Packet Injection Mechanisms and their Impact on Network Throughput

C. Izu[1], J. Miguel-Alonso[2], J.A. Gregorio[3]

[1]Department of Computer Science, University of Adelaide, SA 5005 Australia
[2]Department of Computer Architecture and Technology, The University of Basque Country, P.O. Box 649, 20080 San Sebastian, Spain
[3]Computer Architecture Research Group, Universidad de Cantabria, 39005 Santander, Spain

Technical Report EHU-KAT-IK-01-05

This work has been done under support of the Spanish Ministerio de Educación y Ciencia, under grants TIN2004-07440-C02-01 and TIN2004-07440-C02-02, and also by the Diputación Foral de Gipuzkoa under grant OF 846/2004.

Packet Injection Mechanisms and their Impact on Network Throughput

Abstract

This paper analyzes the impact that the injection interface has on maximum sustained throughput in an adaptive cut-through torus network. Recent work has pointed out that head-of-line blocking (HOLB) at the network interface, due to the use of single FIFO injection queues, may prevent a network from sustaining its peak throughput at heavy loads. Additionally, the most recent commercial parallel systems use multiple injection queues, but little is known about the rationale behind these design decisions, or their implementation details.

This paper will model and thoroughly analyze the effect on performance of the following factors: the number of injection queues (from 1 to 4), the allocation of packets to queues (testing different selection policies, with or without pre-routing at the interface) and the mapping of queues to the available number of injection channels (virtual injection channels vs. physical injection channels). Network evaluations for medium to large size 2D tori will show that designs with multiple FIFO injection queues do not improve performance under uniform traffic, when compared with the simple, single-FIFO interface. On the contrary, for some injection policies, throughput loss increases for loads beyond the saturation point. At the hearth of this behavior is network congestion: more injection ports results in more pressure from the injection interface to acquire the scarce network resources of an already clogged system.

We conclude that new, restrictive injection policies are required that, while being starvation-free, prevent processing nodes from overflowing routers with new packets for loads beyond the network's saturation point. Interestingly, for small networks, a single injection FIFO queue, with the HOLB it entails, may actually be a good design choice as it indirectly provides the much-needed injection control. For networks with thousands of nodes, as those being implemented in current massively parallel processors, this basic form of congestion control is not enough. Regardless of the number of injectors, an injection-throttling mechanism is essential to reduce throughput losses and maintain, or even increase, maximum sustained throughput.

1. Introduction

The interconnection network is a key element of a parallel computer system, which provides the necessary high-bandwidth and low-latency communication channels. Networks of this kind are now found in many systems to provide not only inter-processor communication or processor-memory interconnect, but also input-output and storage switches, as well as replacing dedicated wiring.

An ideal network should reach full link utilization and be able to sustain peak throughput under heavy loads. However, most networks reach lower link utilizations due to network contention. There is a large body of research in 2D torus networks, which has focus on reducing contention by increasing the number of requests made by incoming packets: adding virtual channels (VCs) [10], providing adaptive routing [8] or both [12]. The router architecture may be another source of contention by causing head of line blocking or limiting the crossbar complexity [13]. As silicon area is less of a premium nowadays, a simple way to reduce contention is to implement virtual cut-through (VCT) flow control with enough buffer capacity so that it is not overflowed at medium loads. For example, the Alpha 21364's router, which uses VCT, provides a total buffer space for 316 packets [16].

Most router design parameters such as number of virtual channels, buffer capacity and routing functions have been thoroughly evaluated, and their impact on network performance is reasonably well understood. For example, it is known that a small number of virtual channels, in the range 2 to 4, increases throughput by reducing head-of-line blocking (HOLB), regardless of the routing strategy applied. Although the organization of the injection queues at the network interface also has a significant role in the way traffic is accepted, it has received less attention in the literature. Network performance is usually evaluated using register-level simulators with synthetic or real traffic loads. These simulators model a simple network interface with a single FIFO queue where new packets wait to be transferred into the router's injection port [7,8,15,16]. For large k-ary n-cube networks, HOLB at the injection queue has been identified as the reason why some network resources were under-utilized [15]. Under uniform loads, such networks exhibit a clear load unbalance between their +X and –X channels. This is explained by the fact that the first set of channels to saturate, lets say +X, will fill its buffers and stop accepting new packets, while the other direction (–X), will be prevented from receiving new packets until the blocked packets at the head of the queue have been injected into +X. Such blockage at the injection queue is reflected in its lower buffer occupation. That study suggested that if we eliminated the HOLB at injection, by providing separate injection queues in each direction, both set of channels, +X and –X, would be able to reach their peak, increasing the overall network performance.

This paper explores the impact that the injection interface, the number of injection ports in particular, has on maximum sustained throughput. By adding multiple ports with their associated injection queues, we may reduce or eliminate that head-of-line blocking. The majority of implemented networks, ranging from the Torus Routing Chip [11] to the Cray T3E Network [18] offered a single full duplex connection between the router and its processing element, with a single injection port. On the other hand, more recent networks have multiple injection ports: the Alpha 21364 router [16] has 4 local ports (from the two on-chip memory controllers, the I/O external chip and the on-chip second level cache) and the BlueGene/L network incorporates 8 injection ports per router (one per dimension plus two high-priority), but in [1,6] we cannot find a in depth description of how the injection process is managed.

