Proc. Parallel Computing, Routing and Communication Workshop (PCRCW'97), Atlanta, June 1997.

Preliminary Evaluation of a Hybrid Deterministic/Adaptive Router

Dianne Miller and Walid A. Najjar
Department of Computer Science
Colorado State University
Ft. Collins, CO 80523 USA
{*najjar, millerdi*}*@cs.colostate.edu*

Abstract

A novel routing scheme is proposed for virtual cut-through routing that attempts to combine the low routing delay of deterministic routing with the flexibility and low queuing delays of adaptive routing. This hybrid routing mechanism relies on a pipelined implementation where different paths and stages of the router are used for different routing modes. A simulation based experimental evaluation of these three schemes shows that the hybrid scheme does indeed achieve its objectives.

1 Introduction

This paper reports on the preliminary results of the evaluation of a hybrid deterministic and adaptive routing algorithm. The objective of this new approach to routing is to combine the advantages of both models.

In deterministic, or dimension-order, routing, a message is routed along decreasing dimensions with a dimension decrease occurring only when zero hops remain in all higher dimensions. Virtual channels are included in the router to avoid deadlock [5]. However, deterministic routing algorithms can suffer from congestion since only a small subset of all possible paths between a source and destination are used.

In adaptive routing, messages are not restricted to a single path when traveling from source to destination. Moreover, the choice of path can be made dynamically in response to current network conditions. Such schemes are more flexible, can minimize unnecessary waiting, and can provide fault-tolerance. Several studies have demonstrated that adaptive routing can achieve a lower latency, for the same load, than deterministic routing when measured by a constant clock cycle for both routers [12, 14].

The delay experienced by a message at each node can be broken down into: *routing* delay and *queuing* delay. The former is determined primarily by the complexity of the router. The later is determined by the congestion at each node which in turn is determined by the degrees of freedom the routing algorithm allows a message. The main performance advantage of adaptive routing (besides its fault-tolerance) is that it reduces the queuing delay by providing multiple path options.

However, the routing delay for deterministic routers, and consequently their corresponding clock cycles, can be significantly lower than adaptive routers as pointed out in [3, 1]. This difference in router delays is due to two main reasons:

- *Number of virtual channels:* Two virtual channels are sufficient to avoid deadlock in dimension ordered routing [5]; while adaptive routing (as described in [8, 2]) requires a minimum of three virtual channels in k-ary n-cube networks.

- *Output channel selection:* In dimension-ordered routing, the output channel selection policy is very simple: it depends only on information contained in the message header itself whereas in adaptive routing the output channel selection policy depends also on the state of the router (i.e the occupancy of various virtual channels) causing increased router complexity and thereby higher routing delays.

The results reported in [3, 1] show that the router delays for adaptive routers are about half to more than twice as long as the dimension-order router for worm-hole routing. These results, however, do not account for the advantage of adaptive routing in reducing queuing delays in the nodes between source and destination. Furthermore, the various routing algorithms evaluated, both deterministic and adaptive, require a variable amount of resources such as buffer area or physical channels between nodes. In [9], the

advantage of adaptive routing in reducing queuing delays in the nodes between source and destination is accounted for in worm-hole routing.

In this paper we propose a novel routing scheme for virtual cut-through routing that attempts to combine the low routing delay of deterministic routing with the flexibility and low queuing delays of adaptive routing. This hybrid routing mechanism relies on pipelined implementation where different paths and stages of the router are used for different routing modes. The experimental, simulation based results show that the hybrid scheme does achieve, under most conditions, the low latency of the deterministic approach as well as the high saturation point of the adaptive one.

The deterministic and adaptive routing algorithms are described in Section 2 along with the model of the routing delay for virtual cut-through routing. The hybrid routing scheme is described in Section 3 along with simulation results for the three types of routing for k-ary n-cube networks and for various message sizes. Concluding remarks are given in Section 4.

2 Deterministic and Adaptive Routing

The interconnection network model considered in this study is a k-ary n-cube using virtual cut-through switching [13]: message advancement is similar to worm-hole routing [15], except that the body of a message can continue to progress even while the message head is blocked, and the entire message can be buffered at a single node. Note that a header flit can progress to a next node only if the whole message can fit in the destination buffer. For simplicity all messages are assumed to have the same length.

2.1 Routing Models

In the deterministic routing scheme [4, 5], a message is routed along decreasing dimensions with a dimension decrease occurring only when zero hops remain in all higher dimensions. By assigning an order to the network dimensions, no cycle exists in the channel-dependency graph and the algorithm is deadlock-free.

