Bubble Coloring: Avoiding Routing- and Protocol-induced Deadlocks With Minimal Virtual Channel Requirement

Ruisheng Wang Lizhong Chen Timothy Mark Pinkston

Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, California, USA
{ruishenw, lizhongc, tpink}@usc.edu

ABSTRACT

Handling routing- and protocol-induced deadlocks is a critical issue in designing a reliable communication system. Generally, to avoid these two types of deadlocks without losing routing freedom requires a large amount of virtual channels (VCs), which imposes significant negative effects on router power, energy and frequency. In this paper, we propose a virtual cut-through switched Bubble Coloring (BC) scheme, which can avoid both routing- and protocol-induced deadlocks and allow fully adaptive routing on any topology without the need for multiple virtual channels. Results from both synthetic and full-system simulation show that, compared to a conventional deadlock-free scheme with 4VCs (i.e., XY_adaptive_4VC), our BC scheme with the minimal 1VC (i.e., BC_1VC) can reduce router energy and area by up to 51.2% and 58.3%, respectively, and has comparable performance at the same time. As the proposed BC scheme does not require multiple virtual channels, it also reduces the complexity of router arbitration logic, which brings the opportunity to increase router frequency and further improve system performance.

Categories and Subject Descriptors

C.1.2 [**Computer Systems Organization**]: Multiprocessors—*Interconnection architectures*

General Terms

Design, Experimentation, Performance

Keywords

Bubble flow control; Routing-induced deadlock; Protocol-induced deadlock; Minimal virtual channel requirement

1. INTRODUCTION

As the number of processors integrated in a system continues to grow, networks on-chip (NoCs) and interconnection networks more generally have become essential components of modern chip multiprocessors (CMPs) and parallel processing systems. In order to provide efficient and reliable communication through the networks, routing- and protocol-induced deadlocks, which are catastrophic to the communication system, should be efficiently handled.

Routing-induced deadlocks occur when there is a cyclic dependency between resources created by the packets transported through various paths in the network. To break such cyclic dependence, the turn model and its extensions [15, 7, 13] have been proposed, which reduce the degree of routing freedom by disallowing certain paths between source and destination nodes. In order to maintain full routing freedom (i.e., to support fully adaptive routing), Duato's Protocol [11] is proposed to avoid routing-induced deadlock by using an additional virtual channel.

Deadlock freedom provided by the schemes mentioned above is based on the consumption assumption [17] where the end node will consume all packets from the network once the destination is reached. However, when interactions and dependencies are created between packets of different message classes at network endpoints, that assumption may not be valid and protocol-induced deadlocks (also known as message-dependent deadlocks) may occur [17]. The conventional solution for protocol-induced deadlocks is to have separate virtual networks for each message class [18]. As the number of message classes increases, the virtual channel (VC) requirement rises proportionally. Thus, current approaches for designing routing- and protocol-induced deadlock-free schemes with complete routing freedom require a large number of VCs. However, implementing such a large number of VCs in routers has considerable negative impacts on router area, power and frequency. Hence, it is important to provide a scheme to avoid routing- and protocol-induced deadlocks with minimal requirement on the number of virtual channels.

In this paper, we propose a virtual cut-through (VCT) switched Bubble Coloring scheme that enables fully adaptive routing on any topology without the need for multiple virtual channels while still avoiding both routing- and protocol-induced deadlocks.

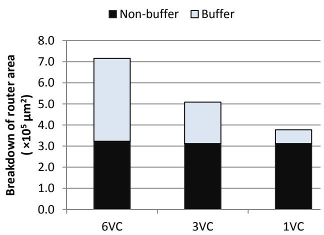
Our contributions are summarized as follows:

1. A topology-agnostic fully adaptive Bubble Ring (BR) scheme is proposed to avoid routing-induced deadlocks without the need for multiple virtual channels.

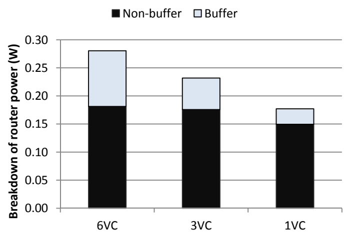
2. A Bubble Coloring (BC) scheme, an extension of the Bubble Ring scheme, is proposed to avoid protocol-induced deadlocks without the need for multiple virtual channels.

3. The results from both synthetic and full system simulation show that, compared to a conventional deadlock-

ICS'13, June 10–14, 2013, Eugene, Oregon, USA.



(a) Router area



(b) Router power

Figure 1: Breakdown of router area and power

free scheme with 4VCs (i.e, XY_adaptive_4VC), our BC scheme with the minimal 1VC (i.e., BC_1VC) can reduce router energy and area up to 51.2% and 58.3%, respectively, and has comparable performance at the same time. As the proposed BC scheme does not require multiple virtual channels, it also reduces the complexity of router arbitration logic which brings the opportunity to increase router frequency and further improve system performance.

The rest of the paper is organized as follows. Section 2 provides motivation for reducing VC cost and background for bubble flow control. Section 3 proposes our Bubble Ring and Bubble Coloring schemes to avoid routing- and protocol-induced deadlocks, providing a proof sketch for deadlock freedom. Section 5 describes our evaluation methodology, and Section 6 presents simulation results. Related work is discussed in Section 7. We conclude the paper and discuss some future work in Section 8.

2. BACKGROUND AND MOTIVATION

2.1 Need for Reducing Virtual Channel Cost

Deadlocks in interconnection networks include routing-induced deadlocks that are caused by cyclic dependence in routing functions and protocol-induced deadlocks that are caused by dependencies among different types of messages (e.g., a reply message depends on a request message). To avoid these network abnormalities, virtual channels (VCs) [9, 10] have been used extensively in many deadlock-avoidance schemes. For example, messages in the MOESI directory cache protocol [14] can be classified into three dependent classes. Within each message class, two VCs can be used to implement deadlock-free adaptive routing in a 2-D mesh, for instance, by applying Duato's Protocol [11] – one VC employs deterministic routing (e.g., XY-routing) to provide escape paths while the other VC acts as an adaptive resource to enable fully adaptive routing. Then, to avoid protocol-induced deadlocks, at least three independent virtual networks (VNs) are needed to separate different dependent message classes.

The above typical way of avoiding deadlocks, however, is achieved at a large overhead of high VC count manifested in router area, power and frequency. To illustrate the overhead, Figure 1 plots the breakdown of router area and power for different number of VCs at 45nm with 3GHz frequency and 1.1V operating voltage (more details of the simulation infrastructure are described in Section 5). The first bars in Figure 1 (a) and (b) correspond to the above example using 6 VCs to avoid routing- and protocol-induced deadlocks. As can be seen, over 54% of the router's area and 35% of the

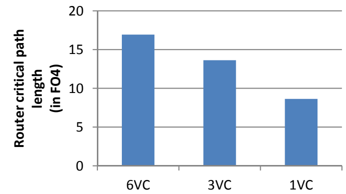


Figure 2: Router critical path length

router power are consumed by the virtual channel buffers, demonstrating a great need to minimize VC requirements.