To the best of our knowledge, there are no theoretical studies about the effect that adding multiple injection ports has on the performance of fully-adaptive VCT networks. Thus, one of the contributions of this work is to close the gap between the massively parallel systems currently being built (using torus networks), and the knowledge of adaptive *k*-ary *n*-cube networks in the literature. To achieve this, we have modeled a range of injection interfaces and then thoroughly analyzed the impact that such designs have on network performance. Our work will prove that a single queue, with the HOLB it entails, does not limit network throughput under uniform traffic. On the contrary, it prevents the network performance from further degrading under heavy loads by providing partial congestion control in one or more directions. This study also highlights the need to propose fair and simple injection throttling policies to increase and sustain peak performance.

The paper is organized as follows: Section 2 describes different injection interface designs. Section 3 describes the evaluation methodology employed. Section 4 presents simulation results for an adaptive bubble router (ABR) for a range of interface designs on them. Section 5 explores the impact of throttled injection and introduces related work and finally, Section 6 summarizes the findings of this work.

2. Injection interface design

The network interface provides queuing both before and after the interconnection network. In a standard injection interface, the packet at the head of the queue will advance to the router's injection port and request one or more output links. Normally, new and transit packets are governed by the same rules: they will advance provided that the flow control allows it and a connection can be established to the selected output port. Figure 1(a) shows a generic router with a single injection port. We can see in this figure that while the –X output is free, the packet with header (Δx, Δx) = (-3, -1) is blocked due to the contention in the opposite direction, that prevents the first packet from being injected.

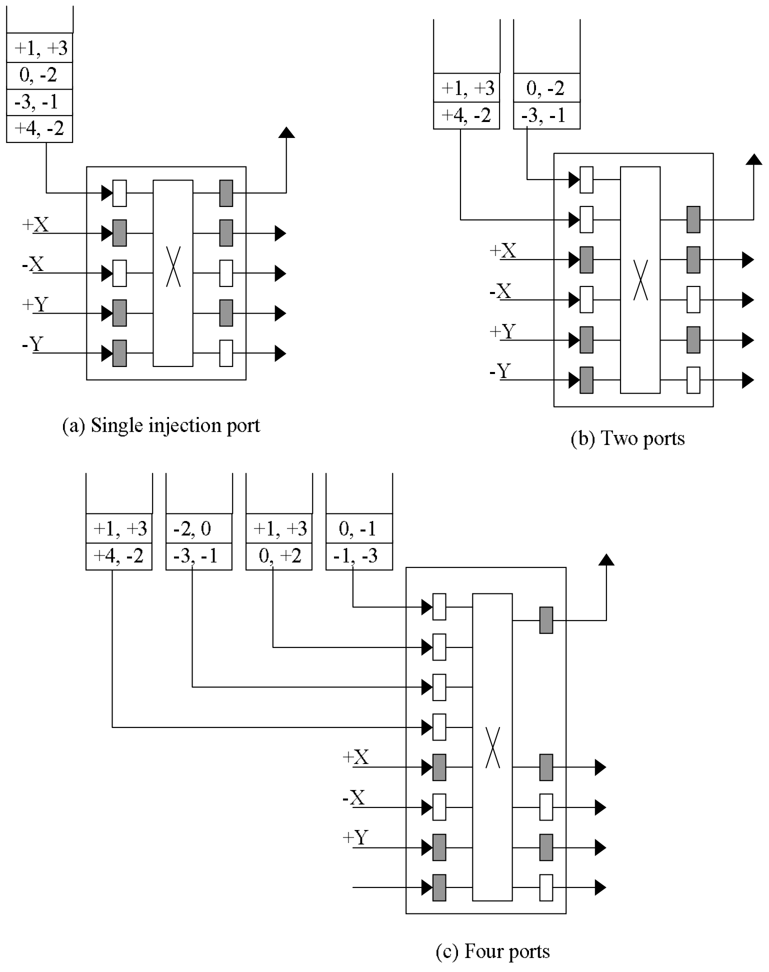


Figure 1. Different organizations for the injection ports. Each injection port has its corresponding queue.

Thus, we need to consider alternative injection interface designs that reduce or eliminate this head-of-line blocking. Figure 1(b) shows an injection interface with two independent queues. Packets at the injection interface can be pre-routed in order to select their injection queue, so that packets travelling in opposite direction, +X and –X, are allocated to different queues. However, this alternative does not eliminate the HOLB for the fraction of packets travelling only in the Y direction or using adaptive paths, when the two packet headers are blocked, waiting on their X channels. We need 4 queues and their associated ports (or *2d* queues for a *d*-dimensional router) to eliminate *all* possible cases of head of line blocking, as shown in Figure 1(c).

When using multiple injection queues, we need to define a policy to select, for each packet, the injection queue in which the packet will be stored. As this is a pre-routing decision, it can be either static or dynamic. There is a large design space for this kind of port selection policies, from which we have evaluated the following ones:

- *Shortest (sh)*: the new packet is allocated to the shortest queue. This is the simplest dynamic policy, and it reduces the impact of HOLB at injection in a similar way that virtual channels do inside the router.