The adaptive routing scheme considered here is described in [7, 8, 2] (also known as the *-channels algorithm). In this algorithm, adaptive routing is obtained by using virtual channels along with dimension-order routing. A message can be routed, adaptively, in any dimension until it is blocked. Once a message is blocked, it is then routed using the dimension-order routing. This algorithm has been proven to be deadlock-free as long as the following routing restrictions are imposed: when the message size is greater than the buffer size (i.e. size of the the virtual channel), deadlock is prevented by allowing the head flit of a message to advance to the next node only if the receiving queue at that node is empty. If the message size is less than the buffer size, then deadlock is prevented by allowing a message to advance only as long as the whole message fits in the receiving queue at that node. This algorithm requires a minimum of three virtual channels per dimension per node for each physical unidirectional channel. Therefore, the number of virtual channels grows linearly with the size of the network.

2.2 Switching Models

In this study, both the deterministic and adaptive routing schemes use one *unidirectional physical channel* (PC) per dimension per node. Figure 1 shows a schematic for each of the routers simulated here for the 2D case. In the deterministic routing case, both high and low virtual channels (VC) of each dimension are multiplexed onto one physical channel. In the adaptive routing case, the deterministic and adaptive VCs are multiplexed onto one PC. For both cases there is only one PC for the sink channel. Once this channel is assigned to a message, it is not released until the whole message has finished its transmission.

The deterministic router uses storage buffers associated with output channels, while the adaptive router uses storage buffers associated with input channels. When using output buffers, the routing decision is made before buffering the message. This type of routing is ideal for deterministic routing because only one choice is available for an incoming message. When a message comes into a node, it can be immediately placed into the appropriate buffer.

When using input buffers, the routing decision is made after buffering the message in the buffer associated with the input channel. This strategy lacks the problem of early commitment of output channels. Since a message can usually be routed on several possible output channels in adaptive routing, this buffering strategy was used for the adaptive router.

The input/output selection policy used for adaptive routing is as follows: a round-robin policy is used for message selection first among all adaptive buffers and then among all deterministic buffers. Output channel selection is performed in each dimension with decreasing number of hops until a free channel is found. By using this output channel selection policy, the greatest

amount of adaptivity for a message is retained which reduces blocking.

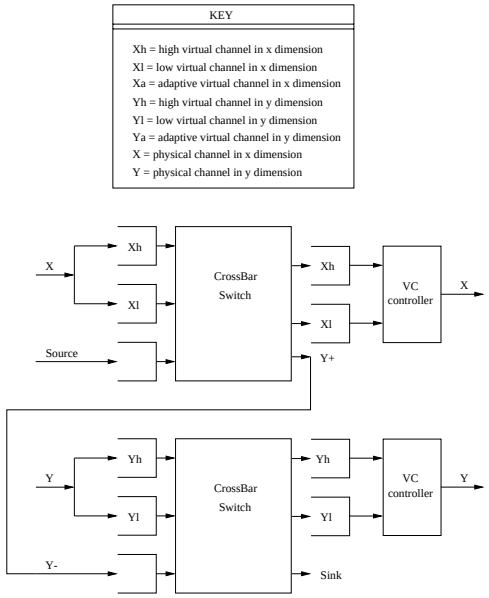
2.3 Modeling Router Delay

In this section we describe a router delay model for the virtual cut-though deterministic and adaptive routers. The model is based on the ones described in [3, 1, 9]. These models account for both the logic complexity of the routers as well as the size of the crossbar as determined by the number of virtual channels that are multiplexed on one physical channel. These models were modified to account for the varying buffer space used in virtual cut-through routing. The parameters of these models are:

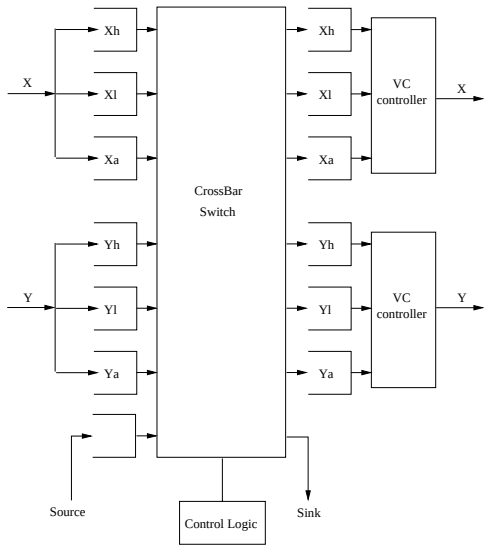
Symbol	Variable (delay)
T_{AD}	Address decoding
T_{ARB}	Routing arbitration
T_{CB}	Crossbar
T_{FC}	Flow control
T_{SEL}	Header selection
T_{VC}	Virtual channel controller
P	Max. no. of IP or OP ports in crossbar
F	Degrees of freedom (OP choices of a message)
C	No. of virtual channels
B	Buffer size (in number of flits)

The address decoding term (T_{AD}) includes the time for examining the packet header and creating new packet headers for all possible routes. The time required for selecting among all possible routes is included in the routing arbitration delay (T_{ARB}). The crossbar delay (T_{CB}) is the time necessary for data to go through the switch's crossbar and is usually implemented with a tree of gates. The flow control delay (T_{FC}) includes the time for flow control between routers so that buffers do not overflow. T_{SEL} is the time for selecting the appropriate header. Finally, the virtual channel controller delay (T_{VC}) includes the time required for multiplexing virtual channels onto physical channels.

