic. However, it provides non acceptable latency for adversarial traffic due to load imbalance. In order to improve the throughput in adversarial traffic, the routing algorithm should balance the load by sending some fraction of packets over non-minimal paths.

Researchers have been trying to address the issue of providing high worst-case performance while preserving locality. Valiant's randomized algorithm [6] gives good performance in worst case traffic but very poor performance for local traffic in terms of latency. Minimal adaptive routing [7] [8] suffers from global load imbalance. GOAL is a load balanced adaptive routing algorithm designed for a torus network [9]. It provides better load balance with improved performance for local traffic. It achieved 58% throughput of the Minimal algorithm on nearest neighbor traffic for a torus network. Adaptive Clos [10] is an adaptive routing algorithm designed for Clos network which provides optimum performance for a high-radix Clos network. The adaptive routing algorithm that we propose in this paper is designed for a 2DFB network and it balances the load efficiently by allowing one non-minimal forwarding in each dimension in case of traffic congestion. It senses the traffic congestion from the packet queue. We observed the performance of this algorithm for local traffic and it has reduced latency than a Clos network with the Adaptive Clos algorithm.

The remainder of the paper is organized as follows.

In Section II we briefly describe 2DFB and few of its topological properties. Section III describes the proposed adaptive load balanced algorithm for 2DFB network. In Section IV we present the simulation results and we conclude in Section V.

II. Background

In this section we describe the 2DFB network [4] and its topological properties.

A. 2-dilated flattened butterfly structure

A 2DFB network is derived from a flattened butterfly structure [3] by either duplicating all the interconnecting links between the switching elements or replacing it with links of double bandwidth. Links between the end-terminals and switching elements remain the same. A 2DFB is composed of N/k routers of radix k'=$n(k-1)+1$ where N is the number of end-terminals in the network, n is the number of columns in a butterfly network, k is the number of end-terminals connected to each router and the radix(k') is the number of external ports associated with each router. The routers are connected by channels in $n' = n-1$ dimensions. In each dimension d, from 1 to n', router i is connected to each router j given by

$$j = i + [m - (\lfloor \frac{i}{k^{d-1}} \rfloor \mod k)]k^{d-1} \qquad (1)$$

for m from 0 to $k-1$, where the connection from i to itself is omitted. For example a 4-ary 2-dimensional 2DFB for N=64 is shown in Fig. 1.

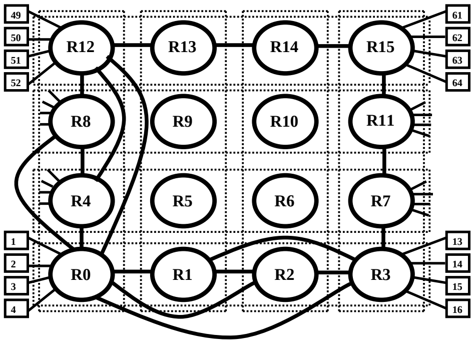

Figure 1. 4-ary flattened butterfly structure

As we can see in the Fig. 1, each switching element is connected to k end-terminals (here k=4). k switching elements in each row are interconnected and it can be considered as a 1-dimensional system. A 1-dimensional system is a fully connected ring structure with each link having double bandwidth. Its bisectional bandwidth is $N/2$ where N is the total number of end-terminal connected to the 1-dimensional system. In [4] we have proved that in a 1-dimensional 2DFB system any routing permutation can be performed without conflict using a maximum of two links. Higher dimensional 2DFB systems are constructed by combining 1-dimensional systems as shown in Fig. 1. For a k-ary d-dimensional [d=$(log_k N)-1$] 2DFB system with a network size (N) of power of k, the bisection bandwidth is $((k^2/2)(k^{d-1}))$ which is equal to $N/2$ (same as that of a hypercube network). Therefore, a properly designed routing algorithm can route any permutation without conflict by making use of a maximum of $2d$ hops (2 hops in each dimension).