- *Shortest with pre-routing (shp)*: we assign a queue to each output link, as shown in Figure 1(c). Each packet can now be assigned to as many queues as directions it needs to travel. For example, the packet with header (-3, +1) can go to the –X queue or the +Y queue. From this set, the one with lower population is selected.
- *Longest path first (lpath)*: each packet is allocated to the queue associated to the direction having the longest coordinate to travel. In the example given above, it will be place on the –X queue, regardless of the queue's population. Thus, this is a static decision.

In any case, when the queue of the selected port is full, the generation of packets stalls until the packet is successfully queued. This means that packets are not dropped and, therefore, when the network is saturated, the load actually generated may be smaller that the target load to apply.

Finally, in Figure 1 a physical channel connects each injection queue to its port. This means that the injection bandwidth grows with the number of ports. When the injection bandwidth is a limiting factor, it is possible to use virtual channels instead. This choice depends on technological and implementation factors, such as if the router is part of an integrated chip or a single chip, what are the package constrains, etc.

To complete this analysis we will evaluate both cases: a physical channel per injection port versus a single physical channel shared by the 4 virtual injection ports. We should note that as the network size increases, the injection rate (8/*k* phits/cycle/node, being *k* the number of nodes per dimension) set by the bisection limit decreases, so that more injection bandwidth will have little impact in the overall network performance for most point-to-point traffic patterns. On the other hand, multiple injection channels could expedite local and collective communication by allowing packets to be simultaneously sent in more than one direction.

3. Evaluation methodology

In order to evaluate the impact that the injection interface design has in a torus network, we have used a cycle-driven register level simulator that models an adaptive bubble router [17] with 3 virtual channels per physical link, and implemented the single-port and 4-port injection interfaces as described in Section 2.

The router architecture used in this work is modeled as shown in Figure 2. This is a fully adaptive virtual cut-through router, and like many of its counterparts, it provides fully adaptive routing by applying Duato's approach [14]: a subset of the total virtual channels is configured as a safe virtual network, also called the escape network, in which packet deadlock never occurs. The remaining VCs are configured as a fully adaptive virtual network. Both the Alpha 21364 and the BlueGene/L torus networks use this strategy, adding more virtual channels to a dimension-order routing (DOR) network. They differ in their choice of the deadlock avoidance mechanism. The Alpha network breaks cycles within a dimensional ring by using two virtual channels as in the Torus Routing Chip [11], while the BlueGene/L relies on Bubble Flow Control (BFC) [9], which prevents the node from injecting a packet if such action exhausts the local buffer capacity in the direction/way (that is, in the ring) it is advancing.

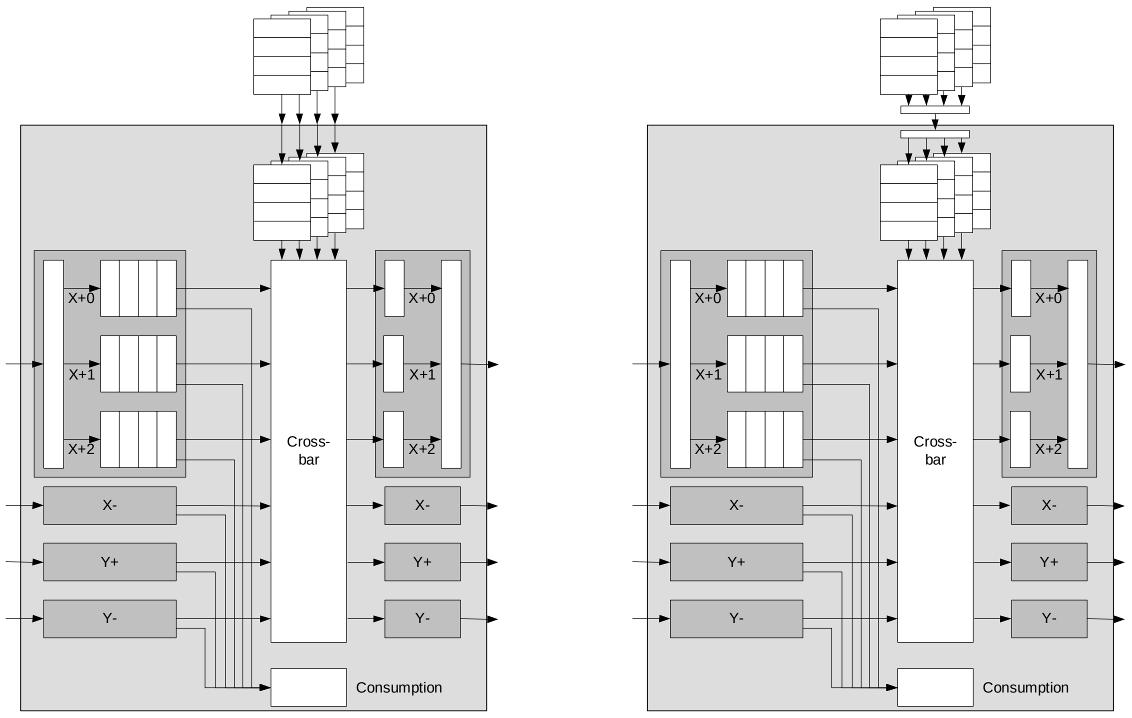


Figure 2. Two variants of the adaptive VCT router architecture, both with 4 injection ports connected to the crossbar. Left: each injection port has a dedicated physical link. Right: injection ports are virtual, as they share a single physical link.