For all dimension-order routers simulated here, the number of degrees of freedom (F) equals the number of switch crossbar ports (P). This results because a deterministic router routes a message in either the same dimension on which the message came (on either the low or high channel) or routes it to the next dimension. For all of the adaptive routers, $F = P - 2(n-1)$ where n equals the number of network dimensions. This relationship holds because adaptive routing can use the adaptive channels in all the dimensions while only two



(a) Deterministic Router



(b) Adaptive Router

Figure 1: Schematics of two routers for the 2D case

virtual channels per physical channel can be used in dimension-order (to avoid deadlock). Note that this relationship includes the delivery port.

Delay equations for the routers are derived, using the above parameters. The constants in these equations were obtained in [3] using router designs along with gate-level timing estimates based on a 0.8 micron CMOS gate array process. Three main operations are used in all of the routers simulated here which contribute to the following three delays:

- T_r: Time required to route a message
- T_s: Time necessary to transfer a flit to the corresponding output channel
- T_c: Time required to transfer a flit across a PC

The equations are:

$$T_r = T_{AD} + T_{ARB} + T_{SEL}$$
$$T_r = 2.7 + 0.6 + 0.6 * \log_2 F + 1.4 + 0.6 * \log_2 F$$

$$T_s = T_{FC} + T_{CB} + T_{Latch}$$
$$T_s = 0.8 + 0.6 * \log_2 B + 0.4 + 0.6 * \log_2 P + 0.8$$

$$T_c = 4.9 + T_{VC}$$
$$T_c = 4.9 + 1.24 + 0.6 * \log_2 C$$

Using the above equations, the delay values were calculated for each of the router algorithms simulated and are shown in Table 1. To decrease the overall router delay, it is assumed that all three operations are overlapped through pipelining as described in [9], and therefore the clock period is determined by the longest delay:

$$T_{ccperiod} = Max(T_r, T_s, T_c)$$

From the data in Table 1, we observe that increasing the buffer size, in deterministic routers, increases the overall router delay when moderate to large buffer sizes are used. For small buffer sizes the clock cycle is dominated by the transfer time T_c while for larger ones it is dominated by the switching time T_s. In adaptive routers, the clock cycle time is dominated by T_r. Increasing buffer size increases the overall router delay only when very large buffer sizes are used. Finally, changes in the buffer size affects deterministic routers' clock cycles more than adaptive routers'.

All of these added delays result in adaptive routers that are 13 to 30 % slower than deterministic routers. These results are similar to the results in [1] where 15% to 60% improvement is required for f-flat routers with similar number of virtual channels and under worm-hole routing.

B	T_r	T_s	T_c	CC Period
8	6.60	5.15	6.74	6.74
16	6.60	5.95	6.74	6.74
24	6.60	6.42	6.74	6.74
32	6.60	6.75	6.74	6.75
48	6.60	7.22	6.74	7.22
64	6.60	7.55	6.74	7.55
96	6.60	8.02	6.74	8.02

a- Deterministic router for k-ary 2-cube and 3-cube networks ($C = 2$ and $P = F = 3$ for all)

B	T_r	T_s	T_c	CC Period
8	7.80	6.19	7.09	7.80
16	7.80	6.99	7.09	7.80
24	7.80	7.46	7.09	7.80
32	7.80	7.79	7.09	7.80
48	7.80	8.26	7.09	8.26
64	7.80	8.59	7.09	8.59
96	7.80	9.06	7.09	9.06

b- Adaptive router for k-ary 3-cube networks ($C = 3$ and $P = 10$ and $F = 6$ for all)

Table 1: Deterministic and adaptive router delays (all values in $nsec$)

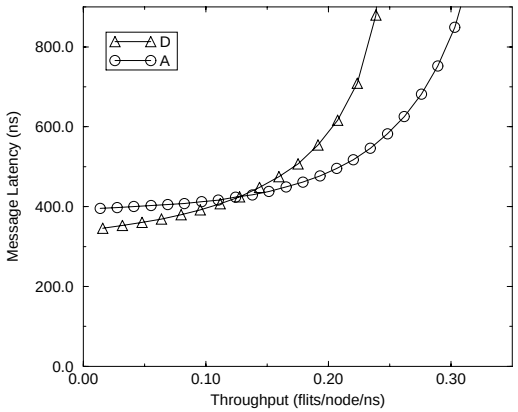


Figure 2: Latency of dimension-order and adaptive routing on a 10-ary 3-cube network under random uniform traffic for buffer area = 48 flits and L = 8 flits.