The overhead of VCs can be reduced in both routing- and protocol-induced deadlock avoidance schemes. Consider first the impacts of reducing VC requirement in avoiding routing-induced deadlocks. The second bars (i.e., the 3VC bar) in Figure 1 (a) and (b) show that, if a deadlock-free scheme only needs one VC per VN to avoid deadlocks in the routing algorithm, the overall buffer area and power can be reduced by 50% and 43%, respectively (which corresponds to a savings of 29% and 18% of total router area and power, respectively). However, to compensate for the performance degradation with fewer VCs, the design of such deadlock-free schemes becomes very challenging in order to provide a higher degree of routing freedom, in particular, to support fully adaptive routing with only 1 VC per VN. In Section 3.1, the Bubble Ring scheme is proposed to achieve the potential benefits of reducing VCs while being able to preserve full adaptivity in routing algorithms.

Another effective but more challenging way to reduce VC requirement is to devise schemes to handle protocol-induced deadlocks more efficiently. In the ideal form, such schemes should avoid all possible deadlocks with minimally 1 VC in total, regardless of the number of dependent message classes. The rightmost bars in Figure 1 (a) and (b) highlight the advantages. Compared to the scheme with 6VCs, the buffer area and power is reduced by 83% and 74%, respectively (which corresponds to a savings of 47% and 37% of the router area and power, respectively). However, to realize those schemes, significant modifications are needed from previous approaches. Section 3.2 proposes a new innovative scheme (i.e., Bubble Coloring) which can avoid protocol-induced deadlocks without requiring multiple virtual channels.

In addition to saving resources, the number of VCs also has a considerable impact on the complexity of router control logic, particularly the VC allocator (VA) and switch allocator (SA). The SA is affected as the input-arbitration step in SA is to select one VC from multiple VCs within the same physical channel to participate in the output-arbitration step in SA. The VA and SA stages typically lie on the critical path of the router pipeline and determine the router frequency [24]. Figure 2 compares the length of the router critical path, calculated from the delay model proposed by Peh and Dally [19]. As shown in the figure, the router critical path can be shortened by 19% and 48% in schemes with 3VCs and 1VC, respectively, indicating a considerable reduction in router latency and increase in throughput when configured at maximum achievable frequency.

In summary, considering the direct impacts of reducing the number of VCs on router area, power and frequency, as well as the indirect impact on network and overall system performance, it is imperative to devise efficient deadlock-free schemes that minimize the VC requirement in avoiding routing- and protocol-induced deadlocks.

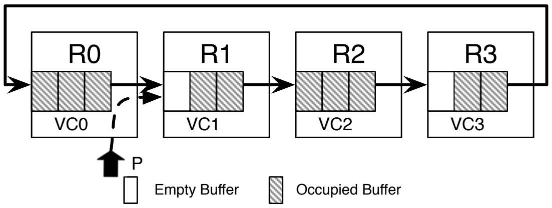


Figure 3: Bubble flow control

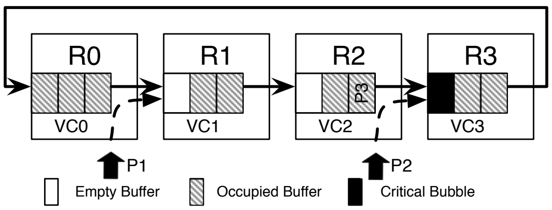


Figure 4: Critical bubble scheme

2.2 Bubble Flow Control and Critical Bubble Scheme

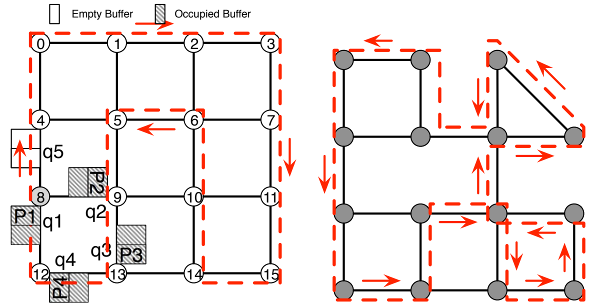
Deadlocks can be handled by the routing algorithm and/or the flow control mechanism. Prior work [21, 6, 5] has been proposed to use flow control instead of routing restrictions on additional virtual channels as a means of avoiding deadlocks to reduce VC cost. But these works have some limiting constraints.

Bubble Flow Control (BFC) [21] has been proposed to avoid routing-induced deadlocks in any embedded ring in tori. A bubble in BFC is an empty packet-sized buffer. According to the theorem of BFC, for any unidirectional ring in a given dimension of a torus network, the ring is deadlock-free if there is at least one bubble located anywhere in the ring after packet injection. Figure 3 depicts an example for a unidirectional ring of an arbitrary dimension in a torus. Each small rectangle represents a packet-sized buffer. Packet P is allowed to inject into VC1, as after the injection there is still a bubble in VC3 which ensures deadlock freedom within the ring. Along with Dimension-Order Routing (DOR) that avoids deadlocks across dimensions by restricting the routes provided to packets, BFC is free from routing-induced deadlock in tori.

To implement the theorem of BFC, global information needs to be communicated at each local node. Otherwise, as in the above example, packet P at R0 has only the information of VC1 (which is stored in the upstream router R0) but is not aware of the buffer utilization of any other node. As a result, P does not have sufficient information to make an optimal decision. The Critical Bubble Scheme (CBS) proposed in [6] solves this difficulty by using the notion of a *critical bubble* to convey the global information locally. As shown in Figure 4, an empty packet-sized buffer in any router along the directional ring is marked as the critical bubble for the entire ring (e.g., the black rectangle in VC3). This critical bubble can only be used by packets that are already in the ring, but cannot be used by injecting packets. For instance, P1 is allowed to inject as there is at least one non-critical bubble in VC1, whereas P2 is not allowed to inject as the only bubble in VC3 is the critical bubble. Essentially, the absence of the critical bubble at VC1 indicates the existence of a bubble (i.e., the critical bubble) elsewhere in the ring, thus providing implicit global information.

The CBS only needs each buffer to be at least one packet-size, and if more than one packet is injecting simultaneously (e.g., P1 and P2 are injecting in the same cycle), it is clear which one should be injected and which one should not, based only on local information. In addition, the critical bubble can be displaced backward in the ring either proactively (e.g., if VC2 is empty and P2 is waiting for injection), or by the incoming packet (e.g., P3 occupying the critical bubble will mark the newly freed buffer in VC2 as the critical bubble so that logically the critical bubble is simply transferred backwards rather than consumed). In both cases, the existence of the critical bubble in the ring is always maintained, ensuring that not all buffers are fully occupied. However, like BFC, CBS is applicable only to torus-like topologies. Moreover, two VCs per VN are needed to implement adaptive routing and separate VNs are required to avoid protocol-induced deadlocks, making both CBS and BFC susceptible to high VC count.

In this paper, we extend the basic notion of bubble flow control and critical bubbles as the basis for our proposed approach to provide an efficient deadlock-free scheme on any topologies but without the need for multiple VCs.



(a) A 4 × 4 mesh (b) An irregular network

Figure 5: Topologies with virtual rings

3. PROPOSED SCHEME

3.1 Avoiding Routing-induced Deadlocks

The Bubble Ring (BR) scheme is comprised of a bubble flow control mechanism and a routing algorithm.