Each physical channel has three input buffers, corresponding to one escape virtual channel and two fully adaptive VCs. Packets move under two different policies. In the adaptive virtual network the injection and forwarding of packets select minimal adaptive paths under virtual cut-through flow control. In the escape virtual network, packets follow static paths as per DOR routing, with the injection regulated by bubble flow control. Packets turning from an X-ring to a Y-ring inside the escape virtual network are considered as new injections into the receiving Y-ring. Packets in transit inside a safe ring are regulated by VCT.

Packets can move freely from the safe to the adaptive network, and allocation policies do favor the use of adaptive VCs in the same ring; if none is available, they may use an adaptive VC in another profitable dimension. The escape network is used as the last resource and the change of packets from the adaptive to the safe network is also regulated by BFC. When several input channels (including injection channels) request the same virtual output, a random arbitration policy is used.

The number of injection ports, its selection policy and injection bandwidth are as described in Section 2. When a packet is generated, if the selected injection buffers are full, packet generation is suspended until the new packet is accepted by the injection interface. As we can see in Figure 2, the consumption interface can receive several packets (from different input ports) simultaneously. This models the best case in with consumption bandwidth is never a bottleneck.

Other simulation parameters are as follows:

1. Two network sizes are considered: **256 nodes** (16x16), which is a medium size, commonly used in network evaluation, and **1024 nodes** (32x32) to represent larger parallel systems.
2. Packets are of fixed size: 16 phits. A phit is the number of bits that are conveyed in parallel through a physical link.
3. The queue capacity of each input port is 8 packets, or 128 phits. Injection channels at the interface side have an additional buffer of 128 phits.
4. Traffic generation follows a Bernoulli temporal distribution, with a parameter that depends on the applied load (expressed in phits/node/cycle). Spatial traffic pattern may follow a random, **uniform** distribution, or a **hot-region** distribution. The latter is defined as follows: "*In this pattern, the destinations of 25% of the packets are chosen randomly within a small "hot" contiguous sub-mesh region consisting of 12.5% of the machine. The remaining 75% of the packets chose their destinations uniformly over the entire machine.*" [6]

Maximum throughput under uniform traffic is limited by the network bisection bandwidth [13] and many studies normalized their loads to this limit, which represent full link utilization under uniform traffic load. For the 16x16 and 32x32 torus networks, this maximum is 0.5 and 0.25 phits/node/cycle respectively. More recently, researchers have started to report network performance for loads above this limit [21]. This reflects the way in which some parallel applications generate network loads. On intensive communication phases, provided the message overheads are low, nodes may send tens of packets in a row, well above the theoretical limit. In such phases, even applications with lower global communication demands may always have a packet to be injected, once the network has reach its saturation point and the injection queues overflow. As we are interested in observing the behavior of a saturated network, in most of our experiments we apply a load of 1.0 phits/node/cycle, which (for uniform traffic) is 2 or 4 times the bisection limit. The use of synthetic loads allows us to cover a wider simulation space, without losing sight of real application loads, which have been shown to match their network response in terms of asymmetric use of network resources at high loads [15].

Although the simulation runs provide a wealth of information about network performance, we will focus on the following subset:

- Total accepted versus offered load (in phits/node/cycle) in order to identify peak and maximum sustained throughput. In all other cases, unless otherwise stated, results correspond to offered loads of 1 phit/node/cycle.
- Evolution of buffer utilization for transit ports, and average of this utilization.
- Average utilization of each virtual channel.

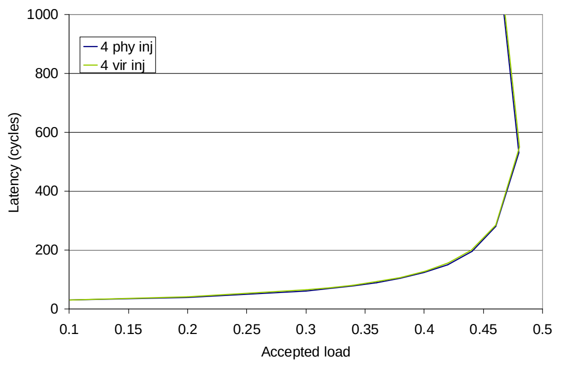
4. Network Evaluation

This section presents the results obtained from the simulation of 2D torus networks with different designs for the injection subsystem. Firstly, we will evaluate the influence of the injection bandwidth on network response. Secondly, we will evaluate the impact than the number of injection ports in combination with the allocation of packets to queues has on the performance of a 256-node 2D torus network. Lastly, we will analyze its impact when the network size increases to 1024 nodes.

4.1 Injection bandwidth

This section analyzes the impact that the injection bandwidth has on network performance. Adding more injection ports results in more bandwidth allocated to the node interface. As the interconnection links of any router are a limited resource, the bandwidth allocated to the transit network links could be constrained by this decision. To avoid this, the injection interface can use virtual channels to connect its multiple queues to their associated ports. Note that, as shown in Figure 2 (right), the internal router architecture remains unaltered, as it emulates the 4 injection ports by means of 4 virtual injection channels; the only difference is that the processing node can send a packet from only one injection queue to the router core at a time.

Figure 3 shows the network latency versus accepted load for a 256-node network with 4 injection queues, varying the injection bandwidth from a single injection channel, shared by the 4 queues (*4_vir_inj*), to 4 dedicated injection channels (*4_phy_inj*).