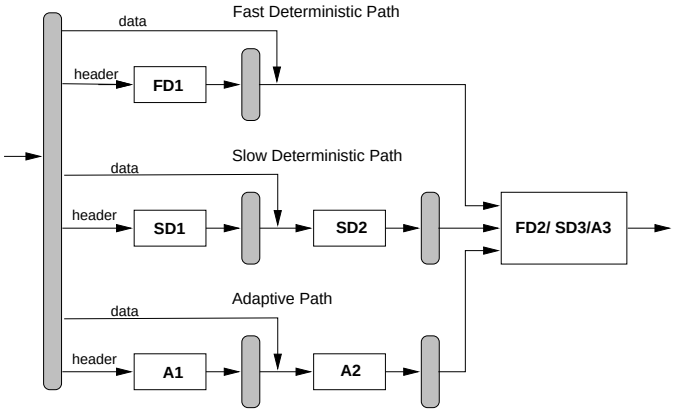


Figure 3: Logic schematic of the hybrid router.

3 Hybrid Routing

A typical comparison of deterministic versus adaptive routing latencies is shown in Figure 2: at low traffic and for short to moderate message sizes, the latency of deterministic routing is smaller. However, the flexibility of adaptive routing provides smaller queuing delays and a much higher saturation point. The objective of the hybrid routing mechanism is to combine the short latency of deterministic routing for low traffic with the shorter queuing delays of adaptive routing at high traffic. In this section we describe the mechanism of the hybrid routing scheme and present the preliminary results of its performance evaluation.

Hybrid Router Model. The hybrid router, shown as a schematic in Figure 3, consists of three logically independent message paths: Fast Deterministic Path (FDP), Slow Deterministic Path (SDP), and Adaptive Path (AP)[1]. The FDP requires two stages for a header flit and one clock cycle for a data flit. The SDP and AP both take three clock cycles for a header flit and two clock cycles for a data flit.

These paths are shown in flow chart format in Figure 4 along with their respective pipeline stages. In this scheme, a header flit entering on a deterministic channel that is also able to leave on a deterministic channel of the same type (low/high) and dimension, goes through the router on the FDP. If a deterministic channel of the same type is not available or a message is being switched to a different type or dimension, then the message is sent through the SDP. A header flit entering on any adaptive channel, is first routed to a deterministic path if possible. Otherwise it is routed to an adaptive channel. In either case, the message goes through the AP.

Since the routing decision and switching logic for routing along the FDP is simpler than traditional deterministic routing, the FDP router requires only two stages. Also, the clock cycle times used for the hybrid router are equal to or larger than those of a purely adaptive router. Therefore more "work" can be accomplished within a clock cycle[2].

Note that this routing scheme is deadlock free: for any given message, the choice of paths selected is always a true subset of those that could be selected by the adaptive algorithm described in [8]. Since the adaptive algorithm has been proven deadlock free, the hybrid is also deadlock free.

Experimental Results. Simulation of the deterministic, adaptive and hybrid routing schemes were performed using a discrete-time simulator. Simulation results were obtained for various 8-ary 3-cube and 10-ary 3-cube networks. The simulation uses a stabilization threshold of a 0.005 difference between traffic 1000 clock cycles apart to determine steady state. Message sizes varied from 8 to 64 flits and traffic from 0.1 until saturation was reached in 0.1 increments. The buffer sizes used in the simulation are all equal to a single message length. The adaptive router and the adaptive path in the hybrid router use three virtual channels per dimension. The deterministic router and the deterministic path in the hybrid router uses

[1] Physical stages are actually shared among these logically independent paths.

[2] As always, it might be necessary to modify this pipeline organization to accommodate a specific physical implementation.

