A Survey of Hybrid Router Architectures

James Balfour
jbalfour@stanford.edu

Rebecca Schultz
rschultz@stanford.edu

1 Introduction

Hybrid routing architectures support two or more routing strategies in the same routers. These can be separated into two categories, dynamic hybrid routers and multi-class routers. Dynamic hybrid routers use multiple routing algorithms to give best performance under different network conditions. In general, this means using a low latency deterministic routing strategy when the load on the network is low and a more complex adaptive scheme when load is high. While the decision is made dynamically in response to network load, at any point in time, a given router operates either deterministically or adaptively. Multi-class routers, on the other hand simultaneously support two or more types of traffic in the same routers. Data can be routed as higher priority packets, which are give lower latency routes or in some cases latency and throughput guarantees, or lower priority packets, which lack these guarantees but may achieve higher total throughput, especially at high network load.

The quality of a routing algorithm may be analytically evaluated by its performance, in terms of the latency and offered throughput, for a variety of representative and worst-case traffic patterns. Ultimately, the quality of a routing algorithm must be evaluated by the performance of an interconnection network constructed from routers implementing the algorithm. More complex routing algorithms lead to complicated router implementations which may require additional logic pipeline stages, limit the maximum operating frequency, and increase the logic area. Furthermore, the additional logic required to implement a complex routing algorithm will increase power dissipation and limit the extent to which an architecture can scale within a fixed power budget.

A reasonable evaluation of a hybrid routing scheme should compare the benefits provided the hybrid approach to the overhead introduced by supporting multiple routing algorithms. The performance gains may be evaluated by comparing the latency and throughput of the hybrid scheme to the latency and throughput of algorithms from the classes of routing algorithms being combined. Because the hybrid algorithm may not combine optimal algorithms, simply comparing the performance of the hybrid algorithm to the performance of the deterministic and adaptive algorithms may not provide an accurate measure of the hybrid algorithm's performance. Instead, the performance of the hybrid algorithm should be compared to the best alternative algorithms.

A thorough comparison should evaluate performance over a variety of traffic patterns and injection rates. When a routing scheme has been designed for a specific class of applications, it is reasonable to evaluate the performance using patterns typical for the intended application. The heterogeneous collection of processing elements likely to use a NoC interconnect is likely to induce traffic patterns which differ from the synthetic patterns often used to evaluate networks. A prudent evaluation should consider patterns likely to arise in system using the NoC when evaluating the network's performance. A fair comparison must account for latency introduced by the additional complexity of supporting multiple routing algorithms in one router. Because more complex logic often entails deeper logic paths with longer clock period requirements, the comparison should address additional latency introduced by a longer clock period requirement when comparing latency performance.

The hybrid routing scheme must also be evaluated to ensure that it can neither lead to deadlock nor livelock under the flow control algorithm used during evaluation. It is not sufficient to assume that because the routing algorithms being combined by the hybrid scheme prevent deadlock and livelock that the resulting hybrid scheme will also be deadlock and livelock free. One should consider the criteria used to select between the routing algorithms. Network conditions influencing local decisions may vary over the network, which may result in routers located in different regions of the network employing different algorithms. Consequently, one must ensure that the both routing algorithms can harmoniously interoperate.

This remainder of this paper is organized as follows. Sections 2 and 3 review two papers on dynamic hybrid routers and sections 4 and 5 review papers on multi-class routers. The reviews each consist of a short summary of the problem, a review of the proposed solution, an overview of the authors evaluation methodolgy, a critique of the extent to which the solution solves the proposed problem, and an assesment of the fairness of the methodology. Section 6 contains a general critique of hybrid routers as described in the papers and section 7 contains conclusions on the benefits

and drawbacks of hybrid routing strategies.

2 DyAD - Smart Routing for Networks-on-Chip

Hu and Marculescu's proposed DyAD (**Dy**namic **A**daptive **D**eterministic) routing scheme uses both deterministic and adaptive routing to reduce routing delay at low injection rates while actively managing queuing delay at high injection rates by dynamically employing the routing strategy best suited to the local network congestion [10]. Each router monitors proxies for local congestion to estimate whether a deterministic or adaptive strategy would provide better performance under the current network conditions. When local congestion appears low, a deterministic routing strategy is used to quickly forward packets with minimal routing delay. When network congestion increases, an adaptive strategy is used to flexibly forward packets away from congested links, thereby reducing queuing delay, and ideally distributing the traffic more evenly over the network. Hu and Marculescu's simulations suggest that a router implementing the hybrid DyAD routing scheme should perform better than routers implementing only the deterministic or adaptive routing algorithm.

2.1 Problem

Hu and Marculescu proposed DyAD for NoC architectures in which tiles of processing elements and embedded routers are arranged in a regular grid pattern. Each embedded router in the resulting mesh topology is connect to the local processing element and four neighboring routers using high-speed directed point-to-point links. To facilitate effective collaboration between processing elements, the interconnection network must provide an efficient communication infrastructure. The routers should introduce minimal latency while providing optimal throughput for any traffic injection rate or pattern. Clearly, these are conflicting objectives, and the router architecture must establish some compromise.

2.2 Proposed Solution

Hu and Marculescu describe and evaluate a hybrid router designed for use in 2D mesh topologies. The router provides input queues capable of buffering several flits. The internal switching fabric is implemented using a 5×5 crossbar switch. Worm-hole flow control without support for virtual-channels is used to avoid the additional buffering and added control logic complexity required to provide virtual-channels. The routing algorithm enforces an odd-even turn policy which prevents deadlock by restricting the locations at which certain turns are permitted.

The hybrid router implements a variant of the odd-even adaptive routing algorithm. Packets are routed along minimal paths, with certain turns prohibited at some locations to prevent deadlock. The deterministic routing algorithm is derived from the adaptive routing algorithm, and operates similarly except that the flexibility in the output port selection is eliminated to provide only one path from a source node to a destination node. Routing decisions are evaluated each time a new header flit arrives. A router may be able to productively forward a packet to more than one of its neighbors. When routing adaptively, the router forwards the packet to the neighbor with more input queue capacity available. When routing deterministically, the router forwards the packet in a fixed direction dictated by the destination address carried in the header flit.