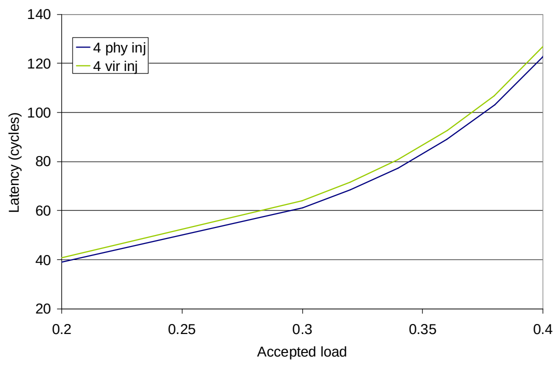


Figure 3. Left: effect of the injection bandwidth on average packet latency, at several levels of accepted load, in a 16x16 torus network with 4 injection ports. The right-side plot is a close-up for medium loads (below saturation).

The impact of the injection bandwidth on network performance is minimal. Sharing the injection channels only increases latency at medium loads by less than 3%. Differences in peak and sustained throughput for both configurations (not shown) are negligible. This confirms our expectations for networks with radix larger than 8, in which the maximum injection rate is less than 1 phit/cycle/node. Networks with larger radix, such as the 32x32 torus, have a lower maximum injection rate per node, so that the impact of the injection bandwidth will be negligible for most point-to-point traffic patterns.

In short, the injection pressure depends mainly on the number of new packets inside the router that can simultaneously request an output link, and not on the physical bandwidth of the injection port, provided this is large enough to cope with a standard injection rate. For the rest of this paper we will not discriminate between physical and virtual injection channels/ports.

4.2 256-node adaptive torus network

This section evaluates the impact that the organization of the injection interface and its queues has on the performance of a 16x16 torus. The standard injection policy applies to each injection port: BFC restriction to inject into the escape network and VCT to inject into the adaptive network.

Figure 4 shows the throughput versus offered load for the three different injection organizations under study. The traffic fluctuations at high loads will cause one arbitrary network direction to saturate first. In this set of experiments, the –X channels were first to saturate. When using a single injection queue, the HOLB caused by a packet waiting to access the –X output link prevents packets traveling in other directions to be injected, so that the rest of the network does not accept loads above their saturation point. This limitation indirectly helps to maintain the actual network load at a manageable level, and sustain this peak throughput, regardless of the offered load. When having multiple queues, the response depends on how these queues are managed. Static port selection policies result in the injection queue of the saturated direction overflowing and causing packet generation to stall until the network is able to accept that queue's first packet. Thus, the HOLB effect is delayed but not eliminated. Dynamic selection increases saturation by spreading the packets over multiple injection queues: when the -X injection queue overflows, the node continues generating packets until it fills up the +Y and –Y queues. As the injection pressure is kept on most network directions, it is not surprising to see that dynamic port selection policies (*4_inj_sh* and *4_inj_shp*) lead to lower sustained throughput. The 4-queue design with the static policy (*4_inj_lpath*) matches, but does not improve, the throughput obtained by its single queue counterpart (*1_inj*).

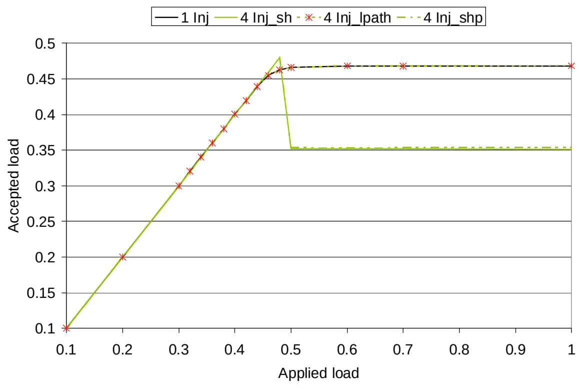


Figure 4. Accepted versus offered load for a 16x16 adaptive bubble torus networks with 1 (black line) or 4 injection queues and different injection port selection policies.

To better explain the network behavior at heavy loads, we will examine the buffer occupation and the utilization of each virtual channel. Figure 5 shows these values at maximum load.

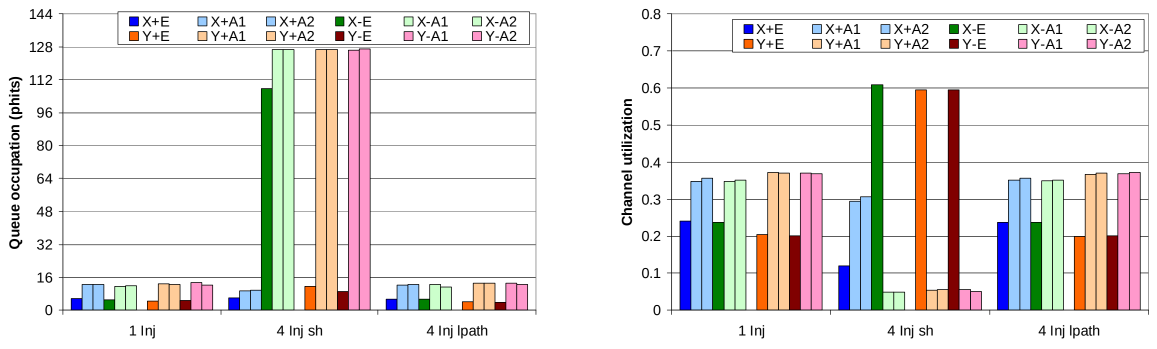
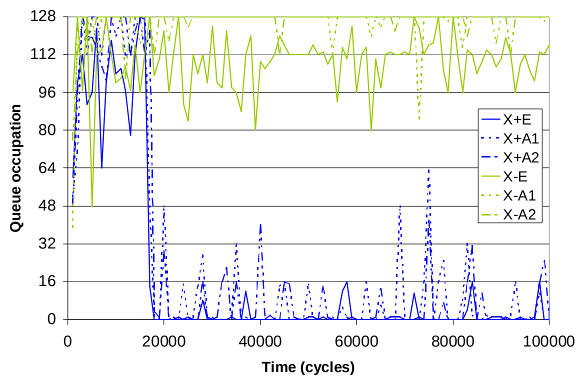


