

A High-Throughput Distributed Shared-Buffer NoC Router

Vassos Soteriou[†], Rohit Sunkam Ramanujam[‡], Bill Lin[‡], Li-Shuan Peh^{*}

[†]Cyprus University of Technology, [‡]University of California, San Diego, ^{*}Princeton University

[†]vassos.soteriou@cut.ac.cy, [‡]{rsunkamr,billin}@ucsd.edu, ^{*}peh@princeton.edu

Abstract—Microarchitectural configurations of buffers in routers have a significant impact on the overall performance of an on-chip network (NoC). This buffering can be at the inputs or the outputs of a router, corresponding to an input-buffered router (IBR) or an output-buffered router (OBR). OBRs are attractive because they have higher throughput and lower queuing delays under high loads than IBRs. However, a direct implementation of OBRs requires a router speedup equal to the number of ports, making such a design prohibitive given the aggressive clocking and power budgets of most NoC applications. In this letter, we propose a new router design that aims to emulate an OBR practically based on a distributed shared-buffer (DSB) router architecture. We introduce innovations to address the unique constraints of NoCs, including efficient pipelining and novel flow control. Our DSB design can achieve significantly higher bandwidth at saturation, with an improvement of up to 20% when compared to a state-of-the-art pipelined IBR with the same amount of buffering, and our proposed microarchitecture can achieve up to 94% of the ideal saturation throughput.

1 INTRODUCTION

NETWORKS-ON-CHIP (NoC) architectures are becoming the de facto fabric for both general-purpose chip multi-processors and application-specific systems-on-chip designs. In the design of NoCs, high throughput and low latency are both important design parameters and the router microarchitecture plays a vital role in achieving these performance goals. High throughput routers allow a NoC to satisfy the communication needs of multi- and many-core applications, or the higher achievable throughput can be traded off for power savings by using fewer resources to attain a target bandwidth. Ultimately, a router's role lies in the efficient multiplexing of packets onto network links.

Router buffering is used to house arriving flits¹ that cannot be forwarded immediately onto output links when contention arises. This buffering can be at the inputs or the outputs of a router, corresponding to an input-buffered router (IBR) or an output-buffered router (OBR). OBRs are attractive because they have higher throughput and lower queuing delays under high loads than IBRs. However, a direct implementation of an OBR would require each router to operate at P times speedup, where P is the number of router ports, which can either be realized with the router clocking at P times the link clock frequency, or the router having P times more internal buffer and crossbar ports. Both are prohibitive given the aggressive design goals of most on-chip network applications, such as high-performance chip-multiprocessors. This is a key reason behind the broad adoption of IBR microarchitectures as the de facto design choice and the extensive prior effort in the computer architecture community on aggressively pipelined IBR designs.

In this letter, we propose a new router microarchitecture that aims to emulate an OBR without the need for any router speedup. It is based on a distributed shared-buffer (DSB) router architecture that has been successfully used in high-performance Internet packet routers [4], [13]. Rather than buffering data at the output ports, a DSB router uses two crossbar stages with buffering sandwiched in between. To emulate the First-Come-First-Serve (FCFS) order of an output-buffered router, incoming packets are timestamped with the same departure times as they would depart in an OBR. Incoming packets are then assigned to one of the middle memory buffers with two constraints. First, incoming packets that are arriving at the same

time must be assigned to different buffers. Second, an incoming packet cannot be assigned to a buffer that already holds a packet with the same departure time². It has been shown in [4], [13] that $M \geq (2P - 1)$ middle buffers are necessary and sufficient to ensure that memory assignments are always possible that would emulate a FCFS output-buffered router if unlimited buffering is available.

However, just as the design objectives and constraints for an on-chip IBR are quite different from those for an Internet packet router, the architecture tradeoffs and design constraints for an on-chip DSB router are also quite different. For one, limited power and area budgets restrict practical implementations to small amounts of buffering. A new flow control protocol which can work under few buffers is necessary since NoC applications such as cache coherence protocols cannot tolerate the dropping of packets. Another reason for requiring a novel flow control protocol is the need for ultra-low latency communication in on-chip networks in order to support a wide range of delay-sensitive applications with diverse traffic characteristics. Unlike Internet routers that typically use the store-and-forwarding of packets, flit-level flow-control is widely used in on-chip routers in which bandwidth and storage are allocated at the level of flits, and flits can be forwarded immediately without waiting for the remaining flits of the same packet to arrive. Finally, another key difference is the need for on-chip routers to operate at aggressive clock frequencies necessitating the careful design of efficient router pipelines with low complexity logic at each stage. Our proposed router microarchitecture tackles all these challenges with novel designs.