The basic idea of Bubble Ring flow control is first to construct a unidirectional virtual ring that connects all the nodes in the network. Figure 5 shows two examples of virtual rings (represented by dash lines) for a 4×4 mesh and an irregular topology[1]. The ring is "virtual" because there is no additional links or queues for implementing the ring. Only nominal control logic is needed for the routers to associate the input ports and output ports that are in this virtual ring. Note that this ring is not necessarily a Hamiltonian cycle since a node can be visited more than once as is shown in Figure 5 (b). For performance reasons, the virtual ring should be as short as possible. Intuitively, such a virtual ring can always be constructed if the topology is strongly connected. With that, the critical bubble scheme [6] can be applied in such a way to make sure there is always at least one free buffer in the virtual ring. In this way, packets in the ring can always be able to move along the ring to reach any destination.

[1]This is one of many possible topologies which could be formed by construction or due to faults occurring in a base network such as a 2D mesh with express links.

The routing algorithm can use the virtual ring as an escape path. The packet will first try to use its preferred output port based on a specified adaptive routing subfunction which provides a subset of the total routing options supplied by the routing algorithm. If there is no preferred output port available, the packet can be forwarded to the escape outport (i.e., the outport in the virtual ring) provided by the escape routing subfunction also defined by the routing algorithm. For example, in Figure 5 (a), packets P1, P2, P3 and P4 are at the head of queue q1, q2, q3 and q4, respectively. Each queue is in the associated router R8, R9, R12 and R13. Assume that all these four queues are full. With the specified adaptive routing subfunction, a cyclic dependency on these queues may occur. For example, the adaptive routing subfunction may supply only q2, q3, q4 and q1 to packet P1, P2, P3 and P4, respectively. In this situation, our Bubble Ring routing subfunction can avoid deadlocks by providing a way of escape from such knotted cyclic dependency. In this example, since P1 cannot choose its preferred output port supplied by the adaptive routing subfunction (i.e., q2), it can be forwarded to the output port in the virtual ring, i.e., q5. Note that because BR flow control guarantees that there is always at least one free buffer in the virtual ring, P1 can move to q5 eventually even if q5 is temporally full. After P1 moves to q5, packets P2, P3 and P4 can continue to move to their preferred output ports so that this deadlock is avoided. Note that in order to allow packets from any direction to move into the virtual ring, we assume that packets can make U turns. Different from Duato's Protocol [11] where the escape path consists of the resources from an additional set of virtual channels, the escape path in our BR scheme is comprised of the same set of resources used for the adaptive paths. For example, q3, q4 and q1 are the queues for both the escape path and the preferred adaptive output queues for P2, P3 and P4.

The formal description is as follows. To facilitate the discussion throughout the paper, we first introduce some basic definitions derived from [11, 12, 21].

Definition 1. An *interconnection network*, I, is a strongly connected directed graph $I = G(N, Q)$. The vertices of the graph, N, represent the set of processing nodes. The arcs of the graph, Q, represent the set of queues associated to the communication links interconnecting the nodes. Each queue $q_i \in Q$ has capacity of $cap(q_i)$ measured in the number of packets, and the number of packets currently stored in the queue is denoted as $size(q_i)$.

The set of queues Q is divided into three subsets: injection queues Q_I, delivery queues Q_D, and network queues Q_N. Each node uses a queue from Q_I to send packets that travel through the network employing queues from Q_N, and when they reach their destination they enter a queue from Q_D. Therefore, packets are routed from a queue in the set $Q_{IN} = Q_N \cup Q_I$ to a queue in the set $Q_{ND} = Q_N \cup Q_D$. Let Q_{VR} be a subset of Q_N consisting of all the network queues along some minimal virtual ring that spans all nodes of I.

Definition 2. A routing function, $R : Q_{IN} \times N \rightarrow \wp\{Q_{ND}\}$, provides a set of alternative queues to route a packet p located at the head of any queue $q_i \in Q_{IN}$ to its destination node $n_d \in N$. $\wp\{Q_{ND}\}$ denotes the power set of Q_{ND}. A deterministic routing function provides only one alternative queue, $R(q_i, n_d) = q_j$, $q_j \in Q_{ND}$. A routing subfunction, R_s, for a given routing function R is a routing function defined in the same domain as R but its range (i.e., set of alternative next queues) is restricted to a subset $Q_{NDs} \subseteq Q_{ND}$. Let R_{VR} be the routing subfunction that provides a next-hop queue in the virtual ring, i.e., $R_{VR}(q_i, n_d) = q_j$, $q_j \in Q_{VR}$.

Definition 3. A flow control function, $F : Q_{IN} \times Q_{ND} \rightarrow \{true, false\}$, determines the access permission for a packet p located at the head of queue q_i to enter queue $q_j \in R(q_i, n_d)$. Thus, the packet p is allowed to advance from q_i to q_j if $F(q_i, q_j) = true$.

Let $cb(q_j)$ be the number of critical bubbles at q_j. In this paper, we assume there is only one critical bubble in the virtual ring, so $\sum_{q_i \in Q_{VR}} cb(q_i) = 1$.

To keep one free packet-sized buffer in the virtual ring at all times, our flow control on the bubble ring, F_{BR}, is based on critical bubble flow control:

$$F_{BR}(q_i, q_j) = true$$
$$if \begin{cases} size(q_j) \le cap(q_j) - 1 & \text{when } q_i \in Q_{VR} \text{ or } q_j \notin Q_{VR} \\ size(q_j) \le cap(q_j) - cb(q_j) - 1 & \text{when } q_i \notin Q_{VR} \text{ and } q_j \in Q_{VR} \end{cases} \quad (1)$$

When a packet outside the virtual ring wants to move into the ring (i.e., $q_i \notin Q_{VR}$ and $q_j \in Q_{VR}$), it cannot occupy the critical bubble. In other words, if the critical bubble is present at q_j (i.e., $cb(q_j) = 1$), the packet needs at least two free buffers to move into q_j (i.e., $size(q_j) \le cap(q_j) - 2$). Otherwise, it just needs at least one free buffer.

For the rest of the conditions, if the packet wants to (1) travel outside the ring (i.e., $q_i \notin Q_{VR}$ and $q_j \notin Q_{VR}$), or (2) go out of the ring (i.e., $q_i \in Q_{VR}$ and $q_j \notin Q_{VR}$), or (3) stay in the ring (i.e., $q_i \in Q_{VR}$ and $q_j \in Q_{VR}$), it only needs at least one free buffer at q_j. Note that based on the Critical Bubble Scheme [6], if a packet in the ring occupies the critical bubble, the newly freed buffer in the upstream router will be marked as the the new critical bubble, which guarantees that there is still at least one free buffer marked as a critical bubble in the ring.

This flow control mechanism will guarantee that there is always a free bubble inside the virtual ring. Due to this free buffer, the packets in the ring cannot be blocked so that the routing algorithm can use this ring as its escape path to avoid routing-induced deadlocks. The formal description of a routing function based on our BR scheme is the following:

$$R(q_i, n_d) = R_{adaptive}(q_i, n_d) \cup R_{VR}(q_i, n_d) \quad (2)$$

The packet will try to choose its preferred queues based on a given adaptive routing subfunction $R_{adaptive}(q_i, n_d)$ (e.g., Region Congestion Aware routing [16]) first, and use the queue in the virtual ring (i.e., $q_j \in Q_{VR}$) supplied by the escape routing subfunction (i.e., R_{VR}) as its alternative escape path.