The hybrid router monitors the occupancy of its input queues and signals congestion its neighbors when the occupancy exceeds a fixed congestion threshold. The router maintains separate congestion states for each of its queues, and is able to selectively signal congestion to any of its neighbors. A router receiving a congestion signal from a neighbor will begin routing adaptively in an effort to route traffic away from the congested link.

Because the processing elements are not expected to implement end-to-end transmission protocols, the router will not drop packets in the network. Instead, backpressure is used to limit the rate at which traffic is injected into the network. A router will not forward a flit until the downstream router indicates that buffer space is available. When the network becomes congested, backpressure propagates to the processing elements, which must reduce the rate at which they inject traffic into the network. Hu and Marculescu do not describe the mechanism used to exchange buffer occupancy information.

2.3 Evaluation Methodology and Results

Hu and Marculescu evaluated the relative performance of their hybrid routing algorithm by comparing the simulated performance of routers implementing dimension-order routing, deterministic odd-even routing, adaptive odd-even routing, and their DyAD routing. Dimension-order routing was included in the evaluation as a proxy for simple deterministic algorithms which facilitate efficient implementations.

Hu and Marculescu used a cycle-accurate simulator to model the behavior of interconnection networks implemented using the different routers. The parameters used to describe models of the routers to the simulator were obtained from synthesized implementations of the designs, and therefore may be considered reasonably accurate. Various synthetic traffic patterns where simulated on 2D meshes of different sizes. Performance was evaluated using the latency and throughput offered by the network at different injection rates. Significantly, latency was measured as the number of cycles elapsed between the cycle in which a source terminal produces a packet for injection into the net-

work and the cycle in which the last flit exits the network, and therefore includes the time during which the packet is queued at the source terminal. An initial period of $2,000$ cycles was allocated for the system to settle into a steady state. Performance data was collected after 20,000 packets had been sent to allow time for initial transients to subside. Packet sizes were fixed at 5 flits, with the first flit presumably transporting the destination address.

Hu and Marculescu report that the DyAD scheme consistently outperformed the adaptive odd-even routing scheme on all of the synthetic traffic patterns simulated for the different mesh networks. As one would anticipate, dimension-order routing provided lower average latency at low injection rates than DyAD for uniform random traffic patterns. However, they report that the DyAD scheme improved upon the throughput offered by dimension-order routing by as much as 61.7% for other synthetic patterns.

Additionally, Hu and Marculescu simulated traffic patterns intended to model communication between processing elements in a multimedia SoC using traces extracting from an H263 video decoder. The communication captured in the traces differs from the synthetic traffic patterns in packets are produced in bursts rather than at continuous rates. The SoC was modeled as a 4×4 array of processing elements. Nine processing elements were randomly selected to inject packets corresponding to the traces while the remaining seven processing elements were programmed to exchange random uniform traffic amongst themselves. The rates at which the trace packets were injected was gradually increased to model the behavior of the decoder as it increases the playback frame rate. The DyAD router consistently achieved the lowest latency at all injection rates and was able to support higher playback rates.

To evaluate the additional logic overhead required to implement the hybrid routing scheme, Hu and Marculescu developed hardware description language models for the routers evaluated in their simulations. Several versions of each design were generated, with the flit width fixed at 32 bits and the input queue depth varied from 2 to 8 flits. Buffers were implemented using registers because of their small size. The models were synthesized, and the gate areas of the synthesized models were compared. The area of DyAD router is reported to be less than 7% larger than the adaptive router, and Hu and Marculescu concludes that the additional overhead required to implement the hybrid routing scheme is minimal.

2.4 Critique of Solution

As with other hybrid router architectures, the DyAD router architecture is best viewed as an adaptive router augmented with a deterministic routing path intended to reduce the routing delay when the network is lightly loaded. The router design adopts a minimalistic approach which eschews enhancements such as virtual-channels and more advanced flow control mechanism to keep device area and power consumption low. While we were not able to review a detailed architectural description, we conjecture that the deterministic mode uses a shorter pipeline to reduce the routing delay.

The deterministic odd-even routing algorithm implemented in the hybrid router was chosen to interact well with the adaptive odd-even algorithm. With the router dynamically switching between routing algorithms as local network congestion varies, it is clearly important for the two algorithms to interoperate without introducing the possibility of deadlock or livelock. With much of the router delay under heavy load attributable to queuing delay, it seems advisable to implement an adaptive algorithm capable of providing excellent load balancing characteristics under heavy load at the expense of incurring additional routing delay. Essentially, the deterministic routing can provide low routing delay under light load, when routing delay tends to dominate queuing delay, and so the adaptive algorithm should provide excellent load balancing under heavy load, when queuing delay completely dominates routing delay. With routing algorithms which provide better load balancing characteristics than odd-even routing available [14, 2, 15], we question whether an alternate adaptive routing algorithm would have provided a more capable hybrid router. Perhaps the choice of odd-even routing was motivated by an expectation that the resulting hybrid router would be smaller and better adapted to a NoC, where device area and power consumption may be more important design considerations than network capacity utilization.

2.5 Critique of Evaluation

A common use for NoC infrastructures is to connect heterogeneous ensembles of processing elements. The traffic patterns induced by the communication amongst these elements is unlikely to exhibit the regular structure of the synthetic traffic patterns. Consequently, it is import to provide some measure of the worst-case throughput offered by the network. Hu and Marculescu unfortunately fail to provide worst-case traffic patterns and throughput measurements. While the simulation of the extracted traffic traces provides some evidence that the performance for real system traffic patterns may be similar to the performance observed for the synthetic patterns, the lack of a worst-case estimate of the throughput renders the evaluation incomplete.

Reporting latency as measured from the source processing element to the destination processing element provides a more meaningful estimate of the latency than that measured from the router input port, and is clearly the correct the correct metric to report when backpressure flows upstream to the source. This consideration is particularly important when evaluating networks with little internal buffer

capacity, such as those simulated by Hu and Marculescu.