The remainder of this letter is organized as follows. Section 2 provides background information on existing router architectures. Section 3 describes our proposed distributed shared-buffer router microarchitecture for NoCs. Next, Section 4 provides extensive evaluations of our proposed architecture using a detailed cycle-accurate simulator on a range of synthetic traffic, while Section 5 reviews related work. Finally, Section 6 concludes the paper.

2 BACKGROUND

It is a well-known fact that OBRs can achieve the theoretical saturation throughput under unlimited buffering and operate at lower latencies when compared to IBRs under high loads, which saturate sooner. To account for the worst case where all arriving flits need to depart via the same output port, OBRs have to either operate at P times the base clock of input routers or have a crossbar of

Manuscript submitted: 21-Feb-2009. Manuscript accepted: 06-Apr-2009. Final manuscript received: 10-Apr-2009.

1. A flit is a fixed-size portion of a packetized message.

2. This is necessary to avoid switch contention.

size $P^2 \times P^2$ instead of $P \times P$. The former configuration leads to considerable power penalties and poor clock rates, while the latter leads to huge area penalties. These two prohibitive factors have led to the wide adoption of IBRs with a wide range of microarchitectural and flow-control variations in NoC router designs [3], [5]–[10], [12].

To address the inherent speedup limitations of OBRs, DSB routers have been used for Internet routing [4], [13] to emulate the behavior of OBRs without requiring internal router speedup. Rather than buffering data at the output ports, a DSB router uses two crossbar stages with buffering sandwiched in between. The input ports are connected via a $P \times M$ crossbar to M middle memories. These M memories are then connected to the output ports through a second $M \times P$ crossbar. In every cycle, one packet can be read from and written to each middle memory.

To emulate a FCFS output-buffered router, a DSB router has to satisfy two conditions: (a) a packet is dropped by the DSB router if and only if it will also be dropped by the output-buffered router, and (b) if a packet is not dropped, then the packet must depart the DSB router at the same cycle as the cycle in which it would have departed the output-buffered router. To achieve this emulation, each packet arriving to a DSB router is *timestamped* with the cycle in which it would have departed from an output-buffered router (i.e., in FCFS order). When a packet arrives, a scheduler chooses a middle memory to which to write this incoming packet and to configure the corresponding first crossbar. Also, at each cycle, packets whose timestamp is equal to the current cycle are read from the middle memories and transferred to the outputs through the second crossbar.

In [4], [13], it was shown that a DSB router with $M \geq (2P - 1)$ middle memories and unlimited buffering in each can exactly emulate a FCFS output-buffered router with unlimited buffering. This number of middle memories is required to resolve two types of conflicts: (a) *arrival conflicts* which stipulates that no more than one packet that arrives at a given time can be written to a given middle memory and (b) *departure conflicts*, which occur when multiple packets in the same middle memory have the same timestamp and need to depart simultaneously through different outputs. With P inputs, a packet can have arrival conflicts with at most $(P - 1)$ other packets. Since there are P outputs, a packet can have departure conflicts with at most $(P - 1)$ other middle memories. Therefore, by pigeonhole principle, $M \geq (2P - 1)$ is guaranteed to suffice in finding a conflict-free middle memory assignment for all incoming packets.

3 PROPOSED DISTRIBUTED SHARED BUFFER ROUTER

3.1 Key Architectural Contributions

The proposed DSB NoC router architecture addresses the bottlenecks that exist in the data path of IBRs, which lead to lower than theoretical ideal throughput. At the same time, it tackles the inherent speedup limitations and area penalties of OBRs while harnessing their increased throughput capabilities. Although based on the DSB architecture used in Internet routers, the proposed NoC router architecture faces a number of challenges and limitations specific to the NoC domain.

First and foremost, NoC applications such as cache coherence protocols cannot tolerate dropping of packets unlike Internet protocols. To guarantee packet delivery the proposed DSB router uses credits so that flits can only be timestamped and placed in a middle memory when its next-hop router has buffers available at the corresponding input port.