Lemma 1. *Packets in virtual ring queues provided by the routing subfunction R_{VR} can always make forward progress under F_{BR} flow control.*

Proof Sketch. This can be proved by contradiction. Assume a deadlock occurs, such that no packet in the virtual ring can move to the next queue. F_{BR} flow control keeps the critical bubble inside the ring. Without losing generality, assume the critical bubble is at router i (i.e., R_i). If there is a packet in the upstream router (i.e., R_{i-1}) along the virtual ring, the routing function supplies a path for the packet to move to router R_i when there is no other preferred output ports for this packet available. Due to the presence of the critical bubble (i.e., one free bubble) at router i, bubble ring flow control will guarantee the packet can move to router R_i. This contradicts with the deadlock assumption. If there is no packet in the upstream router R_{i-1}, we can continue

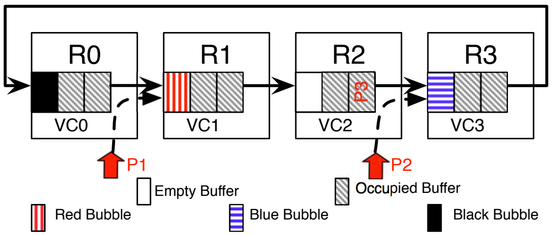


Figure 6: Bubble Coloring flow control

to search backward along the ring until we find a packet at router R_{i-j}. Due to the free buffer available at that downstream router, this packet can move. □

THEOREM 1. *An interconnection network I with a routing function R and flow control F_{BR} is free from routing-induced deadlocks if there exists a virtual ring queue supplied by R_{VR} and controlled by F_{BR} such that, for each cycle, at least one packet occupying the head of a queue can be allocated to the virtual ring queue.*

PROOF SKETCH. If there are packets in the virtual ring, based on Lemma 1, we can always find a packet able to move along the ring. If the ring is empty, the packets outside the ring can make progress by injecting into the queues in the virtual ring as provided by the routing algorithm. □

3.2 Avoiding Protocol-induced Deadlocks

Although our proposed BR scheme is able to avoid routing-induced deadlocks, it is still susceptible to protocol-induced deadlocks. For example, assume there are two message classes A and B, where class A depends on class B (e.g., request-reply classes). Also assume that all the queues in the network are full except for one critical bubble in the ring and that all the packets in the virtual ring are from message class A. In this situation, packets from class A cannot be consumed because they are waiting for the completion of packets from class B. Packets from class B cannot move forward either, because the virtual ring is fully occupied by packets from class A, and there is no room for them to inject. Hence, it is possible for protocol-induced deadlocks to occur under the BR scheme.

To avoid protocol-induced deadlocks, we propose Bubble Coloring which is a novel extension of our Bubble Ring scheme. The Bubble Coloring (BC) scheme guarantees that packets in one message class can always reach their destinations even though packets of other message classes are blocked from being consumed at end nodes. The basic idea is to reserve one additional bubble (i.e., one free packet sized buffer) in the virtual ring for each message class besides the original critical bubble used for avoiding routing-induced deadlocks. These bubbles are represented by different colors corresponding to distinct message classes. The colored bubble for a given message class serves as a normal free buffer for the packets from its own message class (i.e., it can be used for injection), but it serves as a critical bubble for all other message classes (i.e., it cannot be used for injection).

Figure 6 provides an example of the BC scheme. As is shown, VC0, VC1, VC2 and VC3 form the virtual ring. The red packet, $P1$, which is out of the ring can move into the ring (i.e., into VC1) because it can occupy the bubble having the same color (i.e., red). However, Packet $P2$ with red color cannot move into the ring (i.e., into VC3) because it cannot occupy a bubble with a different color (i.e., blue). Packet P3 already inside the virtual ring can move to VC3 and pull the blue bubble to VC2 at the same time even it has a different color from the bubble. This is because it is already in the virtual ring as opposed to trying to inject into the ring. By reserving such a colored bubble in the virtual ring for every message class, packets from the same message class can always have a chance to move into the ring, i.e., not be blocked by packets from the other message classes. Since the packets in the ring cannot be blocked due to the existence of the original critical bubble, no protocol-induced deadlocks can happen.

To make sure colored bubbles will always stay in the virtual ring, the BC scheme has two additional rules beyond the BR scheme: (1) Once a colored bubble is occupied by a packet with the same color at injection into the virtual ring, the packet will be marked as having consumed the colored bubble and carry forward the color mark for that message class as it travels along the ring. (2) When the packet marked as having consumed a colored bubble moves out of the ring, it will leave the color mark to the newly freed buffer so that the colored bubble will reappear in the ring. These two rules will continue a colored bubble in the virtual ring for reuse by other packets having the same color at the injection into the ring.

We assign a distinct color to the original critical bubble, i.e., black, such that there is no black message class. Therefore, no packet can use the black bubble for injection. In this sense, our Bubble Ring scheme can be considered as a special case of our Bubble Coloring scheme.

The formal description of the proposed BC scheme is as follows:

Let $cb_k(q_j)$ be the number of bubbles with color k at queue j and $color(q_i)$ be the color of the packet at the head of q_i. Assume there are $M+1$ colors for M different message classes, where one color (i.e., black) is designated as the original critical bubble. In this paper, we assume there is only one bubble for each color, so for each message class k, $k = 1, 2, ..., M+1$, $\sum_{q_j \in Q_{VR}} cb_k(q_j) = 1$. With this, BC flow control is given by the following:

$$F_{BC}(q_i, q_j) = true$$
$$if \begin{cases} size(q_j) \le cap(q_j) - 1, \text{ when } q_i \in Q_{VR} \text{ or } q_j \notin Q_{VR} \\ size(q_j) \le cap(q_j) - \sum_{k=1, k \ne color(q_i)}^{M+1} cb_k(q_j) - 1, \\ \quad \text{when } q_i \notin Q_{VR} \text{ and } q_j \in Q_{VR} \end{cases} \tag{3}$$

If a packet at the head of queue q_i with $color(q_i)$ wants to move into the virtual ring (i.e., $q_i \notin Q_{VR}$ and $q_j \in Q_{VR}$), it needs one additional free buffer besides all the bubbles at q_j it cannot occupy (i.e., all the bubbles with a different color, $\sum_{k=1, k \ne color(q_i)}^{M+1} cb_k(q_j)$).

If a packet occupies a colored bubble, it will take the color mark of this bubble with it when it travels inside the ring and leave the color mark to the newly freed buffer when it moves out of the ring. In this way, all the colored bubbles will be kept in the ring.

A routing function based on our proposed bubble coloring scheme is the same as that based on our BR scheme in Eq. (2). That is, packets will be supplied with queues in the virtual ring if their preferred output ports are not available.

THEOREM 2. *An interconnection network I with a routing function R and flow control F_{BC} is free from protocol-induced deadlocks if there exists a virtual ring queue supplied by R_{VR} and controlled by F_{BC} such that, packets from any message class cannot persistently be blocked by packets from other message classes.*

PROOF SKETCH. The BC scheme reserves one bubble (i.e., free buffer) in the ring for each message class so that packets from different message classes have a chance to move into the ring at sometime. According to Lemma 1, due to the existence of the original critical bubble (i.e., black bubble), packets in the ring can make forward movement eventually, which means packets from any message class can never be blocked by packets from other message classes, thus no protocol-induced deadlocks can form. □

3.3 Discussion

3.3.1 Misroutes

Our adaptive routing algorithm may cause misroutes when packets need to move to the escape output ports that do not lie on the minimal path to break the knotted cyclic dependencies. To reduce the chance of misroutes, we set a waiting time threshold T_w. With this, a packet will not move to its escape port unless it cannot be allocated any of the queues in its preferred output ports in the successive T_w cycles. The longer T_w is, the less chance the packet will be misrouted. However, increasing T_w will also increase the blocking time of the packets when cyclic dependencies occur. Empirically, we set the T_w to be 5 cycles .