A strength of Hu and Marculescu's evaluation is the use of model parameters extracted from accurate, synthesized hardware descriptions. However, an advantage of deterministic routing is the simplicity of the router architecture in allows. This simplicity manifests itself as fewer pipeline stages separating input ports from output ports, as a higher maximum operating frequency due to shallower logic, or as some combination of the two. Hu and Marculescu do not address the impact of clock frequency on router performance, particularly latency, and do not evaluate how the relative performance of a deterministic router would change if it were able to operate at a higher frequency than the adaptive routers.

More significantly, one of the primary assumptions underlying the argument that a hybrid routing scheme can provide better latency at low injection rates is the assumption that a hybrid router implementation will provide the low latency forwarding delay associated with deterministic routing when routing deterministically. From the limited description of the router implementations, the mechanism by which Hu and Marculescu's hybrid router reduces the forwarding latency when routing deterministically is not apparent. Furthermore, the decision to evaluate the routers at the same operating frequency is not justified. As Hu indicates, supporting multiple routing algorithms and dynamically selecting between them introduces additional complexity into the arbitration and control logic. This additional complexity is likely to reduce the maximum clock frequency at which the router will operate, or to require additional pipeline stages be introduced. Either of these changes will increase the minimum latency across the router as measured in time units. However, only increasing the pipeline depth will increase the latency when measured in cycles. Hu and Marculescu obscure the effects of logic on clock frequency by reporting all latency values in clock cycles and assuming that all routers operate at the same frequency. Clearly, the assumption of identical clock frequencies is only palatable when some external factor, such as the switching frequency of the point-to-point links, limits the router's operating frequency.

The impact of packet length on the performance of the hybrid router was neither addressed analytically nor evaluated through simulation. As the results of Kumar and Najjar demonstrate, the relative performance of hybrid routers often appears better when routing shorter packets because the deterministic routing mechanism often provides the greatest performance advantage when routing header flits [12]. Consequently, one must question whether the performance advantage reported by Hu and Marculescu will diminish when routing longer packets.

3 Combining Adaptive and Deterministic Routing: Evaluation of a Hybrid Router

Kumar and Najjar propose a hybrid minimal routing scheme for virtual cut-through routing on k-ary n-cubes. The hybrid scheme attempts to combine the low routing delay of deterministic routing with the flexibility and low queueing delays of adaptive routing by including multiple paths of differing complexity through the router. Factors including the depth of pipelining, clock cycle time, and packet length are carefully considered when evaluating the hybrid router's performance. Kumar and Najjar conclude that the hybrid routing scheme can perform better than deterministic or adaptive routing on most traffic patterns over most injection rates.

3.1 Problem

Kumar and Najjar limit their consideration of hybrid routing schemes to virtual cut-through routing algorithms for k-ary n-cubes. Packet advancement differs from wormhole routing in that the body flits are allowed to advance even when the header flit is blocked. A single node may buffer an entire packet, and the header flit is only permitted to advance when the next node has enough buffer space to receive the entire packet. No specific characteristics are assumed of the elements attached to the interconnection network. The investigation of hybrid routing scheme is motivated by the observation that dimension-order routing outperforms adaptive minimal routing on k-ary n-cubes when the accepted traffic rate is low.

Kumar and Najjar suggest that the packet delay incurred through each router can be reasoned about as the sum of two components: routing delay, and queuing delay. Routing delay includes the time required for the router to determine how to route a packet, and is influenced by the complexity of the routing algorithm. Intuitively, routing delay is determined by the depth of pipelining in the router and the clock cycle time. Queuing delay includes the time during which packets wait in buffers for resources within the router to become available, which is influenced by the flexibility in assigning resources afforded by routing strategy. Simple deterministic strategies reduce routing delay at the expense of greater queuing delay. Conversely, more flexible adaptive strategies reduce queuing delay at the expense of complex router architectures that introduce additional routing delay.

Accordingly, Kumar and Najjar argue that the primary performance advantage of adaptive routing results from the ability to reduce queuing delay by providing multiple options for advancing packets through the network [12]. Furthermore, the flexibility provided by adaptive strategies allows them to saturate at much higher injection rates by adaptively balancing traffic load over the available capacity. However, the adaptive algorithms considered in their com-

parisons require more virtual-channels and introduce a more complex output channel selection algorithm than a deterministic routing algorithm designed for the same topology. The additional complexity results in routers with deeper pipelines and longer clock cycles. Kumar and Najjar propose a router architecture which provides multiple paths of differing complexity through the router as a means of dynamically providing either the low routing delay of deterministic routing or the low queuing delay of adaptive routing with negligible impact on the clock cycle time [12].

3.2 Proposed Solution

The study considers deterministic and adaptive routing schemes for k-ary n-cube topologies in which nodes are connected by unidirectional links. The deterministic scheme provides two virtual-channels in each dimension, while the adaptive scheme provides three. Virtual-channels are not provided for the link connecting the router and local processing element. The deterministic routing scheme manages buffers associated with input channels, while the adaptive routing scheme manages buffers associated with output channels.

The hybrid architectures proposed by Kumar and Najjar provide three parallel paths through the router. These paths are referred to as the Fast Deterministic Path (FDP), the Slow Deterministic Path (SDP), and the Adaptive Path (AP). The FDP is used to quickly route packets entering on a deterministic virtual-channel which can leave on the same virtual-channel continuing along the same dimension. The SDP is used to deterministically route packets which cannot use the FDP, either because the required virtual-channel is not available, or because the packet must be forwarded along a different dimension. The AP is used to adaptively route packets when neither the FDP nor the SDP are available. The hybrid routing scheme avoids deadlock by ensuring that the path selected by the router for any packet is always a subset the paths which could be selected by the deadlock-free adaptive algorithm presented in [6].

Kumar and Najjar present two related hybrid architectures which differ in pipeline depth and the clock cycle time. These architectures are referred to as the Pipelined Hybrid Router (PHR) and Super-Pipelined Hybrid Router (S-PHR). The architectures are similar, with differences arising from the introduction of finer pipeline stages to the S-PHR to allow the clock cycle time to be reduced. The earlier pipeline stages implement logic specific to a routing algorithm. The paths converge in the later stages, allowing resources such as the crossbar in the later stages to be shared across all paths. Body flits, which simply follow a header flit through the router, bypass the initial stages implementing the channel selection algorithms, and therefore experience less delay across the router. We have reproduced as Figure 1 and Figure 2 the logic schematic of the pipelined and super-pipelined hybrid routers presented by Kumar and Najjar in [12].