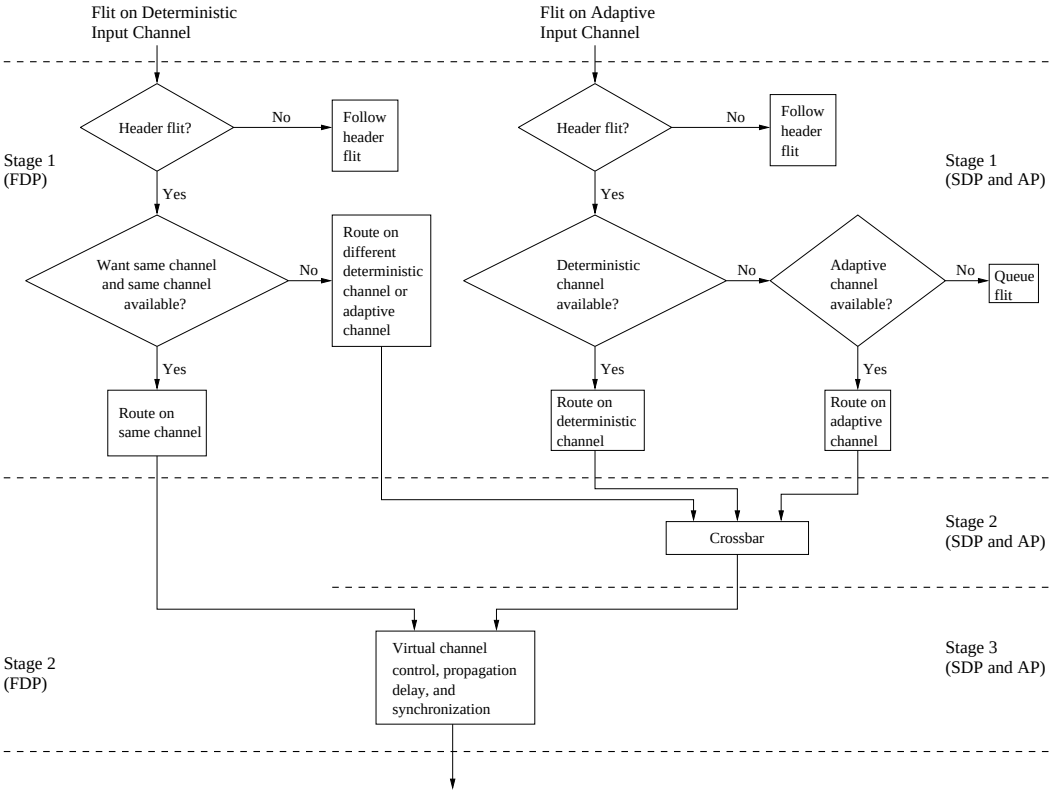


Figure 4: Flow chart of hybrid routing algorithm

two. The simulator implements a back-pressure mechanism which results in a negative slope of the latency versus accepted traffic plots at higher loads.

The hybrid routing scheme is evaluated using two distinct scenarios for a possible clock cycle time. In the first, the clock cycle time of the hybrid router is equal to that of the adaptive router. In the second, the clock cycle time of the hybrid is equal to the adaptive cycle time plus two gate delays to account for the increased critical path length due to a selector. These two options are referred to as H_{min} and H_{max}, respectively.

The H_{min} Scenario. (Figures 5 and 6). For small messages (8 flits) the latency of the hybrid router is not only lower than the adaptive one but is also lower than the deterministic one at low traffic. This is due to the fact that the hybrid router has a 2-stage/1-stage pipeline for header/data flits, while the deterministic router has a 3-stage/2-stage pipeline. Even though each stage in the deterministic router is shorter than the hybrid's router, the greater number of stages a message must go through dominates. For medium messages (16 flits) the latency of the hybrid router is very close to that of the deterministic one at low traffic and follows the adaptive one at higher traffic. For larger messages (64 flits) the hybrid router latency is lower than the adaptive one at low traffic and slightly higher at high traffic. In general, under this scenario the latency of the hybrid router follows the deterministic one at low traffic and the adaptive one at high traffic.

Note that as message size increases, the performance advantage of the hybrid router decreases compared to the other two routers. This is due to the facts that more messages, and therefore headers, are needed to acheive the same utilization with short message length and the hybrid router has a performance advantage for header flits, especially at low utilization. While the deterministic router has a 3-stage header flit pipeline with a low clock cycle time, the hybrid router has a 2-stage deterministic header flit pipeline with a higher clock cycle time. Since the number of pipeline stages dominates performance (and not the clock cycle time), the performance difference between the routers is greater for small message sizes than for large message sizes. This difference also exists at high traffic, although it's much smaller due to the fact that more message blocking occurs covering up differences in header flit time. This difference is exaggerated in larger sized networks because the average number of hops per message increases, thereby increasing the header flit contribution.

Network	L	B	D	A	H_{min}	H_{max}
8by3	8	8	0.139	0.253	0.253	0.219
	16	16	0.169	0.281	0.281	0.244
	64	64	0.175	0.268	0.267	0.234
10by3	8	8	0.142	0.248	0.248	0.215
	16	16	0.170	0.276	0.267	0.231
	64	64	0.173	0.263	0.263	0.230