Figure 5. Transit queue occupation (left) and channel utilization (right) in a 16x16 torus at 1 phit/node/cycle applied load, for 1-4 injection ports, and for sh and lpath injection port selection policies. Each column represents a virtual (transit) channel.

When the adaptive networks saturates, many packets are forced to use their escape paths. This is reflected in virtual channel utilization, which is low for adaptive channels but high for its escape counterparts. The channel utilization in the +X direction differs because most packets use adaptive VCs, which indicates that this direction has not reach saturation. Average buffer occupation is another parameter that reflects network saturation and its impact on network throughput. We can see that by using 4 queues and dynamic port selection (*4_Inj_sh*), we have significantly increased the population in the adaptive networks, with the average occupation being very close to the queue's 128-phit capacity On the other hand, the other injection policies maintain a lower buffer usage and are able to sustain their peak throughput for loads beyond saturation.



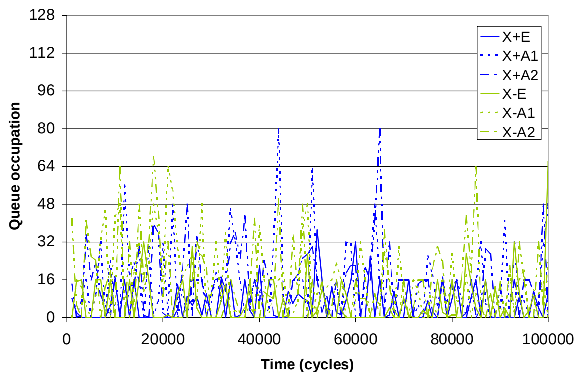


Figure 6. Transit queue occupation on the X-axis over time in a 16x16 torus with 4 injection ports, for shortest (left) and lpath (right) injection port selection policies.

Let us explain these two different scenarios by looking the buffer occupation in the first dimension over time for both shortest and lpath policies, as shown in Figure 6. We have included the warm-up period because it illustrates, for the shortest policy, the time taken to reach saturation on the –X axis, and how, once this occurs, the buffer occupation in the opposite direction falls to a lower value, reflecting the asymmetry described in [15]. It also indicates that the network could cope with traffic bursts, provided they don't last long (not more that 10,000 or 15,000 cycles).

As this policy eliminates the HOLB at injection, there is constant pressure to inject packets in all directions but +X. The adaptive network gradually fills up its transit queues and injection pressure is then transferred to the escape network, in the –X direction, whose queues fill up as well. Remember that virtual cut-through flow control requires 16 free phits for a packet to advance to its next input queue. Thus, it is clear why the adaptive channel usage (except for +X) drops dramatically when the processing node floods the router with new packets, leading to the 20% throughput loss shown in Figure 4.

In the other alternatives, the network appears to be nearly *empty*, with an average occupation of around one packet per virtual channel, or around 3 packets per physical channel. The remaining buffer space is used to cope with fluctuations over time as shown in Figure 6 for the *4_inj_lpath* strategy. HOLB at injection prevents congestion in the 256-node network. This explains why many studies of adaptive torus networks did not exhibit performance degradation at high loads. Adding more injection ports requires an adequate selection policy so that the adaptive network is not flooded with new requests beyond its saturation point.

4.3 1024-node adaptive torus network

This section evaluates the impact that the injection interface has on the performance of a 32x32 torus. Although is common for large direct networks to arrange nodes in three dimensions, our experiments reflects realistic designs such as the BlueGene/L, whose rings can be even larger that the ones we tested here. Figure 7 shows the throughput versus offered load for this network with different injection organizations.

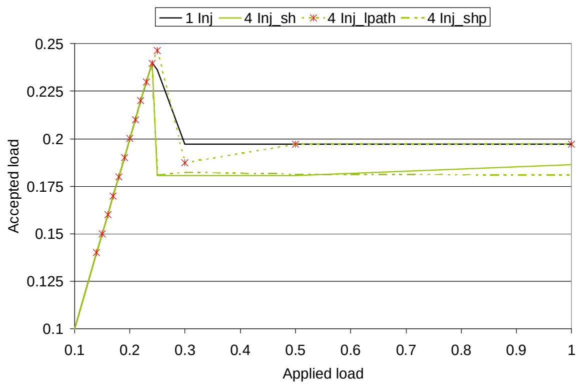
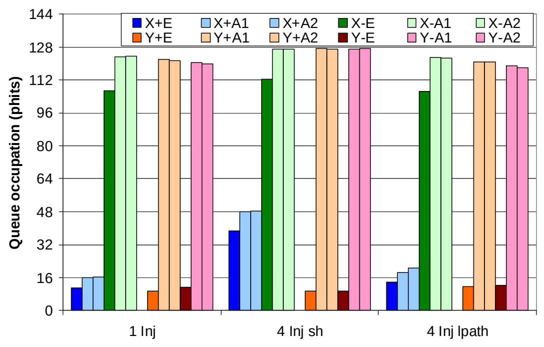


Figure 7. Accepted versus offered load for a 32x32 adaptive bubble torus with 1 (black line) or 4 injection queues and different injection port selection policies.