The need for small buffers due to power and area constraints and the need to achieve ultra-low latency communication in NoCs in order to support a wide range of delay-sensitive applications with diverse traffic characteristics necessitate a novel flow control to efficiently

manage buffers. In the proposed architecture flow control is applied on a flit-by-flit basis, advancing each flit from an input queue towards any time-compatible middle memory and ultimately to the output link. The middle memories decouple input virtual channel queuing [2] from output channel bandwidth, as any flit can acquire any middle memory bank given that there are no timing conflicts with other flits already stored in the same middle memory buffer.

Finally, on-chip routers need to operate at aggressive clock frequencies, pointing to the need for careful design of router pipelines with low complexity logic at each stage. Our design assumes a delay- and complexity-balanced 6-stage pipeline. The proposed DSB architecture can achieve higher performance than virtual-channel IBRs with comparable buffering capacity requirements (see Section 4) while adding reasonable area and power overheads in controlling its middle memories and assigning timestamps to flits.

3.2 DSB Microarchitecture and Pipeline

Figure 1 shows the 6-stage pipeline diagram of the proposed DSB router and its corresponding microarchitecture. The input buffers are segmented into several atomic virtual channels (VCs). When a head flit arrives at a VC, the route computation (RC) stage of the DSB pipeline determines the output port of the flit based on its coordinates. This works in the same way as the route computation logic in an IBR. We assume dimension-ordered routing.

The remaining pipeline stages of a DSB router are substantially different from those of IBRs. Instead of arbitrating for free virtual channels (buffering) and passage through the crossbar switch (link), flits in a DSB router compete for two resources: middle memory buffers (buffering) and a unique time at which to depart from the middle memory to the output port (link). The timestamping stage (TS) deals with the timestamp resource allocation. Timestamps refer to the future cycle in which a flit will be read from a middle memory, through the second crossbar, XB2, onto the output port. In each cycle, flits at the head of the input VCs issue requests to the timestamper, which keeps track of the status of middle memories and output ports with a middle memory and output port reservation table. The timestamper randomly picks a winning VC from every input port and tries to assign the earliest possible departure time for the output port requested by the flit in the selected VC. Essentially, the timestamps for each output port are assigned to flits on a FCFS basis, emulating an OBR. Input ports are assigned fixed priorities so that when flits from multiple inputs request for the same output port, timestamps can be assigned in the order of input priority.

Conflict resolution (CR) and virtual channel allocation (VA) comprise the third pipeline stage of the DSB router. The CR and VA operations are carried out in parallel. Once flits are assigned timestamps in the TS stage, the CR stage tries to find a conflict-free middle memory assignment. As mentioned earlier, there are two kinds of conflicts in shared buffer routers – *arrival conflicts* and *departure conflicts*. Arrival conflicts are handled by assigning a different middle memory to every input port with timestamped flits. Departure conflicts are avoided by ensuring that the flits stored in the same middle memory have unique timestamps. These restrictions are enforced due to the fact that middle memories are uni-ported and only one flit can be written into (using XB1) and read from (using XB2) a middle memory in a given cycle.

The virtual channel allocator arbitrates for free virtual channels at the input of the next-hop router in parallel with conflict resolution. The VC allocator maintains two lists of VCs – a *reserved* pool and a *free* pool. VC allocation is done by picking the next free output VC from the *free* VC list of the given output port. Additionally, when output VCs are freed, their VC number is moved from the *reserved* VC list to the end of the *free* VC list. If a free VC exists and the flit