Table 1. Clock cycle times for k-ary 3-cube routers

Buffer	Deterministic		Adaptive		Hybrid	
Depth	PR	S-PR	PR	S-PR	/ PR	S-PR
8 flits	6.74 ns	4.90 ns	7.80 ns	4.90 ns	8.40 ns	5.00 ns
16 flits	6.74 ns	4.90 ns	7.80 ns	4.90 ns	8.40 ns	5.00 ns
32 flits	6.74 ns	4.90 ns	7.80 ns	4.90 ns	8.40 ns	5.00 ns

3.3 Evaluation Methodology and Results

The routing delay for the deterministic, adaptive, and hybrid routers were estimated using an analytic model for virtual cut-through routers based on those described in [3, 1, 8]. The model attempts to account for the factors such as the differing degrees of logic complexity and sizes of crossbar switches required by the different router architectures. The model is very similar to that presented by [3], from which many of the values for gate-level delay estimates were borrowed. Kumar and Najjar extended the model to account for the additional delay introduced by the larger buffer sizes used in their virtual cut-through switching architectures. A fixed wire delay (the time required for a flit to traverse the physical channel connecting two routers) was assumed for all implementations, and the routers were designed such that their clock cycle times exceeded the wire delay. Table 1 replicates the clock cycle times used in the simulations.

Kumar and Najjar present results for a discrete-time simulation of an 8-ary 3-cube. Latency measurements are reported in time units to account for the different clock cycle times estimated for the routers. The simulated traffic was assumed to have reached a steady-state when the difference in traffic measured at intervals 1000 cycles apart was sufficiently low. Simulated packet sizes varied from 8 to 32 flits. The buffers were sized to match the packet length. The following traffic patterns were simulated: random uniform, compliment, perfect shuffle, bit-reversal, and butterfly. Kumar and Najjar draw a

number of conclusions based on the performance of the hybrid router on the simulated traffic patterns. Generally, the latency and throughput performance of the hybrid router is similar to the deterministic router at low injection rates, and is similar to the adaptive router at high injection rates.

The relative performance advantage of the hybrid router declines as the message length increases. The hybrid router performs better than the adaptive when it can lower the routing delay by processing header flits more efficiently. The improvement is particularly noticeable at low injection rates when the low latency deterministic paths are more likely to be used. For small message sizes, the hybrid router was reported to perform better than the deterministic router on uniform random traffic. Furthermore, the observed perfor-

Figure 1. Kumar and Najjar's Pipelined Hybrid Router Architecture

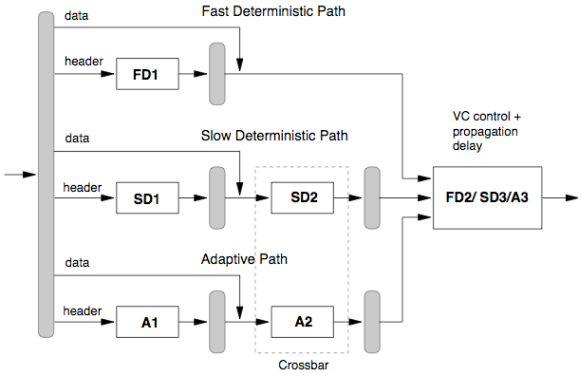
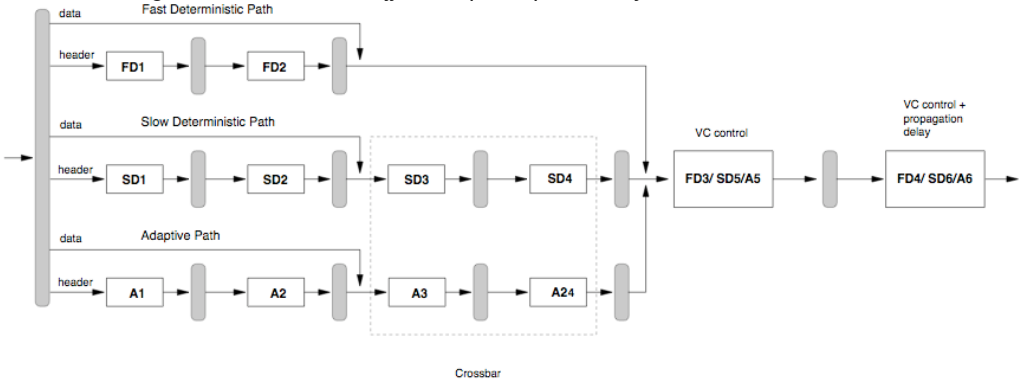


Figure 2. Kumar and Najjar's Super-Pipelined Hybrid Router Architecture



mance advantage decreases at high injection rates because increased congestion and blocking in the router prevents the deterministic paths from being used as frequently.

The saturation injection rate observed for the hybrid router was below that of the adaptive router. Kumar and Najjar conjecture that this results because the hybrid router routes packets onto the deterministic channels first, thereby reducing the number of options later available for routing packets.

Super-pipelining improved the performance of all the routers. The super-pipelined routers demonstrated better overall performance, both in terms of latency and saturation throughput, on all of the traffic patterns that were simulated. The obvious explanation for the improved performance provided by Kumar and Najjar was the increased throughput resulting from the reduced the clock cycle time.

Kumar and Najjar also investigated the impact of the order of path selection on the performance of the hybrid router under congestion. Additional simulations were performed in which the router was modified to use the adaptive path before the slow deterministic path to determine whether the increased flexibility offered by adaptive routing would improve throughput at high injection rates. The modifications did not substantially affect the router's performance. The authors conclude that the hybrid router performs well because the fast deterministic path is used when possible.

3.4 Critique of Solution

Kumar and Najjar's primary contribution is to demonstrate that a hybrid router architecture which supplements an adaptive routing strategy with a fast deterministic path can reduce routing delay when the network is lightly loaded without adversely affecting queuing delay when the network is heavily loaded. Although some degradation in the offered throughput was observed when the network was congested, the hybrid router performed similarly to the deterministic router when the network was lightly loaded and performed similarly to the adaptive router when the network was heavily loaded.