Note that the escape outports may be on the minimal path for a packet which avoids misroutes. Also, packets that are deflected to the virtual ring and misrouted can return to their minimal paths (i.e., they do not need to stay in the ring) once there are resources available. It is also possible to minimize the misroutes with more advanced virtual ring structures for some particular topology. For example, for a mesh with bidirectional links, two virtual rings with opposite directions can be constructed to provide more escape output ports, so that packets will have higher chance to choose the escape output ports that are on the minimal paths. Different types of virtual rings (e.g., hierarchical rings) at the different virtual networks can also be constructed to further improve the performance. These are topology-specific optimizations which can be explored for the specific network structures.

3.3.2 Livelock

A pathological problem that could arise with misrouting is livelock. When a packet continues to be misrouted, it may not be able to move closer to its destination. To solve this livelock problem, we set a threshold for the maximum number of allowed misroutes (T_m). If a packet has been misrouted for more than T_m times, the packet will be forced to stay in the virtual ring until it reaches its destination.

3.3.3 Starvation

Similar to original bubble flow control [21], our scheme is also susceptible to the starvation problem. Some routers may continually inject packets into the virtual ring, which cause other routers to lose the ability to inject their own packets into the ring. We apply the same starvation prevention design as the original bubble follow control mechanism [21]. If a router detects starvation, it will send a '*starve*' signal to its upstream router along the ring. If a router receives a '*starve*' signal, it will stop injecting packets into the ring and propagate this '*starve*' signal to the next upstream router. Once the packet in this router gets served, the '*starve*' signal will be de-asserted and propagated to the rest of the routers, so that other routers can resume to inject packets into the ring.

4. IMPLEMENTATION

The block diagram in Figure 7 shows the micro-architecture of a typical four-stage virtual cut-through router. Arriving packets will first be written into the input buffer, and then travel through the stages of Virtual channel Allocation (VA), Switch Allocation (SA), Switch Traversal (ST), and finally move out to the next router through Link Traversal (LT) stage. Note that the VA stage is not required unless additional VCs are used for mitigating head-of-line (HOL) blocking. That is, multiple VCs are not required by our proposed schemes to avoid deadlocks but can be used to reduce congestion. Look-ahead routing is adopted to reduce the zero-load latency.

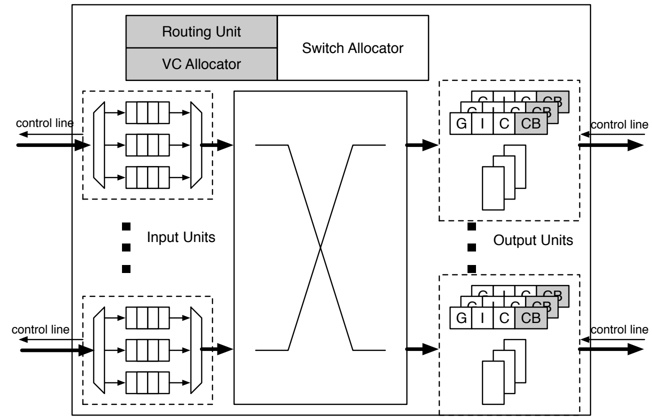


Figure 7: Router micro-architecture

The modifications needed to implement BC scheme is indicated as the shaded areas in Figure 7. The Routing Unit is modified to support BC routing algorithm (i.e., Eq. (2)). To support the rules of BR/BC flow control, Virtual channel Allocation (VA) logic is slightly modified based on Eq. (1) or Eq. (3.2). Some control logic is also added to associate the input and output ports in the virtual ring.

At the output units, the fields of the output virtual channel state are extended. Besides the conventional fields, such as global state (G), input VC (I) and credit count (C) [8], a new field for counting number of bubbles in each color (CB) is added.

To exchange the information about colored bubbles between routers, one control signal for each color is added. To prevent starvation, one '*starve*' signal is also included. Note that these extra control signals are only needed for the input and output ports along the virtual ring.

5. METHODOLOGY

The proposed scheme is evaluated experimentally under a full-system simulation using GEM5 [4], with Garnet [2] for detailed timing of the interconnection network. DSENT [22] configured with a model of an industrial standard 45nm CMOS process is used for router power and area estimation. All the additional hardware described in Section 4 is modeled in the simulator and accounted for in the evaluation. A canonical four-stage Virtual Cut-Through (VCT) switched router with credit-based flow control is assumed. We use a routing algorithm conforming to Eq. (2) where the adaptive routing subfunction provides output ports along all minimal paths to the destination of the packet. Output ports with more credits are prioritized over those with lower credits. Table 1 lists the key parameters of the simulation configuration. We compare all the schemes with mesh networks, on which various deadlock-free schemes have been studied extensively in prior works [11, 10].

Table 1: Full system simulation configuration

Network topology	4x4 or 8x8 Mesh
Router	4-stage VCT, 2GHz
Input buffer	10 flits per VC
Link bandwidth	128 flit/cycle
Cores	Alpha EV5 Core, 2GHz
L1 cache (I & D)	Private, 2-way, 32kB, 2 cycle
L2 cache	Private, 8-way, 512kB each, 6 cycle
Coherence protocol	directory MOESI with 3 message classes
Memory controller	4, located one at each corner
Memory latency	128 cycles

With a typical 128-bit link width, short packets of 16-bits only contain a single flit while long packets carrying 64 byte data plus addition control information have 5 flits. The depth of the buffer for each virtual channel is 10 flits which can hold at most two long packets. A directory MOESI coherence protocol with three message classes [14] are used in the simulation.

We compare the following different deadlock-free schemes with varying number of VCs in our experiments:

1. XY_3VC: XY routing is used to avoid routing-induced deadlocks and 1 VC for each message class is assigned to avoid protocol-induced deadlocks. This scheme requires three VCs in total.

2. XY_adaptive_4VC: Compared to XY_3VC, this scheme adds one additional adaptive virtual channel shared by all message classes and uses XY_3VC as its escape path. It requires four VCs in total.

3. BR_3VC: The Bubble Ring scheme is used to allow fully adaptive routing with routing-induced deadlock-freedom, but it requires one VC for each message class to avoid protocol-induced deadlocks. This scheme requires three VCs in total.

4. BR_adaptive_4VC: Compared to BR_3VC, this scheme adds one additional adaptive virtual channel shared by all message classes and uses BR_3VC as its escape path. It requires four VCs in total.

5. BC_xVC: This group of schemes applies the Bubble Coloring (BC) scheme to avoid both routing- and protocol-induced deadlocks. The number of virtual channels is x (x = 1, 2, 3 or 4).

Both synthetic traffic and multi-threaded applications are used as workloads. For synthetic traffic, the simulator is warmed up for 10,000 cycles and the statistics are collected over another 100,000 cycles. Four traffic patterns are simulated: uniform random (UR), transpose (TP), bit complement (BC) and Hotspot [8]. For Hotspot traffic, the four nodes at the corner of the mesh network have four times higher probability to be chosen as destinations than all other nodes. We assume there are three message classes. The first two message classes send short 1-flit packets, and the third one sends long 5-flit packets. The third message class has twice the injection rate as the first two (i.e., measured in packets/node/cycle).