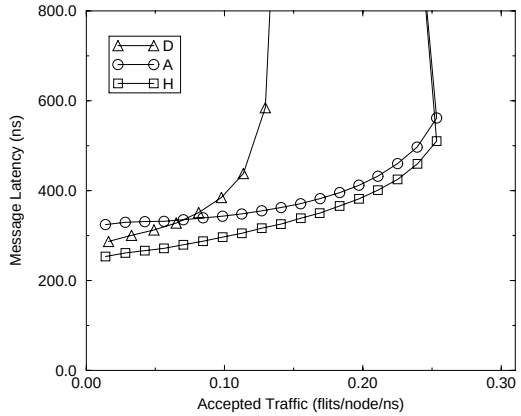
Table 2: Traffic saturation points (flits/ns/node) for deterministic, adaptive, and hybrid routing

The H_{max} Scenario (Figures 7 and 8). In this scenario the hybrid router clock cycle equals the adaptive router clock cycle plus two gate delays. For small and medium size messages (8 and 16 flits), the latency of the hybrid router is better than the adaptive one at low traffic and in between the deterministic and the adaptive one at medium and high traffic. For a message size of 64 flits, the latency of the hybrid router is always worse than the adaptive but is better than the deterministic at high traffic.

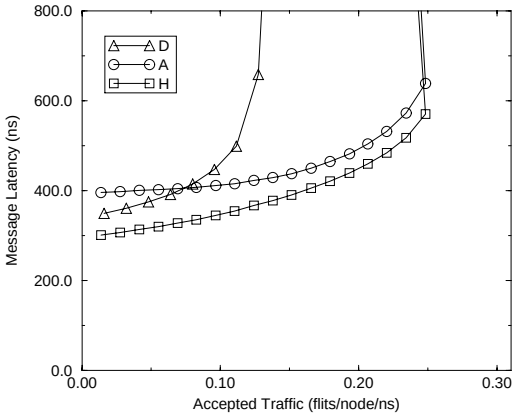
Saturation Point. The saturation point of the hybrid router is, in all cases, much higher than that of the deterministic router. The saturation point of the hybrid router is either equal or lower by at most 3.3% under the H_{min} scenario and by 12.5% to 16.3% under the H_{max} scenario. One reason for the slight decrease in saturation point for the hybrid router, is that the hybrid router routes messages onto the deterministic channels first reducing the number of options available to a message later on. As traffic increases, this less availability cause more blocking and slightly smaller saturation points.

4 Related Work

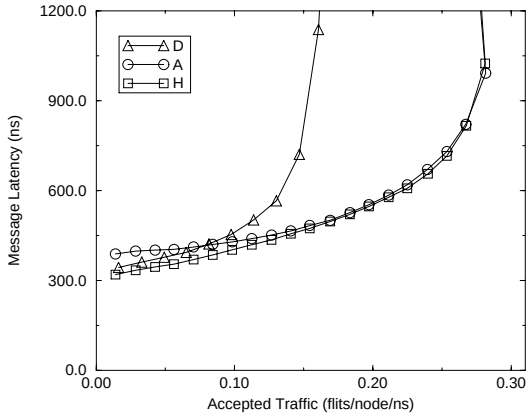
The architectural support for the reduction of communication overhead is described in [6]. This scheme exploits the communication locality in message passing programs to distinguish between cacheable and non-cacheable virtual channels. Cacheable virtual channels are retained for multiple messages thereby allowing an overlap of communication and computation and eliminating the overhead of multiple message set-up. This mechanism is a hybrid scheme combining circuit and worm-hole switching. The implementation