B. Network diameter

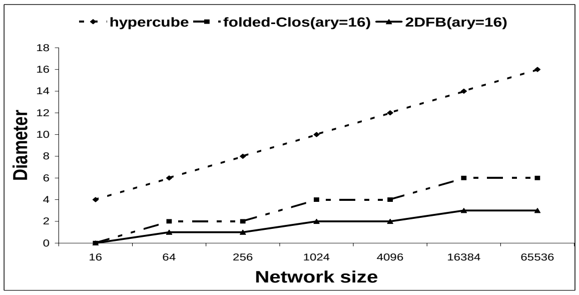

Figure 2. Network diameter

Network diameter is a measure of shortest distance between the source and destination nodes. Since high priority traffic can be routed through this shortest path, the network diameter plays an important role in a multi-processor communication system. A comparison of network diameter of 2DFB with other topologies for different network size is shown in Fig. 2. The diameter of a hypercube network is $log_2 N$, the diameter of a k-ary folded-Clos network is $2\{\lceil(log_k N)\rceil - 1\}$ and the diameter of a k-ary 2DFB is $\lceil(log_k N)\rceil - 1$. As we can observe, 2DFB has the smallest network diameter compared to other network topologies.

C. Number of hops

Message latency in a network is proportional to the number of hops required for routing the message. Fig. 3 represents the number of hops needed for routing the message for different network topologies with varying network sizes. Number of hops required in 2DFB is not same as the network diameter for all source-destination pair. For example in Fig. 1 if end-terminals 1,2,3 and 4 are sending messages to end-terminals 5,6,7 and 8 respectively with full bandwidth, then only messages from terminal 1 and 2 can be routed through the direct link between R_0 and R_1 and the messages from 3 and 4 should be routed through R_2 or R_3. In this case the number of hops required in the worst case is 2. In higher dimension 2DFB, in worst case, 2 hops

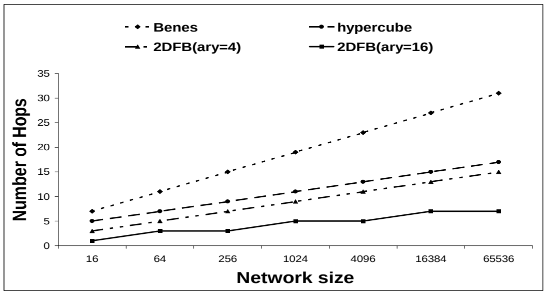

Figure 3. Number of hops needed for routing

are required for routing the message in each dimension. The dimension of a k-ary 2DFB is $\lceil(log_k N)\rceil - 1$. So the number of hops required (worst case) to complete any routing request in a k-ary 2DFB for $k \gg 2$ is $2\{\lceil(log_k N)\rceil - 1\}$. For $k = 2$, 2DFB becomes a normal hypercube structure with 2 end-terminals connected to each switching element and with interconnecting links of double bandwidth. In this case the number of hops required is $log_2(N/2)$. In [11] it is shown that the number of hops required in a hypercube network is $log_2 N$. The number of hops required for a k-ary folded-Clos network is $2\{\lceil(log_k N)\rceil - 1\}$. From the comparison we can see that the number of hops required for a k-ary 2DFB in the worst case is the same as that of a k-ary folded-Clos network. Unlike folded-Clos, in 2DFB the number of hops required is not same for all the source-destination pair. Large amount of source-destination pair need only one hop to traverse in one dimension. So the average number of hops in a 2DFB will be always less than that of the corresponding folded-Clos network. Thus it is clear that k-ary 2DFB provides better message latency than corresponding folded-Clos network.

D. Cost Analysis

A key determinant of the effectiveness of a network topology is the cost of the network relative to the performance it delivers. Cost of the network is decided by cost of routers and links. The number of switching elements and links required to implement a 2DFB network is less than other nonblocking networks such as folded-Clos and hypercube and therefore the implementation cost of a 2DFB network will be lesser than other nonblocking networks [4].