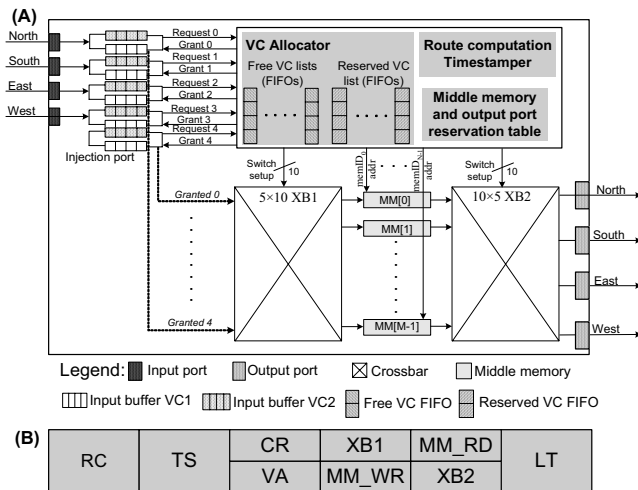


Fig. 1. Distributed shared-buffer (A) router microarchitecture with 10 middle memory banks and (B) its 6-stage pipeline: (1) route computation (RC), (2) timestamping (TS), (3) conflict resolution (CR) and virtual channel allocation (VA), (4) first crossbar traversal (XB1) and middle memory write (MM_WR), (5) middle memory read (MM_RD) and second crossbar traversal (XB2), and (6) link traversal (LT).

is granted a middle memory $MM[j]$, it subsequently proceeds to the fourth pipeline stage, where it traverses the first crossbar (XB1) and is written to its assigned middle memory (MEM_WR). If no free VC exists (all VCs belong to the reserved VC list) or if CR fails to find a conflict-free middle memory, the flit has to be re-assigned a new timestamp and therefore re-enters the TS stage.

When its timestamp matches the current router time, a flit is read off from the middle memories (MM_RD) and passed through the second crossbar (XB2) in the fifth pipeline stage. Finally, in the link traversal (LT) stage, flits traverse the output links to reach the downstream router.

These stages were architected based on FO4 delay calculations targeted to fit within an aggressive 3 GHz clock. It must also be noted that while this DSB microarchitecture is applicable to 2D routers, it can be readily expanded to higher radix architectures.

4 EVALUATION RESULTS

4.1 Throughput Evaluation

In this section, we evaluate the effectiveness of our proposed DSB router by comparing it with a baseline IBR with virtual channel flow control. We implemented flit-level simulators for both router architectures. The baseline IBR simulator has a five-stage pipeline comprising route computation, VC allocation, switch arbitration, switch traversal, and link traversal. The DSB simulator models the six-stage pipelined architecture described in Section 3. Both simulators support k -ary 2-mesh topologies with their corresponding pipelined routers, which are targeted to be clocked at a frequency of 3GHz. Packets are composed of five 128-bit flits with each flit transported in 1 link cycle over links of 384Gbps bandwidth. Dimension-ordered X-Y routing (DOR) is used for both network architectures, where packets are first routed in the X-dimension, followed by the Y-dimension. We use the simulators to evaluate the average routing delays under different injection loads. For each experiment we completed a simulation run of a million cycles. The latency of a packet is measured as the delay between the time the header flit is injected into the network and the time the tail flit is consumed at the destination. The performance of the following router microarchitectures are compared:

- IBR-175: An input-buffered router with 175 flits of aggregate buffering. The buffers at each input port are partitioned into 7 VCs with 5-flit buffers per VC.
- IBR-300: An input-buffered router with 300 flits of aggregate buffering. The buffers at each input port are partitioned into 12 VCs with 5-flit buffers per VC.
- DSB-175: A DSB router with 175 flits of aggregate buffering. The buffers are divided between 5 middle memory banks with 10-flit buffers per bank and 125 input buffers comprising 5 VCs with 5-flit buffers per VC at each input port.
- DSB-300: A DSB router with 300 flits of aggregate buffering. The buffers are divided between 10 middle memory banks with 10-flit buffers per bank and 200 input buffers comprising 8 VCs with 5-flit buffers per VC at each input port.

For both the IBR and DSB architectures, we tried to evaluate the best performing configuration for the given design point. The division of input buffers into VCs and middle memory buffers into banks was done in a manner that maximizes throughput. For example, 7 VCs with 5-flit buffers deliver the highest throughput for IBR-175, rather than 5 VCs with 7-flit buffers.