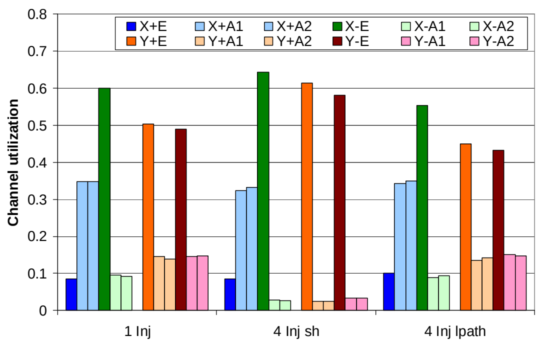


Figure 8. Transit queue occupation and channel utilization in a 32x32 torus at 1 phit/node/cycle applied load, for 1-4 injection ports, and for shortest and lpath injection port selection policies. Each column represents a virtual (transit) channel.

In this large network, all configurations achieved similar peak throughput for uniform traffic. Furthermore, all of them exhibited significant throughput loss for loads above their saturation point. Figure 8 shows the buffer occupation and channel usage. We can see they all exhibit a usage pattern similar to the one given by *4_Inj_sh* in the smaller network, which indicated that network congestion soared for loads above saturation. The injection policy has only a minor impact on the level of congestion reached by the adaptive network, and in all cases there is a significant throughput loss due to congestion.

In the larger network, the HOLB at injection cannot prevent nodes from filling up the network buffers in the direction that saturates first. As the radix increases, packets travel longer paths, and more nodes contribute to the saturation of a given link. At the same time, network backpressure takes longer to reach the sources. Thus, the restrictive effect of HOLB arrives too late, when network buffers are already (almost) full.

In short, our network evaluation has identified network size and the number of injection channels as important factors in the way congestion builds up in fully adaptive networks. As parallel systems grow larger, as well as adding more packet sources, congestion will become a critical factor that needs to be dealt with.

5. Congestion control and related work

The above evaluation indicates that the adaptive bubble router is not stable for heavy loads and large networks, suffering up to a 20-25% loss due to increased network congestion. This is a common problem for fully adaptive routers because the strategies that reduce contention result on both high peak throughput and the ability for new packets to obtain network resources when they are scarce. In other words, low contention leads to higher congestion at heavy loads. As we have observed in Section 4.2, the throughput loss in the ABR is minor compared to other fully adaptive routers [21], due to the constraints set by bubble flow control, which prevents nodes from using the last escape path resource (or resources) of its local router.

Congestion is exacerbated when more injection queues are added. This is true for any VCT adaptive router based on Duato's approach [13], regardless of the choice of escape network. Thus, we have two choices:

1. Allow new and incoming packets to compete on a fair basis for the output links and accept the congestion this causes.
2. Limit congestion by giving priority to transit traffic, which introduces the risk of starvation for nodes whose associated routers are swamped by upstream traffic.

In this section we present results that show the gains achieved by the latter, which is also the choice taken by the multicomputer manufacturers.

5.1 Throttled injection

As we have seen in Section 4.2, BFC provides congestion control for escape virtual channels, as it allows packet injection when having space at least for two packets at the requested virtual channel.

In addition, we can use BFC to limit the packet population in the adaptive channels as well. We can apply BFC only to the escape channels (*2-Esc*) as the standard ABR, or to all requests from any injection port to **any** virtual output link (*2-All*). Besides, we can limit the network population by increasing the number of free resources required to grant injection from the minimum of 2 packets in the local queue to 4 (*4-All*), or even 6 (*6-All*). Figure 9 shows the impact that these restrictive injection policies have on maximum sustained throughput in the 32x32 torus, for the single and multi-queue injection interfaces.

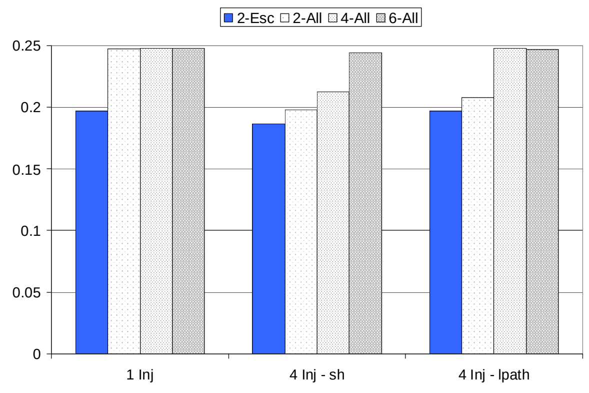
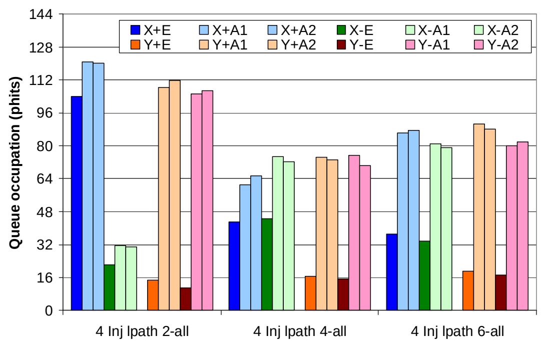


Figure 9. Sustained throughput at saturation, in a 32x32 torus, for 1 and 4 injectors, for shortest and lpath injection port selection policies, and different restrictive injection strategies.