III. Routing Algorithm

The proposed routing algorithm is designed to explore the topological properties of a 2DFB network. A 2DFB network is similar to a k-ary generalized hypercube (GHC) except that in a 2DFB k end-terminals are connected to each switching element. A 2DFB can be considered as a k-way bristled 2-dilated GHC. If r is the dimension of a k-ary flattened butterfly, then there will be k^r nodes(switching elements) in the system and each node can be represented using a r-digit number, i.e. any node $x = x_{r-1}...x_i...x_0$ where $x_i \in [0, k-1]$. In a 2DFB network any two nodes, whose numbers differ only in the ith digit, are joined by a duplex channel and it is known as the ith dimension channel. Thus by comparing the r bit number associated to the current switching element and the destination switching element, one can find out the set of dimensions in which forwarding of the packet is required. Every node contains $(k-1)$ channels in each dimension.

The proposed routing algorithm has two phases of operation, minimal forwarding phase and non-minimal forwarding phase. In the minimal phase, the algorithm considers the set of dimensions in which forwarding is required and it adaptively selects the dimension if the direct link in the selected dimension is ready to use. We are using a sequential allocation method in our algorithm which gives maximum performance. If no direct link is available in any of the selected dimension in the minimal phase, then the algorithm will turn in to non-minimal phase of operation.

In non-minimal phase the algorithm will consider all selected dimension and adaptively check the availability of any of the non-minimal link in the selected dimension. If it finds any available non-minimal link, the packet will be forwarded to that link. We constrain this non-minimal forwarding by adding one bit flag in the header of each packet and we call this flag as the priority flag. The algorithm allows only one non-minimal forwarding in each dimension. If the switching element sees that the priority flag is set for the received packet, then that packet will be sent to a minimal direct link even though all minimal output queue have packets more than the threshold level. In the next cycle some portion of the traffic coming from the other switches will be adaptively rerouted to any non-minimal link which will reduce the traffic congestion. Thus, by the combined use of minimal and non-minimal phase of operation the algorithm will balance the load efficiently and it will reach the steady state within a few iterations.

The algorithm always gives priority to the minimal forwarding and therefore for local traffic and benign traffic, the performance of this algorithm will be very close to the minimal routing. With the worst case traffic the algorithm will use at most two links per dimension. In the worst case also a fraction of traffic is routed through direct links. So the average latency will be still less than that of a Adaptive Clos algorithm in a Clos network.

A. Algorithms used for comparison

We have selected Minimal and Adaptive Clos routing algorithms for the performance comparison with our proposed adaptive algorithm. The Minimal algorithm will always route packets in the shortest path. Adaptive Clos routing have forward and backward phases. In the forward phase any of the output queue in the forward path is adaptively selected by considering the number of packets in each output queue. In the reverse phase routing is deterministic as there exists only a single path to the destination. The Minimal routing algorithm is implemented in a 2DFB and the Adaptive Clos routing is implemented in Clos networks.

B. Terminologies used in the algorithm

The proposed adaptive routing algorithm is shown in the Algorithm 1. A one bit flag is added to the header of each packet to indicate the switching priority and it is represented as h_1. An output port is selected by considering the number of packets already in queue in the corresponding output queue. The port is selected if the number of packets in the output queue is less than the threshold value T_h. The preferred output ports are also decided by comparing the r digit representation of the current switching element and the destination switching element, where r is the dimension of the network. r digit representation of current switching element and destination switching element is represented as $s_d[r]$ and $d_d[r]$ respectively. $dimsel$ is a pointer to the selected dimension and P_s is a flag indicating whether a port is selected or not.