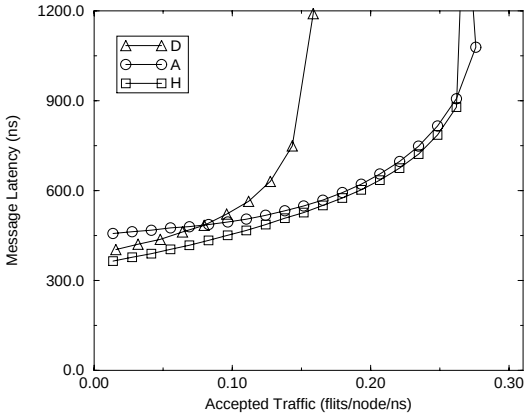
(a) L=8, B=8



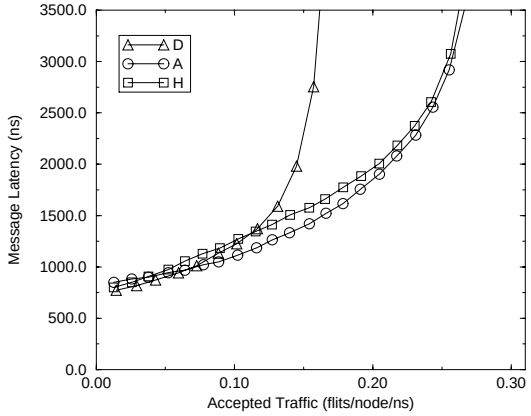
(a) L=8, B=8



(b) L=16, B=16

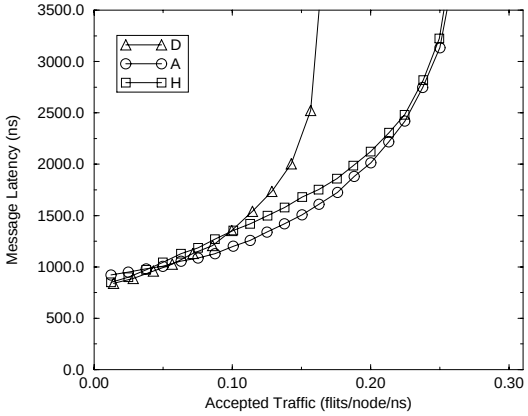


(b) L=16, B=16



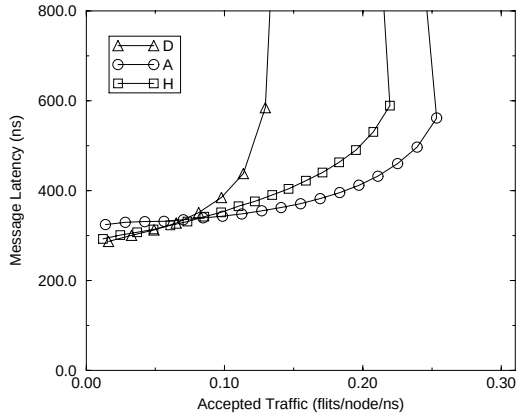
(c) L=64, B=64

Figure 5: 8-ary 3-cube (H_{min} scenario)

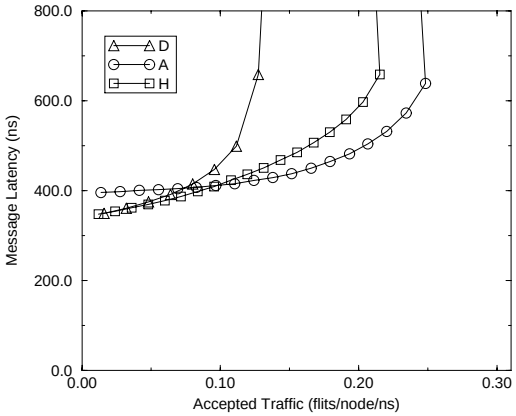


(c) L=64, B=64

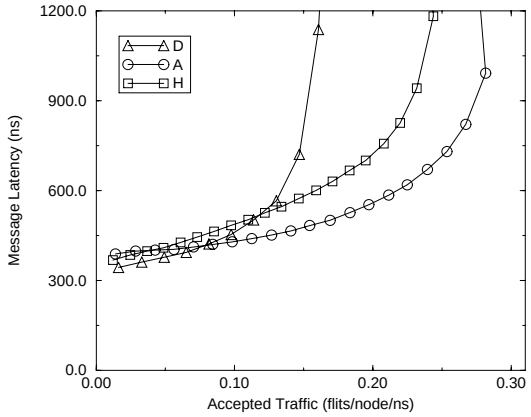
Figure 6: 10-ary 3-cube (H_{min} scenario)