The deterministic algorithm was chosen to allow a simple, low latency deterministic datapath to be implemented in the hybrid router. Similarly, the adaptive algorithm was chosen to integrate well with the deterministic algorithm while providing a reasonable proxy for adaptive routing algorithms. Although capacity utilization under worst-case traffic patterns was not provided, it is known that the minimal algorithms will not provide optimal worst-case throughput on a k-ary n-cube network [5]. Adaptive algorithms which provide better worst-case throughput have been demonstrated [14, 2, 15]. For a hybrid router to compare well against adaptive routers, it must use an adaptive routing algorithm which provides near optimal throughput when the injection rate is high. A hybrid router architecture is particularly amenable to complex adaptive routing algorithms because the deterministic algorithm can provide a low delay path which bypasses the complexity of the adaptive algorithm under light loads. Accordingly, we suggest that, when latency and throughput are the dominant performance metrics under consideration, a better adaptive algorithm should be chosen to improve the throughput offered under heavy load.

Kumar and Najjar's results illustrate the impact of clock cycle time and pipelining on routing delay. One must avoid introducing additional pipeline stages into the deterministic routing path as the different datapaths coalesce at the shared output ports. Similarly, one must consider how the more complex logic required to implement the adaptive routing algorithm will affect the frequency at which the router operates, and attempt to prevent the additional logic from adversely increasing the clock cycle time and penalizing the deterministic path. Kumar and Najjar provide a reasonable example of how one may structure the router to isolate the different paths and preserve the low routing delay expected from deterministic routing.

However, increasing the degree of pipelining and the frequency at which a router operates will significantly increase the amount of power consumed by a router. Similarly, the additional logic required to implement multiple paths through a router will increase the power consumption in proportion to the additional logic. Kumar and Najjar do not consider power consumption when evaluating architectures and do not provide a detailed assessment of the amount of logic required to implement the additional paths through the hybrid router. While their architecture appears to be able to efficiently share resources in later pipeline stages, there is obviously some overhead required to support multiple datapaths.

Unlike Hu and Marculescu's DyAD architecture, Kumar and Najjar's do not use information gathered from neighboring routers to determine which routing strategy should be used. Instead, decisions are made based on local resource availability and preferred strategy. However, once a packet is diverted from the the deterministic path to the adaptive path, it must continue along the adaptive path until it reaches its destination. Effectively, the routing strategy decision at one router continue to be enforced as the packet proceeds through the network. Consequently, packets originating in congested regions of the network will not be able to benefit from the fast deterministic path when they travel into lightly congested regions. The synthetic patterns, which tend to equally distribute packet injection across the network, would not reveal the such effects.

3.5 Critique of Evaluation

The description of the simulation methodology does not explicitly specify whether the latency measures reported in

[12] were measured from the time a packet is created at the source or from the time the packet entered the network. The later latency clearly underestimates the total source to destination latency by failing to include the queuing delay in the source terminal element.

Kumar and Najjar do not provide worst-case throughput, which renders their evaluation incomplete. Because the reported throughput measurements are not expressed as ratios of the network capacity, it is difficult to evaluate how efficiently the hybrid router utilizes the capacity under different traffic patterns.

It is disconcerting that the hybrid router provides better average latency than the deterministic router on uniform random traffic when the network is lightly loaded. The deterministic and hybrid routers should use the same deterministic routing strategy at low traffic rates, and should therefore perform similarly. The discrepancy in the measured performance suggests that the deterministic router could be implemented more efficiently simply by adopting the hybrid routers deterministic path.

When estimating the minimum clock cycle time at which the various routers would operate, the authors prohibited the clock cycle time from falling below the channel transmission delay time. Consequently, the clock cycle times used for the deterministic routers reflect the limitations of the channel delay rather than the router architecture. While the assumption that the clock cycle time will be kept high enough to accommodate channel transmission delays is reasonable when modeling a system, the assumed transmission delay operates to unfairly bias the comparison against the deterministic routers. The improved performance resulting from increasing the degree of pipelining and reducing the clock period clearly illustrates the impact of the clock cycle time on the routers' performance, and the artificial restriction on the deterministic router's clock cycle time unnecessarily confuses the evaluation.

4 Trade Offs in the Design of a Router with Both Guaranteed and Best-Effort Services for Networks on Chip

Rijpkema et. al propose a hybrid router architecture for networks on chip that combines what they refer to as guaranteed and best effort services [13]. Their approach conceptual consists of two routers, a guaranteed services router that uses circuit switching to provide lossless in-order packets delivery with bounded latency and throughput, and a best-effort router that uses packet switching to provide network services without these guarantees. Because network on chip systems are tightly resources constrained, the two router systems are designed to share resources. By providing guaranteed quality of service only to the portion of the traffic that requires it, the network can be provisioned to handle a smaller total load. Also, by using the same network for guaranteed and best-effort service the network resources that have been provisioned for worst case guaranteed traffic, but are idle during average load can be used to deliver the remainder of the traffic, providing better total link utilization.

4.1 Problem

In network on chip systems it is often essential to provide guaranteed quality of service for some applications that require realtime data delivery. Examples are video processing applications, which require in-order data streams at some fixed throughput, and cache updates, which require lossless communication with low-latency. However, providing guaranteed services requires that network be provisioned for the worst case load, even if the average is much lower or if some of the network traffic does not require these guarantees. The situation is further aggravated by the fact that while NoCs in many cases require these guarantees, they are also heavily resource constrained due to area issues.

4.2 Proposed Solution

The paper proposes a router architecture designe conceptually as two separate blocks that share some resources. These are the guaranteed router (GT) and the best-effort router (BE).