For real applications, we use the multi-threaded PARSEC benchmark suite [3]. Each core is warmed up for sufficiently long time (with a minimum of 10 million cycles) and then run until the end of the parallel region. All PARSEC benchmarks use the simsmall input set.

6. EVALUATION

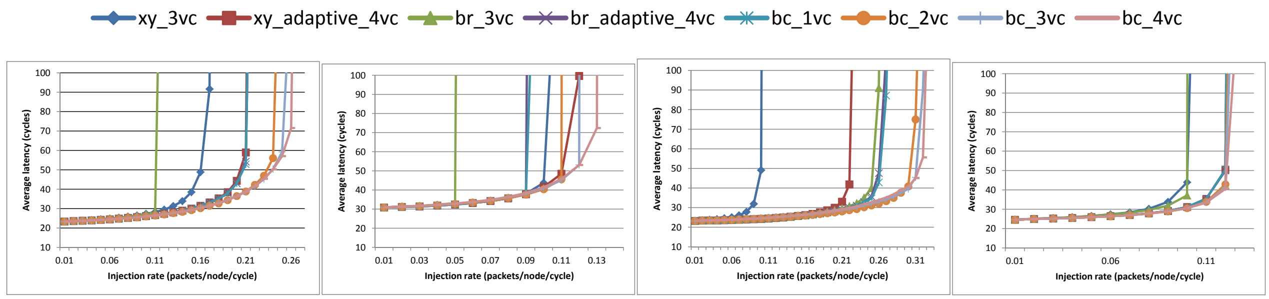
6.1 Synthetic Workload

Figure 8 shows the performance comparison of different deadlock avoidance schemes with varying number of VCs on a 4×4 mesh. As can be seen, XY_3VC scheme has the worst performance under adversarial traffic, i.e., TP, among all the schemes. Under TP traffic, XY routing is not able to balance the traffic between the X dimension and the Y dimension, which leads to the situation where network queues in the X dimension are saturated, while ones in the Y dimension are only lightly loaded. However, by adding one additional adaptive channel over XY_3VC, XY_adaptive_4VC can largely avoid such unbalanced saturation and improve the saturation throughput significantly (over 109%)[2]. This additional virtual channel also helps XY_adaptive_4VC to achieve higher throughput under UR (29.4%), BC (9.1%) and Hotspot (16.7%) due to less head-of-line (HOL) blocking.

Compared to XY_3VC, the proposed BR scheme (BR_3VC) has worse performance under UR and BC traffic patterns. The performance degradation results from the misrouting penalty. When the injection rate increases, packets are more likely to go along the unidirectional virtual ring instead of using preferred outports. When the network is saturated, the throughput of BR_3VC is almost equivalent to the throughput of a unidirectional ring since the majority of packets need to travel along the ring to reach their destinations. However, for TP traffic, the full routing adaptivity that the BR scheme provides enables higher throughput than XY_3VC (145%). The throughput is better than XY_adaptive_4VC (17%) because it allows more VCs to be used adaptively. By adding one adaptive VC on BR_3VC, BR_4VC has its throughput improved even more (83.3%, 18% and 66.7% under UR, Hotspot and BC respectively). This is because that the fourth adaptive channel can be used to relieve the misrouting that would otherwise occur in the first three adaptive channels.

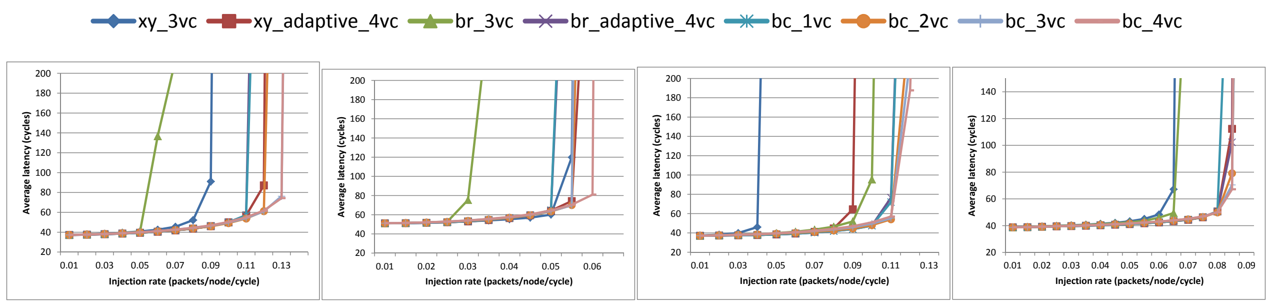
The proposed BC schemes have overall better throughput than XY_3VC and BR_3VC even with only one virtual channel under most of the traffic patterns. With the same number of VCs, BC_3VC outperforms both XY_3VC and BR_3VC under all the traffic patterns. Compared to BR_3VC, the existence of multiple bubbles in the virtual ring can accelerate packet flow in the ring, which increases its throughput. BC_3VC provides 83%, 3.8%, 18.1% and 66.7% throughput improvement over BR_3VC under traffic patterns of UR, TP, Hotpot and BC, respectively. Compared to XY_3VC, BC_3VC has better routing adaptivity, which yields higher throughput (i.e., 29%, 118%, 18% and 10% under UR, TP, Hotspot and BC, respectively). By employing more VCs to reduce HOL blocking, BC schemes can gain higher throughput. Different from XY_3VC and BR_3VC where packets from different message classes can only occupy the assigned virtual channel, the BC scheme allows packets from different message classes to share any virtual channel freely, which allows the traffic to be balanced efficiently among all the virtual channels for the situations where different message classes have different injection rates. This is another source of throughput improvement. From the figure, we can see that no significant throughput improvement can be gained, when the number of VCs in the

[2]Saturation throughput corresponds to the load rate at which the average latency is three times that of the zero-load latency.



(a) uniform random (b) bit complement (c) transpose (d) hotspot

Figure 8: Performance comparison for 4x4 mesh



(a) uniform random (b) bit complement (c) transpose (d) hotspot

Figure 9: Performance comparison for 8x8 mesh

BC schemes increases from 2 to 4, which indicates that the most cost-effective design is the BC scheme with 2 VCs.

In evaluating scalability of the proposed schemes, Figure 9 compares the performances of the same set of schemes mentioned above on an 8x8 mesh. A similar trend of the BC schemes performing better than XY and the BR schemes can be observed for all traffic patterns. Compared to the XY and BR schemes with the same number of VCs, BC_3VC achieves higher throughput over XY_3VC by 40%, 140%, 28% and 9% and over BR_3VC by 133%, 9%, 28.5% and 71.4% under UR, TP, Hotspot and BC, respectively. Thus, as the network scales, the proposed BC scheme continues to outperform other schemes significantly which implies it has good scalability.

In summary, better routing adaptivity provided by the BR and BC schemes enables them to have significant performance improvement under adversarial traffic like TP. The BC scheme is able to use virtual channels more efficiently by freely balancing traffic of different message classes among all VCs. Our BC scheme can outperform other schemes with the same number of VCs and can provide comparable performance even with minimally only one VC.