IV. Results

We have modeled 2DFB and folded-Clos networks for different network sizes using the OMNeT++ simulation library [12]. These topologies are implemented using interconnecting links of 2 Gb/s bandwidth. All the end-terminals are sending packets with a maximum bandwidth of 1 Gb/s. We have used a packet size of 121 bytes. Higher size packets are also following the same trend. The default OMNeT switch model was modified in order to include a 2 Gb/s channel. We have compared the throughput and latency of these network topologies for different traffic patterns. We assume that the data transmission through the network is permutation type - i.e. a unique source and destination are assigned to any data element and the elements are permuted upon transmission. We have selected three traffic patterns to consider the best case and worst case scenario of 2DFB topology which are named as below.

1) $Benign$: In a 2DFB structure each switching element is connected to $k-1$ switching elements using direct links in each dimension. In $benign$ traffic pattern all the traffic can be routed through these directed links, that is in this pattern the number of hops required for the routing of any packet will be equal to the diameter of the 2DFB network. In this pattern each pair of end-terminals connected to a switching element will be sending traffic to different directly connected switching elements. 2DFB provides minimum latency for $benign$ traffic pattern.

2) $Adversarial$: In this traffic pattern all the end-terminals connected to a switching element S_i will be sending traffic to end-terminals which are connected to another single switching element S_{i+j}. If this pattern is used in a 2DFB only two end-terminals which are connected to a switching element can send traffic through the direct link. All the other $k-2$ end-terminals should send traffic through indirect links. 2DFB provides worst case latency for $adversarial$ traffic pattern.

3) $Random$: In this pattern destination terminals are selected randomly. Latency provided by 2DFB for this pattern will be between that of $benign$ and $adversarial$ patterns.

A. Throughput comparison

We have compared the average throughput of a 8-ary 1-dimensional network with a network size of 64 for three different routing algorithms, Minimal, Adaptive Clos and our proposed algorithm which is named as Adaptive 2DFB. Minimal and Adaptive 2DFB algorithms are implemented over a 2DFB network. Adaptive Clos routing algorithm is implemented over a Clos network with the same size.

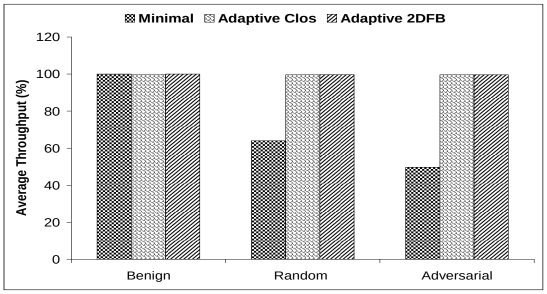

Figure 4. Throughput comparison of 1-dimensional networks

The throughput comparison is done for three different traffic patters as mentioned before. As shown in Fig. 4, like the Clos network, 2DFB also provides throughput which is very close to 100% for all the given traffic patterns. The Minimal algorithm provides 100% throughput only for benign traffic and it provides 50% throughput for adversarial traffic pattern. This shows the benefit of our adaptive algorithm as it is able to maintain high throughput in both adversarial and benign traffic patterns.

We have also compared the average throughput of a 8-ary, 2-dimensional 2DFB with 8-ary, 2-dimensional Clos network. Both of the network have a network size of 512. The throughput comparison is shown in Fig. 5. Two dimensional network also provides similar average throughput as one dimensional network.

B. End-to-end packet delay comparison

We have compared the average end-to-end packet delay of a 8-ary 1-dimensional and 2-dimensional networks for dif-

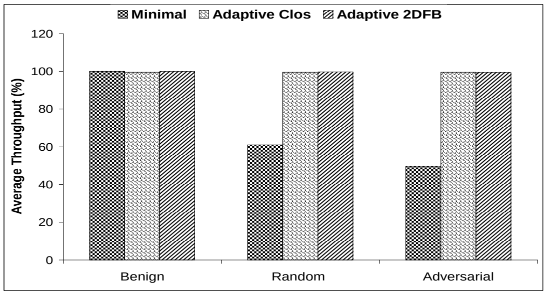

Figure 5. Throughput comparison of 2-dimensional networks

ferent routing algorithms. Average end-to-end packet delay comparison of 8-ary 1-dimensional networks with a network size of 64 is shown in Fig. 6.