The guaranteed router (GT) uses a form of circuit switching to ensure data can be delivered in-order and losslessly at a fixed latency and throughput. Circuit-switching inherently provides these guarantees because contention is resolved when the circuit is established. Each router has a *slot table* consisting of a number of entries mapping blocks of data to input and output ports during a given time slot. The slot tables in separate routers are synchronous, that is the time slot i on each router represents a reservation for an input or output during the same period in time. The separate routers can operate in this lockstep fashion because the NoC has short communication links and is tightly coupled. To begin sending guaranteed traffic a circuit is set up to send data. To establish a circuit a slot is reserved in each of the routers the data will traverse from source to destination, allocating a block in time slot i in the first router along the path $i+1$ in the second and so on until it reaches its destination. Once a route has been reserved it has guaranteed access to the input and output ports it requires in the routers along the path and can not be lost or delayed due to contention.

The best-effort router (BE) provides lossless, in-order delivery without the latency and throughput guarantees. The best-effort router does contention resolution at the granularity of packets rather than circuits. As a result it is dynamically scheduled and requires buffering in the routers. Contention for both the links themselves and the buffers delays

delivery of the packets in a non-predictable manner preventing guarantees. However, the elimination of the set up phase required for circuit switching means the best-effort routers may be able to provide better total throughput under some network loads. Rather than describing a specific design the paper provides a strategy for buffering data and scheduling traffic and indicates that a final design point depends on the requirements of a specific network as far as trade offs between hardware complexity and link utilization.

A number of buffering strategies are described in the paper. One strategy is *output queuing* where a router that has N inputs and N outputs will have N^2 queues at the outputs of the router. The inputs and outputs are then connected using a fully connected interconnect. This provides best performance, but wiring the interconnect is very expensive even for small values of N. A second strategy is *input queuing* where queues are located at the inputs of the router, one for each input. However, do to *head-of-line blocking*, in which contention for outputs from the packets at the heads of the queues prevents packets deeper in the queues from using other idle outputs, for large N utilization saturates at 59%. The best solution proposed in the paper is *virtual output queuing*, a scheme that combines the simplicity of input queuing with the performance benefits of output queuing. In this case each of the N inputs has N queues, one for each output. The queues can be multiplexed into a crossbar switch that connects at most one queue from each input to its corresponding output. These input queues could be mapped onto a RAM, however, because of the area constraints of NoCs the paper uses custom fifos to implement the buffers.

In addition to a buffering strategy, the paper also provides a scheduling approach for the BE routers. This strategy is called matrix resolution and is modeled as a bipartite graph problem in which every input port is modeled by a node in one half of the graph and every output port by a node in the other half. Edges represent a non-empty queue between those inputs and outputs. The problem is solved using a simple iterative algorithm.

The BE and GT routers are combined to share the network links and the switch fabric in the router. Incoming traffic is routed to either the BE or GT portion of the router. The BE portion does scheduling at the flit level, and the GT at the packet level. In order for the BE router to respond to changes in GT traffic the block size of GT messages is a multiple of the BE flit size. GT traffic is always given priority over BE traffic. Additionally, GT circuits are set up using BE packets. There are three control packets *SetUp*,*TearDown* and *AckSetUp*. *SetUp* packets travel from source to destination reserving entries in the slot table at each router. If router does not have a block available in the necessary time slot a *TearDown* packet is sent back down the same path. When the *SetUp* successfully reaches the destination a *AckSetUp* can be sent back to the source and the circuit is established. *TearDown* packets are also used to close the circuit after communication completes. Circuit establishment is pipelined and concurrent, that is multiple control packets for different circuits can be active in the network at the same time. Slot allocation for circuit setup can be done at compile or run-time, however because run-time slot allocation is distributed, resolving conflicts uses a heuristic and may be suboptimal.

4.3 Evaluation Methodology and Results

The authors synthesized a combined GT-BE router that uses wormhole routing queues 8 flits deep each of which is 3 words of 32 bits each. Slot tables had 256 entries, and made up a significant portion of the total area. The total bandwidth in each router was 5×500 MHz $\times$ 32 bits $=$ 80 Gbits/s and the total area was .26 mm^2. The inclusion of custom fifos, instead of RAM based buffers significantly reduced the router size.

4.4 Critique of Solution

The paper addresses the problem of handling mixed traffic in NoC systems. In particular, the authors attempt to create a routing solution that can provide lossless in-order delivery with guaranteed throughput and latency for traffic that requires it and use the resources idle during the average case to deliver best-effort services to traffic that does not require these guarantees. For their scheme to be successful, there must be a mix of the two types of traffic, and a low enough average case load on the guaranteed routers to provide sufficient idle resources to handle the best-effort traffic in a timely manner. However, the authors do not provide any analysis of real workloads that fit these characteristics, nor do they quantify the ratios of types of traffic and the total loads for which the proposed routers perform well.

The authors provide a proposed system to handle mixed traffic but they do not compare their hybrid GT-BE router to any other hybrid or traditional design, nor do they demonstrate that a mixed solution is necessary. The paper does not demonstrate that it is not possible to build a guaranteed network for all NoC traffic at a reasonable cost, nor does it address the possibility of using two separate networks to handle the separate types of traffic.

The time slot based circuit switching algorithm requires that routers be synchronized to these time slots, essentially forcing the each hop in the network to have the same delay as the slowest hop. The authors justify this decision by stating that NoC systems have short communication links and thus can be tightly synchronized. While NoC systems do have much shorter links than off-chip networks, they are consequently expected to operate at much higher frequencies. With communication overheads due to wire delays dominating clock frequencies in many modern processors

it seems undesirable to artificially slow some of the links to maintain synchronicity.

4.5 Critique of Evaluation

The authors provide a synthesized design to justify that the hybrid router architecture they have proposed is feasible in NoC systems, both in area and speed. However, they do not show any performance numbers for the router on an actual workload. While they do give the total bandwidth of the router they do not demonstrate its throughput as a function of capacity under an actual workload of mixed guaranteed and best-effort traffic.

5 Triplex: A Multi-class Routing Algorithm

Triplex is a router proposed by Fulgham and Snyder [9] to simultaneously handle the classes of routing, minimal deterministic, minimal adaptive and non-minimal fully adaptive, for k-ary, n-cubes. The authors contend that in many cases it is desirable to provide deterministic routing to guarantee in-order delivery, but such routers perform poorly in congested networks and, lacking path diversity, fail to provide any fault tolerance. As a compromise, they propose a router that can effectively handle both deterministic and adaptive traffic while remaining deadlock free. Triplex is also of interest because it proposes the first fully-adaptive deadlock avoidance wormhole algorithm for tori.