6.2 PARSEC Benchmark Workloads

6.2.1 Execution time

Figure 10 shows the normalized execution time for six PARSEC workloads. The results are normalized to the execution time of the workloads using the XY_3VC scheme. For real workloads such as PARSEC benchmarks, their load rates are relative low, i.e., no more than 0.1 packet per node per cycle on average, which makes little performance differences between various schemes with the same router frequency since all the schemes have the same zero-load latency. The biggest difference in execution time among all the schemes for a single benchmark (i.e., ferret) is within 5%. On average, even the best scheme (i.e., BC_4VC) has only less than 2% execution time reduction over the worst scheme (i.e., XY_3VC). However, as we discussed in Section 2.1, reducing the number of VCs can reduce the length of the router critical path, which gives the opportunity to increase the router frequency. The BC_1VC_HF scheme, which has the same core frequency (2GHz) as the rest of schemes but increases router frequency slightly from 2Ghz to 2.3Ghz (i.e., 15%), can achieve significant performance improvement, i.e., 12% execution time reduction on average. These results show the ability of our BC scheme with minimal number of VCs to provide an increased opportunity of improving overall system performance.

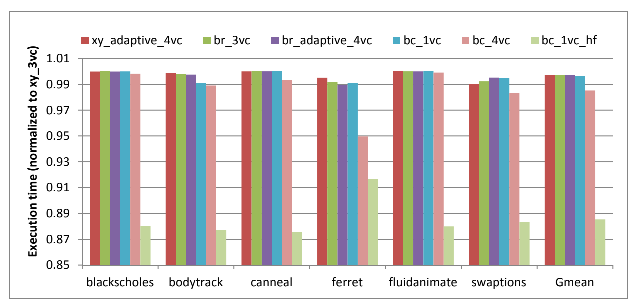


Figure 10: Execution time for PARSEC

6.2.2 Energy

Figure 11 shows the normalized router energy consumption for six PARSEC benchmarks with different deadlock-free schemes. The results are normalized to XY_3VC. From the figure, we can see that the total router energy is largely affected by the number of virtual channels. The more virtual channels are needed, the more buffers that will be used, which causes more static power consumption. Compared to the schemes with 4VCs (e.g., XY_adaptive_4VC), BC_1VC

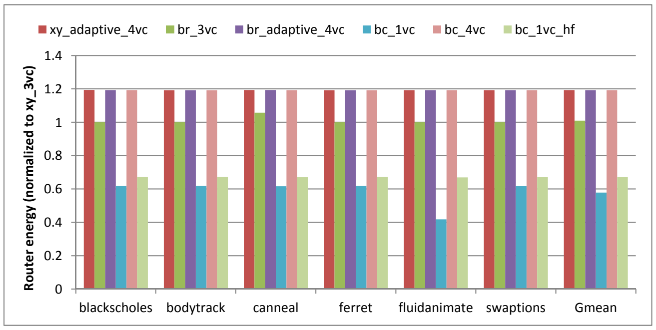


Figure 11: Router energy consumption

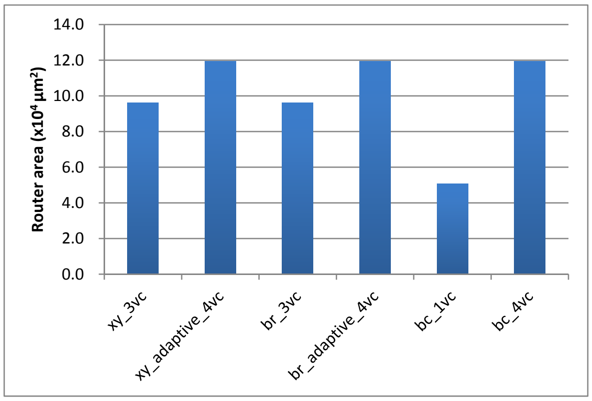


Figure 12: Comparison of per router area

can reduce router energy by up to 51.2%. Under the relatively low traffic load rate, all the schemes will have similar total hop counts since packets in all schemes will likely travel the minimal paths. Therefore, dynamic router energy among different schemes with the same router frequency has no significant differences since the total amount of router activities (e.g., buffer writes, VA/SA arbitrations) is largely dependent on the average packet hop count (i.e., number of packets being forwarded through a router on average). BC_1VC_HF has a higher router energy consumption than BC_1V since the dynamic power has been increased when router frequency becomes higher. In sum, on the one hand, our BC scheme can save a large amount of router energy (shown in Figure 11) without significantly degrading the system performance (shown in Figure 10). On the other hand, our BC scheme provides the opportunity to gain significant performance improvement by increasing router frequency with comparable energy consumption.

6.2.3 *Area*

Figure 12 shows the comparison of per router area for different schemes. The router area consumption can be divided into buffer or non-buffer (e.g., control logic). Although different schemes have relatively the same non-buffer area, the different number of VCs among different schemes largely affects router buffer area. As expected, the BC scheme with the minimal one virtual channel (i.e., BC_1VC) consumes the least router area. Compared to schemes with 4VCs (e.g., XY_adaptive_4VC), BC_1VC can save router area by up to 58.3%. Considering the fact that there is little performance degradation with reduced number of VCs (shown in Figure 10), it is clear that BC_1VC is most area-efficient among all the schemes.

In summary, by comparing performance, router energy and router area consumption, the simulation results for real multithreaded applications show that the proposed Bubble Coloring scheme with minimal VC requirement can achieve significant improvements in terms of router energy/area efficiency. Compared to conventional XY routing schemes, BC schemes can achieve comparable performance with significant router energy/area savings. The BC scheme also provides the opportunities to significantly improve the system performance by increasing router frequency with no more energy consumption than other schemes.

7. RELATED WORK

Several foundational theories have been established in [11, 12], on which many deadlock-free routing algorithms are based. To improve the effectiveness of routing, previous research has also investigated various selection metrics that utilize local and/or regional information [16]. However, as all these routing algorithms are based on Duato's Protocol [11], at least two virtual channels are needed to implement fully adaptive routing.

The turn model and its extensions [7, 13, 15] avoid routing-induced deadlocks using minimally one VC by disallowing certain paths between the source and destination. Therefore, these schemes can only provide partially adaptive routing as opposed to the fully adaptive routing supported in our proposed approach that can better balance traffic load and reduce contention latency.

While the schemes above can avoid routing-induced deadlocks, they all rely on separate virtual networks to avoid protocol-induced deadlocks, which have considerable negative impacts on area, power and frequency as described in Section 2.1. A formal model of protocol-induced deadlocks is derived in [23], and a recovery-based approach is proposed in [17] to achieve deadlock-freedom; whereas our approach is avoidance-based. Detailed comparison and discussion between the two types of approaches are provided in [20].

Another way of avoiding routing-induced deadlocks is to use bubble flow control mechanisms. Bubble flow control (BFC) was proposed in [21] and adopted in the IBM Blue Gene/L [1]. Recently, more efficient implementations of BFC have been proposed to increase buffer resource utilization [6, 5]. As briefly mentioned before, these works are different from our scheme in two distinct ways. First, these works are applicable only to torus-like topologies whereas our scheme can be applied to a general set of topologies including both regular networks, such as meshes, and irregular networks. Second, existing BFC and its implementations still require separate VNs to avoid protocol-induced deadlocks while our scheme reduces the number of virtual networks to the theoretical minimal value of one.