TABLE 1
Normalized saturation throughput comparison

Traffic Pattern	IBR-175	IBR-300	DSB-175	DSB-300
Uniform	78%	78.5%	89%	94%
Complement	84%	84%	92%	94%
Tornado	81%	81.75%	91.5%	93.75%

Figure 2 shows the simulation results. The average packet latency is plotted against the normalized injected load (normalized to the ideal throughput for DOR) for uniform, complement and tornado traffic. Table 1 shows the saturation throughput of the four router architectures evaluated for each of the three traffic patterns. As Table 1 shows, increasing the amount of buffering for an IBR from 175 flits to 300 flits results in little or no improvement in the saturation throughput. On the other hand, increasing the amount of buffering for our proposed DSB router from 175 flits to 300 flits considerably improves the saturation throughput, from 89% to 94% in the case of uniform traffic. Overall, our proposed DSB router, using the DSB-300 configuration, can outperform both the IBR-175 and IBR-300 configurations by 20%, 12%, and 16% for uniform, complement, and tornado traffic, respectively, and achieve up to 94% of the ideal saturation throughput.

4.2 Power Consumption Evaluation

Estimates of power consumption for the four router microarchitectures are shown in Table 2. Each line in the table compares a DSB router to an IBR that has the same amount of buffering. We used the power models in Orion 2.0 [15] for our analysis. The models used are for a 65nm process technology and include both the dynamic and leakage power components. The operational frequency used is 3GHz and 30% activity is assumed at each input port. As can be seen in Table 2, the DSB-175 router consumes a factor of approximately 1.5 more power than a corresponding IBR-175 router. The high power dissipation in the DSB architecture can be attributed to the presence of an extra crossbar and a more complex arbitration scheme compared to an IBR. Although the DSB-300 router achieves near-ideal throughput, it pays a substantial penalty in terms of power consumption, a factor of approximately 2 more than a corresponding IBR-300 router. Its high power consumption is due to the large 5×10 and 10×5 crossbars needed for the 10 middle memory banks and a complex arbiter needed for timestamping and conflict resolution.

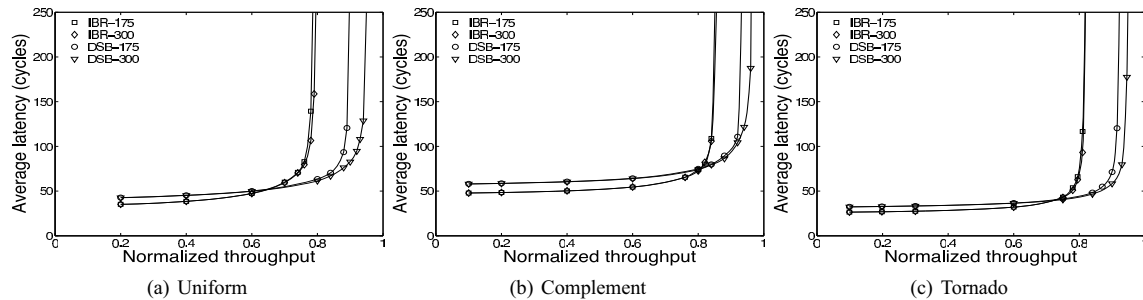


Fig. 2. Throughput comparison of different router architectures

TABLE 2
Router power comparison

Router Config.	Power (mW)	Router Config.	Power (mW)	Power penalty	
				router	tile
IBR-175	747.85	DSB-175	1136.27	1.52	1.15
IBR-300	931.31	DSB-300	1860.46	2.00	1.28

Although the increased power cost per router for a DSB router is substantial when compared to the IBRs, the overall power increase for an NoC application is often less. For example, in the Intel Teraflops chip-multiprocessor [14], the router and links together at each processor tile only contribute to around 28% of the total tile power. Based on this percentage figure, the power cost per tile with a DSB router would increase by only a factor of approximately 1.15 or 1.28 when compared to the respective cases of incorporating an IBR-175 or an IBR-300 router. We believe that the increased power cost is justified for applications that demand high bandwidth.

5 RELATED WORK

To the best of our knowledge, all previously proposed on-chip network routers comprise input-buffered architectures, where routers have buffers at the input ports for housing flits that are multiplexed onto the crossbar switch and links. Sophisticated input-buffered router microarchitectures have been proposed for extending throughput. For instance, flit-reservation flow control [12] sends control flits ahead of data flits, and timestamps these control flits so that buffers can be allocated just-in-time when data flits arrive. This, however, still relies on input buffers. Vichar [10] is another input-buffered architecture that tackles throughput. It extends throughput by supporting a variable number of virtual channels, so that the router can use as many VCs as the current number of flows in order to maximize the multiplexing of these flows. Another input-buffered architecture that also pushes throughput is express virtual channels [6], which does so by pinning virtual channels across multiple hops, so when a packet grabs an express multi-hop VC, it bypasses the router pipeline at intermediate routers. This causes flow speedup and reduces the pressure on arbitration, thus extending throughput. As all these prior designs are input-buffered, they are unable to share their buffering across input ports, unlike our proposed DSB router whose middle memories can be multiplexed across multiple input ports, thereby emulating the behavior of output-buffered routers.