5.1 Problem

Fulgham and Snyder contend that no single routing algorithm can effectively handle the complex and variable needs of most interconnect networks. While deterministic routing offers low latency and the added benefit of in-order delivery, a requirement for some applications, it can not handle high network load. Minimal adaptive routers can achieve a higher total throughput, but sacrifice in-order delivery and still do not perform well under severe congestion. Fully-adaptive solutions can achieve high throughput even under congestion, but can be very complex to design and the lack of in-order delivery may require additional synchronization be done at a higher level, further congesting and complicating the network.

5.2 Proposed Solution

The Triplex design is a multi-class router than simultaneously supports all three routing algorithm. Both wormhole and packet-switched flow control are supported. Packet-switched flow control can use virtual cut-through or store and forward techniques. The routers can be configured to handle a single routing algorithm and flow control technique at boot time, dynamically or on a per message basis. As a result Triplex can be considered to be both a multi-class, simultaneously supporting multiple types of traffic, and a dynamic, switching between routing algorithms, router.

Each router in the network has a set of buffers including two special buffers, the injection buffer and the delivery buffer. Messages in the injection buffer are sent on the network withing a finite amount of time, and messages in the delivery buffer are removed from the network within a finite amount of time. Effectively these buffers represent reservations for the input and output ports of the router. Routing decisions are made locally within the routers using a routing function to select the buffers which should be used by each message. The basic routing algorithm is non-minimal, but as no message is forced to select a non-minimal route, a message may chose to be routed deterministically, adaptively non-minimal, or fully adaptively. The basis for the routing algorithm is the dimension order Dally-Seitz oblivious, wormhole routing algorithm (DO)[4]. There are two versions of the algorithm, one for wormhole routing and a less complex version for packet-switching. The algorithm can be applied with some slight variation to mesh or torus networks. Each channel in the network is represented by a number of virtual channels, two for mesh networks and three for tori. The routing algorithm separates the virtual channels into two classes, *restricted channels*, the channels in the minimal deterministic route indicated by the traditional DO algorithm, and *unrestricted*, the virtual channels not used by the dimension order routing in the DO algorithm. Buffers are considered *restricted* or *unrestricted* depending on whether the corresponding virtual channels are restricted. For wormhole routing buffers are considered *empty* if both the input and output buffers of the virtual channel are empty. For packet-switching, buffers are *empty* simply if they are non-full, as a single empty buffer will guarantee acceptance of the entire packet by the router. A route which makes use of unrestricted channels, traveling along a path not specified by the dimension order algorithm is called a *deroute*.

The algorithm routes data according to a predefined set of rules. The rules for packet-switched mesh networks are slightly simpler than for tori and are as follows. Messages can be routed using DO rules into restricted buffers at any time. Messages can also chose to deroute into any empty unrestricted buffer. Finally, a message that needs to move in the negative direction of the lowest dimension still uncorrected, l, can use any buffer in a dimension i where $i > l$. The final rule allows messages to use restricted buffers while violating dimension order routing, allowing additional route flexibility. The paper includes a proof of deadlock freedom for the algorithm that is based on the fact that messages follow special, more restrictive routing rules for the lowest uncorrected dimension. To support wormhole networks, provisions must be made to prevent arbitrary length messages from having cyclical dependencies waiting

Table 2. Summary of difference in routing algorithms

Router	Adaptivity	Node latency	Buffers
Oblivious	3	none	18
Obliv Triplex	4	none	18
Duato	4	min adaptive	26
Min Triplex	4	min adaptive	26
Chaos	4	non-min adaptive	15
Triplex	4	non-min adaptive	26

for buffers. To do this, message are allowed to deroute into empty unrestricted buffers only on dimensions higher than the lowest, uncorrected dimension requiring negative correction, $i > l$.

To support torus networks some further restrictions are required to prevent deadlock from occurring around the wrap links.

5.3 Evaluation Methodology and Results

The Triplex router is compared in simulation to the Dally-Sietz oblivious router [4], the Duato router[7] and the Chaos router [11]. These are a dimension order router, a minimal adaptive router and a fully adaptive router that prefers minimal paths. The differences between the routers are summarized in Table 2.

A number of traffic patterns were simulated including random, bit reversal, transpose, and hot spot traffic. Traffic patterns also included a mix of both short, 40 word, and long, 400 word, messages at a ratio of 10 to one.

At low loads with no congestion present the oblivious router showed the best performance, because it has a lower node latency. As load increased the throughput of the oblivious Triplex router did not match the oblivious router by a small fraction due to the extra cycle of node latency. On half of the applied patters, the minimal Triplex router matched or exceeded the Duato throughput performance. In the other patters, however, Duato outperformed the Triplex router despite a more restrictive adaptive algorithm. The authors hypothesize that this is because of the increased opportunities for conflicts present in the more flexible algorithm. Non-minimal Triplex failed to perform as well as minimal Triplex.

5.4 Critique of Solution

The performance of the Triplex router was comparable to other oblivious and minimal adaptive routers but it failed to perform well as a fully adaptive router.

While the authors state that the flexibility of a multi-class router is a qualitative benefit they fail to show a motivating example for providing such flexibility. Also, although they site the flexibility of dynamically changing the routing strategy or supporting multiple types of traffic as a strength of the Triplex system they did not evaluate the router in a mixed mode. The authors tested simulating the three separate routing algorithms, but they did not provide any results for the Triplex router as a dynamic hybrid router switching between strategies or as a multi-class router supporting multiple strategies simultaneously.

5.5 Critique of Evaluation

The main problem with the evaluation was the authors failed to quantify the additionally complexity of supporting multiple algorithms either in design complexity, area or impact on clock cycle time. Simulation results were reported for a Triplex router with a node latency of 4 cycles, identical to the other routers simulated, excepting the dimension order router, however no indication was given of the relative cycle times of the routers. It seems logical to conclude that a more complex design such as Triplex would either have a deeper pipeline, a longer clock cycle time or both. Additionally, it would probably be much larger in area than a comparable less complex design.