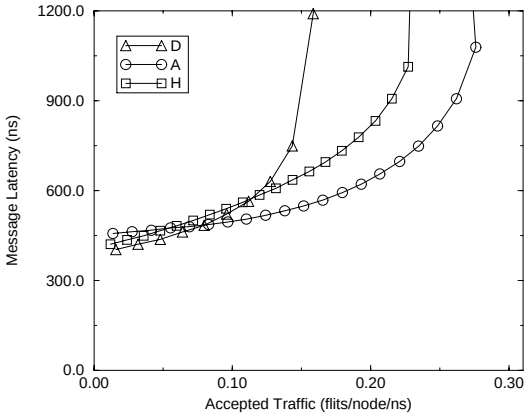
(a) L=8, B=8



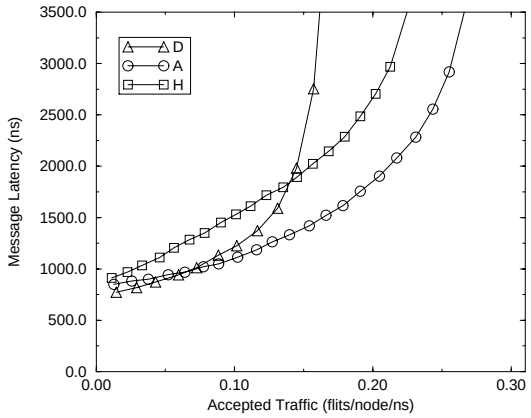
(a) L=8, B=8



(b) L=16, B=16

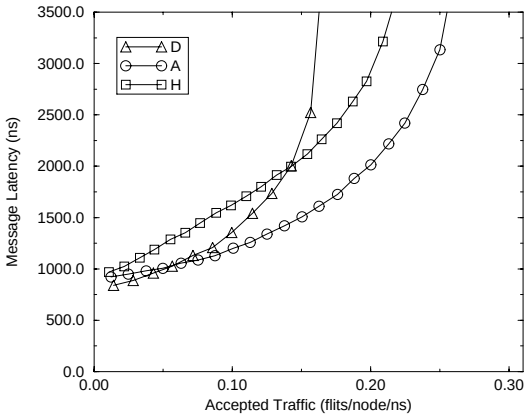


(b) L=16, B=16



(c) L=64, B=64

Figure 7: 8-ary 3-cube (H_{max} scenario)



(c) L=64, B=64

Figure 8: 10-ary 3-cube (H_{max} scenario)

of a router supporting this scheme is described in [10]. Its routing properties are discussed in [11].

Comparisons of adaptive and deterministic router implementations, for worm-hole routing, are described in [1, 3] and [9]. However, the comparison in [1, 3] does not account for the reduced queuing delay in adaptive routing. In [9] the reduction in queuing delay for worm-hole routing is taken into account and the comparison is based on a constant total buffer area.

5 Conclusions

This paper reports on the preliminary evaluation of a hybrid deterministic-adaptive routing scheme. This scheme relies on a pipelined implementation of two routers within each node: a deterministic and an adaptive one. The delay along the deterministic path is one clock cycle shorter than the adaptive one. If the resources are available an arriving message header is routed, by default, on the deterministic path thereby achieving a lower latency per node.

The results from the simulated evaluation of this scheme show that it does achieve its objective: a message latency comparable to that of the deterministic router at low traffic and a saturation point close to that of the adaptive router at high traffic when the hybrid router clock cycle is close to that of the adaptive and for small message sizes when the hybrid router clock cycle is two more gate delays than that of adaptive.

We are currently developing an architecture implementation of the hybrid router in order to evaluate the feasible range of its clock cycle time. We are also evaluating its performance under non-uniform source destination distributions.

References

[1] K. Aoyama and A. Chien. The cost of adaptivity and virtual lanes in wormhole router. *J. of VLSI Design*, 2(4), 1995.

[2] P. Berman, L. Gravano, G. Pifarre, and J. Sanz. Adaptive deadlock and livelock free routing with all minimal paths in torus networks. In *Proc. of the Symp. on Parallel Algorithms and Architectures*, pages 3–12, 1992.

[3] A. Chien. A cost and speed model for k-ary n-cube wormhole routers. In *IEEE Proc. of Hot Interconnects*, Aug. 1993.

[4] W. Dally, A. Chien, and et al. The J-Machine: a fine-grain concurrent computer. In *Proc. of the IFIP Congress*, pages 1147–1153, Aug. 1989.

[5] W. J. Dally. Virtual-channel flow control. *IEEE Trans. on Computers*, 3(2):194–205, March 1992.

[6] B. Dao, S. Yalamanchili, and J. Duato. Architectural support for reducing communication overhead in multiprocessor interconnection networks. In *High Performance Computer Architecture*, pages 343–52, 1997.

[7] J. Duato. Deadlock-free adaptive routing algorithms for multicomputers: Evaluation of a new algorithm. In *Proc. of the 3rd IEEE Symp. on Parallel and Distributed Processing*, Dec. 1991.

[8] J. Duato. A new theory of deadlock-free adaptive routing in wormhole networks. *IEEE Trans. on Parallel and Distributed Systems*, 4(12):1320–1331, December 1993.

[9] J. Duato and P. Lopez. Performance evaluation of adaptive routing algorithms for k-ary n-cubes. In *Parallel Computer Routing and Communication*, pages 45–59, 1994.

[10] J. Duato, P. Lopez, F. Silva, and S. Yalamanchili. A high performance router architecture for interconnection networks. In *Int. Conf. on Parallel Processing*, August 1996.

[11] J. Duato, P. Lopez, and S. Yalamanchili. Deadlock- and livelock-free routing protocols for wave switching. In *Int. Parallel Processing Symp.*, April 1997.

[12] C. L. Glass and L. M. Ni. The turn model for adaptive routing. In *Int. Symp. on Computer Architecture*, pages 278–287, 1992.

[13] P. Kermani and L. Kleinrock. Virtual cut-through: a new computer communication switching technique. *Computer Networks*, 3:267 – 286, 1979.

[14] Annette Lagman. *Modelling, Analysis and Evaluation of Adaptive Routing Strategies*. PhD thesis, Colorado State University, Computer Science Department, November 1994.

[15] L. M. Ni and P. K. McKinley. A survey of wormhole routing techniques in direct networks. *IEEE Computer*, pages 62–76, 1993.