8. CONCLUSION AND FUTURE WORK

The common way to avoid routing- and protocol-induced deadlocks without losing routing freedom is to use additional virtual channels. However, implementing a large number of VCs in routers has considerable negative impacts on router area, power and frequency. The contribution of this paper is to enable fully adaptive routing on any topology with minimally one virtual channel while still avoiding both routing- and protocol-induced deadlocks. The proposed Bubble Ring flow control mechanism is applied to avoid routing-induced deadlocks through a virtual ring that spans all nodes in the network by maintaining a free packet-sized buffer inside the ring. Routing algorithms, with full routing adaptivity can leverage this deadlock-free virtual ring as its escape path to avoid routing-induced deadlocks. Based on the BR scheme, the Bubble Coloring scheme is proposed to further avoid protocol-induced deadlocks by maintaining one additional free buffer (i.e., colored bubble) in the virtual ring for each message class so that packets from each message class can always reach their destinations through the virtual ring no matter whether other message classes blocked or not. Simulation results for both synthetic traffic and real multi-threaded applications show that, compared to a conventional deadlock-free scheme with 4VCs (i.e, XY_adaptive_4VC), our BC scheme with the minimal 1VC (BC_1VC) can reduce router energy and area by up to 51.2% and 58.3%, respectively, and have comparable performance at the same time. As the proposed BC scheme does not require multiple virtual channels, it also reduces the complexity of the router arbitration logic which brings the opportunity to increase router frequency and further improve system performance.

In the future, the proposed BC scheme can be further extended in promising directions. If a particular topology is given, more virtual rings can be constructed. For example, for meshes with bidirectional links, two unidirectional rings with opposite directions can be constructed, which will reduce the chances of misroutes. The number of misroutes could be further reduced by constructing more advanced rings (e.g., hierarchical rings) on separate virtual channels so that more escape ports along the minimal paths can be found. Our BC scheme can also be used to control network congestion status by adjusting the number of colored bubbles in the virtual rings. The increasing number of bubbles reserved in the virtual ring can accelerate the packet flow along the ring and relieve the congestion status. Considering that the ring is "virtual" and can be easily reconfigured, a class of intelligent congestion control mechanisms with the adaptivity of both virtual ring structures and the number of bubbles in the rings can be derived from our BC scheme in future research.

9. ACKNOWLEDGMENTS

We sincerely thank José Duato for his helpful comments and suggestions. This research was supported, in part, by the National Science Foundation (NSF), grant CCF-0946388.

10. REFERENCES

[1] N. Adiga, G. Almasi, G. Almasi, Y. Aridor, R. Barik, et al. An overview of the bluegene/l supercomputer. In *Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, pages 1–22. IEEE Computer Society Press, 2002.

[2] N. Agarwal, T. Krishna, L.-S. Peh, and N. Jha. Garnet: A detailed on-chip network model inside a full-system simulator. In *Proceedings of the 2009 IEEE International Symposium on Performance Analysis of Systems and Software*, ISPASS '09, pages 33–42, 2009.

[3] C. Bienia and K. Li. Parsec 2.0: A new benchmark suite for chip-multiprocessors. In *Proceedings of the 5th Annual Workshop on Modeling, Benchmarking and Simulation*, June 2009.

[4] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, et al. The gem5 simulator. *ACM SIGARCH Computer Architecture News*, 39(2):1–7, Aug. 2011.

[5] L. Chen and T. M. Pinkston. Worm-bubble flow control. In *Proceedings of the 19th International Symposium on High Performance Computer Architecture*, HPCA '13, pages 366–377, 2013.

[6] L. Chen, R. Wang, and T. M. Pinkston. Critical bubble scheme: An efficient implementation of globally aware network flow control. In *Proceedings of the 25th IEEE International Parallel Distributed Processing Symposium*, IPDPS '11, pages 592–603, 2011.

[7] G.-M. Chiu. The odd-even turn model for adaptive routing. *IEEE Transactions on Parallel and Distributed Systems*, 11(7):729–738, 2000.

[8] W. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[9] W. J. Dally. Virtual-channel flow control. *IEEE Transactions on Parallel and Distributed Systems*, 3(2):194–205, 1992.

[10] W. J. Dally and H. Aoki. Deadlock-free adaptive routing in multicomputer networks using virtual channels. *IEEE Transactions on Parallel and Distributed Systems*, 4(4):466–475, 1993.

[11] J. Duato. A necessary and sufficient condition for deadlock-free routing in cut-through and store-and-forward networks. *IEEE Transactions on Parallel and Distributed Systems*, 7(8):841–854, 1996.

[12] J. Duato and T. Pinkston. A general theory for deadlock-free adaptive routing using a mixed set of resources. *IEEE Transactions on Parallel and Distributed Systems*, 12(12):1219–1235, 2001.

[13] B. Fu, Y. Han, J. Ma, H. Li, and X. Li. An abacus turn model for time/space-efficient reconfigurable routing. In *Proceedings of the 38th International Symposium on Computer architecture*, ISCA '11, pages 259–270, New York, NY, USA, 2011. ACM.

[14] GEM5. Cache coherence protocols, http://gem5.org/cache_coherence_protocols#moesi_cmp_directory.

[15] C. J. Glass and L. M. Ni. The turn model for adaptive routing. In *Proceedings of the 19th International Symposium on Computer Architecture*, pages 278–287, 1992.

[16] P. Gratz, B. Grot, and S. Keckler. Regional congestion awareness for load balance in networks-on-chip. In *Proceedings of the 14th International Symposium on High Performance Computer Architecture*, pages 203–214, 2008.

[17] Y. Ho Song and T. M. Pinkston. A progressive approach to handling message-dependent deadlock in parallel computer systems. *IEEE Transactions on Parallel and Distributed Systems*, 14(3):259–275, Mar. 2003.

[18] S. S. Mukherjee, P. Bannon, S. Lang, A. Spink, and D. Webb. The alpha 21364 network architecture. In *Proceedings of the Ninth Symposium on High Performance Interconnects*, HOTI '01, pages 113–117, Washington, DC, USA, 2001. IEEE Computer Society.

[19] L.-S. Peh and W. J. Dally. A delay model and speculative architecture for pipelined routers. In *Proceedings of the 7th International Symposium on High-Performance Computer Architecture*, HPCA '01, pages 255–266, Washington, DC, USA, 2001. IEEE Computer Society.

[20] T. M. Pinkston and S. Warnakulasuriya. On deadlocks in interconnection networks. In *Proceedings of the 24th International Symposium on Computer Architecture*, ISCA '97, pages 38–49, New York, NY, USA, 1997. ACM.

[21] V. Puente, C. Izu, R. Beivide, J. Gregorio, F. Vallejo, and J. Prellezo. The adaptive bubble router. *Journal of Parallel and Distributed Computing*, 61(9):1180–1208, 2001.

[22] C. Sun, C.-H. O. Chen, G. Kurian, L. Wei, J. Miller, et al. Dsent - a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling. In *Proceedings of the 2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip*, NOCS '12, pages 201–210, Washington, DC, USA, 2012. IEEE Computer Society.

[23] S. Wamakulasuriya and T. Pinkston. A formal model of message blocking and deadlock resolution in interconnection networks. *IEEE Transactions on Parallel and Distributed Systems*, 11(3):212–229, 2000.

[24] Y. Xu, B. Zhao, Y. Zhang, and J. Yang. Simple virtual channel allocation for high throughput and high frequency on-chip routers. In *Proceedings of the 16th International Symposium on High Performance Computer Architecture*, HPCA '10, pages 1–11, 2010.