There have been several input-buffered router proposals that target network latency, making single-cycle routers feasible, such as speculative allocation [8], [9], [11] and lookaheads [3], [7]. There have also been router microarchitectures that specifically target network power, such as RoCo [5] which uses two smaller crossbars to achieve similar bandwidth as a classic router at lower energy, and the work in [16] which proposes straight-through buffers and segmented crossbars to optimize power. These techniques are all orthogonal to our proposal as they do not target throughput. For instance, we can similarly

use lookaheads to shorten our pipeline, and use smaller dimension-sliced crossbars, straight-through buffers and segmented crossbars to improve our energy-delay.

Finally, as already mentioned, distributed shared-buffer routers [4], [13], which can emulate an output-buffered router without router speedup, have been successfully used for Internet routing. In addition, Chuang et al. [1] showed that a combined-input-output-buffered router can also emulate an output-buffered router. However, this emulation requires a router speedup of 2 along with a complex and impractical matching problem, both of which are hard to achieve with aggressive design targets.

6 CONCLUSIONS

In this letter, we proposed a distributed-shared-buffer (DSB) router for on-chip networks. DSB routers are successfully used in Internet routers to emulate the ideal throughput of output-buffered routers, but porting them to on-chip networks that face much more stringent constraints presents tough challenges. Our proposed DSB router for NoCs achieves substantially higher throughput than input-buffered routers, the currently predominant microarchitecture of on-chip network routers.

REFERENCES

- [1] Shang-Tse Chuang, A. Goel, N. McKeown, and B. Prabhakar. Matching output queueing with a combined input/output-queued switch. *IEEE Journal on Selected Areas in Communications*, Jun 1999.
- [2] W. J. Dally. Virtual-channel flow control. In *ISCA*, May 1990.
- [3] P. Gratz et al. Implementation and evaluation of on-chip network architectures. In *ICCD*, Oct. 2006.
- [4] S. Iyer, R. Zhang, and N. McKeown. Routers with a single stage of buffering. In *ACM SIGCOMM*, September 2002.
- [5] Jongman Kim, C. Nicopoulos, and Dongkook Park. A gracefully degrading and energy-efficient modular router architecture for on-chip networks. *ISCA*, pages 4–15, 2006.
- [6] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha. Express virtual channels: Towards the ideal interconnection fabric. In *ISCA*, June 2007.
- [7] A. Kumar et al. A 4.6Tbits/s 3.6GHz single-cycle NoC router with a novel switch allocator in 65nm CMOS. In *ICCD*, Oct. 2007.
- [8] S. S. Mukherjee et al. The Alpha 21364 network architecture. *IEEE Micro*, 22(1):26–35, Jan./Feb. 2002.
- [9] R. Mullins et al. Low-latency virtual-channel routers for on-chip networks. In *ISCA*, June 2004.
- [10] C. A. Nicopoulos et al. ViChAR: A dynamic virtual channel regulator for network-on-chip routers. In *MICRO*, Dec. 2006.
- [11] Li-Shiuan Peh and William J. Dally. A delay model and speculative architecture for pipelined routers. In *HPCA*, Jan. 2001.
- [12] Li-Shiuan Peh and W.J. Dally. Flit-reservation flow control. *HPCA*, pages 73–84, 2000.
- [13] A. Prakash, A. Aziz, and V. Ramachandran. Randomized parallel schedulers for switch-memory-switch routers: Analysis and numerical studies. In *IEEE INFOCOM*, March 2004.
- [14] S. Vangal et al. An 80-tile 1.28TFLOPS network-on-chip in 65nm CMOS. In *ISSCC*, Feb. 2007.
- [15] H.-S. Wang et al. Orion: A power-performance simulator for interconnection networks. In *MICRO*, Nov. 2002.
- [16] H.-S. Wang et al. Power-driven design of router microarchitectures in on-chip networks. In *MICRO*, Nov. 2003.