6 General Critique

The hybrid routing papers consistently suffered from a number of problems, including failure to provide clear evaluation metrics, failure to justify the mix of routing strategies selected, and most importantly failure to motivate the problem or to evaluate the solution on real, representative traffic patterns.

6.1 Evaluation Metrics

In many cases they failed to provide clear evaluation metrics. The hybrid routing papers studied all are effectively attempting to better utilize available resources by offering a more complex solution combining one or more existing routing algorithms. However, the papers failed to express resource utilization in a way that makes evaluating their effectiveness possible. In general, if the papers aim at providing better total throughput, reporting results as a fraction of network capacity would have made it possible to examine the cost-benefit trade-offs for introducing the additional complexity.

6.2 Selection of Routing Strategies

It is clear that the hybrid routers must select a set of routing strategies that can interoperate without risking deadlock or livelock and that proving this is a non-trivial problem, however in many cases the specific strategies seemed to be selected more for ease of interoperability than performance. Hybrid router architectures are particularly amenable to complex adaptive routing algorithms, because the inclusion of a deterministic algorithm to provide a low delay

that bypasses slower adaptive path. As such, if latency and throughput are to be the dominant performance metrics under consideration, the best adaptive algorithms should be used for maximizing performance under heavy load.

6.3 Traffic Patterns

The largest problem with the papers was that they failed to provide real, representative traffic patterns for motivation or evaluation. Hybrid routing is focused on obtaining good performance in a network that either has a widely varying load, for the case of dynamic routers, or supports a variety of types of traffic, for multi-class routers. Without real traffic patterns that demonstrate these behaviors, it is difficult to justify the inclusion of the additional complexity. There is little gain from providing low latency routing at low injection rates if injection rates are continuously high. Similarly, if most of the traffic requires high priority quality of service, why include a lower priority routing strategy. For the case, of multi-class networks, there was no evaluation as to when it might be more appropriate to support two separate networks for the two types of traffic, particularly in the case of NoCs where wires and bandwidth are less expensive than buffers. Also, one can not evaluate the performance of a router intended to handle a mixed traffic pattern without testing it on one. Specifically, the papers did not fully evaluate router performance when traffic patterns varied widely through the network. In the case of the dynamic routers, they failed to explore how quickly routers would switch from adaptive to deterministic and what effects oscillating between these conditions would have on total performance. For the multi-class schemes they did not give a satisfactory explanation of how priority decisions would be made, and what effect the mix of priorities had on performance at various loads.

7 Conclusions

Providing a deterministic routing strategy for use at low injection rates may be appropriate when the overhead of doing so is acceptable and there is a reasonable expectation of traffic loads that will benefit from the deterministic scheme. A low latency deterministic path through the router may offer an attractive mechanism by which the complexity of capable adaptive routing algorithms, which are better able to balance traffic across the network under load, can be offset when the network load is light. However, if the network will usually be heavily loaded, so that only the adaptive path is used, then the presence of deterministic path incurs wasteful overhead. One could argue that providing multiple routing strategies can provide a single router architecture that could be used in different systems-on-chip with very different communication patterns and network loads, which could facilitate reuse of the router architecture, but the authors do not advance this argument. With regards to providing tiered classes of service, we feel that one should carefully evaluate when it may be better to provide independent networks. When one considers that wire costs may be much lower for a NoC than standard interconnection network, it may be more sensible to construct physically disjoint networks from simpler routers able to provide the latency and throughput characteristics required by the different classes of traffic rather than building logically disjoint networks which share physical links. Again, the authors do not evaluate the costs and benefits of this alternative.

In conclusion, while hybrid routing does appear to present a more complete solution to providing high quality of service with low latency and high throughput under a variety of network conditions, its cost in complexity requires that it only be used where such variety in traffic exists. Assuming that this is the case in some systems, we believe that hybrid routing may prove to be the best solution.

References

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

[2] B. T. Arjun Singh, William J. Dally and A. K. Gupta. Globally adaptive load-balanced routing on tori. *Computer Architecture Letters*, 3, March 2004.

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

[4] W. Dally and C. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *IEEE Transactions on Computers*, C-36, May 1987.

[5] W. J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kauffman Publishers, San Francisco, CA, USA, 2004.

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

[7] J. Duato. A new theory of deadlock-free adaptive routing in wormhole works. *IEEE Transactions on Parallel and Distributed Systems*, 4, April 1993.

[8] J. Duato and P. Lopez. Performance evaluation of adaptive routing algorithms for k-ary n-cubes. In *Proceedings of Parallel Computer Routing and Communication Workshop*, volume 853, pages 45–59. Springer-Verlag, 5 1994.

[9] M. Fulgham and L. Snyder. Triplex: A multi-class routing algorithm. *Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures*, 1997.

[10] J. Hu and R. Marculescu. Dyad: smart routing for networks-on-chip. In *DAC '04: Proceedings of the 41st annual conference on Design automation*, pages 260–263, New York, NY, USA, 2004. ACM Press.

[11] S. Konstantinidou and L. Snyder. The chaos router. *IEEE Transactions on Computers*, 43, December 1994.

[12] D. Kumar and W. A. Najjar. Combining adaptive and deterministic routing: Evaluation of a hybrid router. In *Proceeding of the Third International Workshop on Network-based Parallel Computing: Communication, Architecture, and Applications*, volume 1602. Springer, 1999.

[13] E. Rijpkema, K. Goossens, A. Radulescu, J. Dielissen, J. van Meerbergen, P. Wielage, and E. Waterlander. Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip. *IEEE Procedings on Computers and Digital Techniques*, 150, 2003.

[14] A. Singh, W. J. Dally, A. K. Gupta, and B. Towles. Goal: a load-balanced adaptive routing algorithm for torus networks. In *ISCA '03: Proceedings of the 30th annual international symposium on Computer architecture*, pages 194–205, New York, NY, USA, 2003. ACM Press.

[15] A. Singh, W. J. Dally, A. K. Gupta, and B. Towles. Adaptive channel queue routing on k-ary n-cubes. In *SPAA '04: Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures*, pages 11–19, New York, NY, USA, 2004. ACM Press.