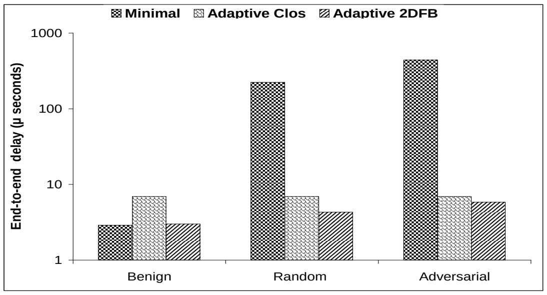

Figure 6. End-to-end packet delay comparison of 1-dimensional networks

In Fig. 6 we can notice that the average end-to-end packet delay of 2DFB for benign traffic pattern is less than that of adversarial traffic pattern. We can also notice that the average end-to-end packet delay of the Adaptive 2DFB algorithm is less than that of Adaptive Clos algorithm for all the traffic patterns. As would be expected, the Minimal algorithm shows poor load balancing and provides very high packet delay for adversarial traffic pattern compared to other algorithms.

Average end-to-end packet delay comparison of 8-ary, 2-dimensional networks with a network size of 512 is shown in Fig. 7. We can observe that 2-dimensional networks also follow the same trend as 1-dimensional networks. Both the 1-dimensional and 2-dimensional 2DFB networks with Adaptive 2DFB algorithm provide maximum end-to-end packet delay for adversarial traffic pattern. This maximum value is still less than corresponding Clos network with Adaptive Clos algorithm. Practical traffic patterns will be random in nature and the end-to-end packet delay of Adaptive 2DFB algorithm in a 2DFB network, for the random traffic pattern will be in between the end-to-end packet delay of benign and adversarial traffic patterns. This end-to-end packet delay comparison reveals the effectiveness of the proposed Adaptive 2DFB routing algorithm on 2DFB networks. This comparison also shows the benefit of the 2DFB architecture with respect to Clos in that maintains high throughput with lower latency costs than the more expensive Clos architecture.

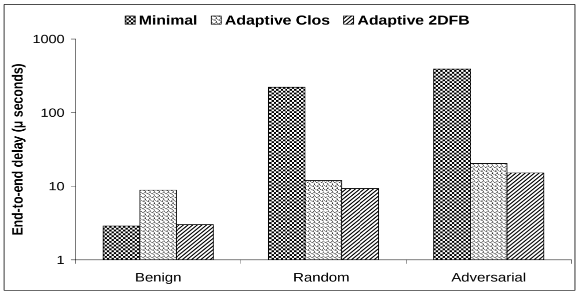

Figure 7. End-to-end packet delay comparison of 2-dimensional networks

V. Conclusion

In this paper, we have introduced an adaptive load balanced routing algorithm for 2DFB switching network. The proposed algorithm is designed to exploit the nonblocking property of 2DFB network. The algorithm also takes full advantage of the reduced diameter of 2DFB network. It provides better load balancing by allowing one non-minimal forwarding in each single dimension of 2DFB which is a 2-dilated fully connected ring structure. This algorithm also provides good performance for local and benign traffic by providing priority to the selection of direct links. We have compared the performance of the proposed algorithm running over a 2DFB with the Adaptive Clos algorithm running over a Clos network and Minimal routing algorithm over a 2DFB network, and we have observed that our algorithm provides reduced latency for all the traffic patterns while maintaining the same throughput of the Adaptive Clos algorithm. Thus, we conclude that the 2DFB with the proposed algorithm will be an optimal candidate for a high performance interconnection system with reduced cost.

Acknowledgment

This work was supported in part by a National Science Foundation High End Computing University Research Activity grant (award number CCF-0621448). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

Algorithm 1: Adaptive routing algorithm.

```
begin
    if $s_d == d_d$ then
        $P_s$ = 1
        port = read (port for the end-terminal)
        send(frame, port)
    else
        % Minimal forwarding phase
        Set i= msb and db=0
        dimsel= new int[r]
        repeat
            if $s_d[i] == d_d[i]$ then
                goto deci
            else
                *(dimsel+db)=i
                port = read (direct port)
                if $h_1 == 1$ then
                    chk:if port == input port then
                        db = db+1 and goto deci
                    else
                        set $h_1$ to 0 and $P_s$ to 1
                        send(frame, port)
                        goto sel0
                    end
                else
                    if packets in queue $<= T_h$ then
                        goto chk
                    else
                        db=db+1 and goto deci
                    end
                end
            end
            deci: $i = i - 1$
        until $i >= 0$
        sel0:if $P_s$ == 0 then
            % Non-minimal forwarding phase
            s= ary-2
            for $b \leftarrow 0$ to $db$ do
                dims = *(dimsel+b)
                repeat
                    port = (dims*(ary-1))+s
                    if port == input port then
                        goto decrement
                    else
                        if packets in queue $<= T_h$ then
                            Set $h_1$ to 1 and $P_s$ to 1
                            send(frame, port)
                            break
                        end
                        goto decrement
                    end
                    decrement: $s = s - 1$
                until $s >= 0$
            end
        end
    end
end
```

References

[1] L. N. Bhuyan and D. P. Agrawal, “Generalized hypercube and hyperbus structures for a computer network,” *IEEE Trans. Computers*, vol. 33, no. 4, pp. 323–333, 1984.

[2] C. Clos, “A study of non-blocking switching networks,” *The Bell System Technical Journal*, vol. 32, pp. 406–424, Mar. 1953.

[3] J. Kim, W. J. Dally, and D. Abts, “Flattened butterfly: A cost-efficient topology for high-radix networks,” in *Proc. of the International Symposium on Computer Architecture (ISCA)*, pp. 126–137, June 2007.

[4] A. Thamarakuzhi and J. A. Chandy, “2-dilated flattened butterfly: A nonblocking switching network,” in *International Conference on High Performance Switching and Routing (HPSR 2010)*, June 2010.

[5] A. Thamarakuzhi and J. A. Chandy, “Design and implementation of a nonblocking 2-dilated flattened butterfly switching network,” in *IEEE Latin-American Conference on Communications 2010*, 2010.

[6] L. G. Valiant and G. J. Brebner, “Universal schemes for parallel communication,” in *In Proc. of the ACM Symposium on the Theory of Computing*, pp. 263–277, 1981.

[7] L. Gravano, G. Pifarre, G. Pifarre, P. Berman, and J. Sanz, “Adaptive deadlock- and livelock-free routing with all minimal paths in torus networks,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 5, no. 12, pp. 1233–1252, 1994.

[8] D. Linder and J. Harden, “An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes,” *ACM Trans. on Computer Systems*, vol. 40, no. 1, pp. 2–12, 1991.

[9] A. Singh, W. J. Dally, A. K. Gupta, and B. Towles, “Goal: A loadbalanced adaptive routing algorithm for torus networks,” in *In Proc. of the International Symposium on Computer Architecture*, pp. 194–205, June 2003.

[10] J. Kim, W. J. Dally, and D. Abts, “Adaptive routing in high-radix clos network,” in *In International Conference for High Performance Computing, Networking, Storage, and Analysis (SC06)*, 2006.

[11] Z. Liu and D. W. . Cheung, “Oblivious routing for lc permutations on hypercubes,” *Parallel Computing*, no. 25, pp. 445–460, 1999.

[12] Varga, András, “The OMNeT++ discrete event simulation system,” *Proceedings of the European Simulation Multiconference (ESM’2001)*, 2002.