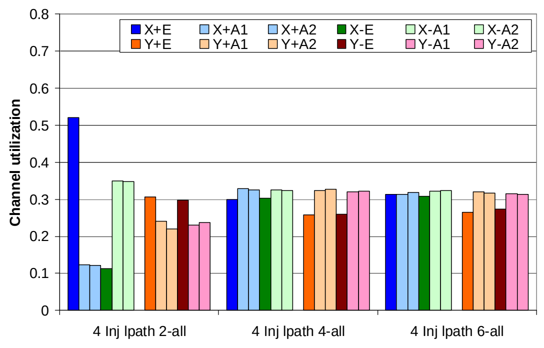


Figure 10. Transit queue occupation and channel utilization in a 32x32 torus at 1 phit/node/cycle applied load, for 4 injection ports, and lpath selection policy. Each column represents a virtual (transit) channel.

As expected, restrictive injection reduces buffer occupation and increases channel utilization in all cases. All of them are able to reach and sustain maximum performance, very close to the bisection limit (0.25 phits/node/cycle). For the single-queue case, a bubble of size 2 is enough to prevent congestion in the adaptive network, because of the additional restrictive injection caused by the HOLB intrinsic of FIFO queues.

Figure 10 shows the buffer occupation and channel utilization for *4_Inj_lpath* with variable bubble size, from 2 to 6. Note that although we apply the same bubble size to all virtual channels, buffer occupation in the adaptive channels is always higher than in the escape channel, because packets in the adaptive network can move from one dimension to the other with no restriction but VCT. We can see that, for *2-All*, there is an asymmetry in channel usage as well as in buffer occupation between the +X and –X, which clearly shows how the network is congested in +X. To control congestion in the adaptive network, we have to increase the bubble size from 2 up to 4. This reduces buffer occupation in all +X transit queues, which is the direction that saturates first in this set of experiments, and balances the use of the three virtual channels.

The most restrictive injection policy (*6-All*) reduces the buffer occupation in all escape channels as expected. On the other hand, this strategy also prevents many transit packets from using their escape paths, so that buffer occupation in the adaptive channels is actually increased, as shown in Figure 10. Hence, the minor reduction in network throughput observed in Figure 9.

In any case, this set of experiment proves that a large fully adaptive network can achieve and sustain high throughput, provided that it manages to control congestion at high loads. This can be achieved with one or multiple injection ports.

5.2 Other traffic patterns

The conclusions from this analysis are valid for uniform traffic patterns in which the number of packets injected into each direction is balanced. Note that permutation patterns have a very unbalanced used of the injection queues, as all packets from one node have a common destination. Any static selection will put all generated traffic into a single queue, thus using only one of the injection ports. Thus, we have chosen the hot-region pattern as described in [6] as a representative non-uniform pattern. Results from experiments using this pattern are represented in Figure 11.

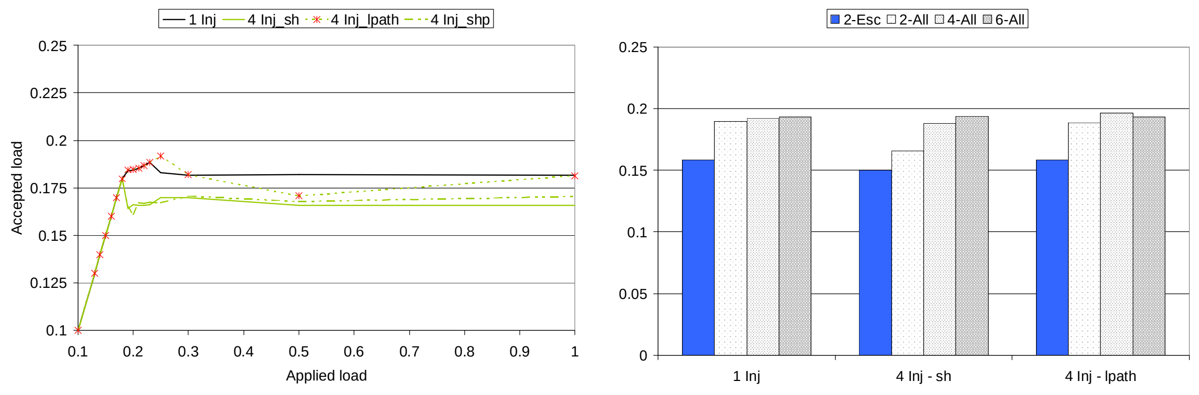


Figure 11. Data from experiments with a 32x32 torus, under hot-region traffic. Left: accepted vs. applied load for 1 and 4 injectors using different selection policies. Right: sustained throughput at saturation for 1 and 4 injection ports, for shortest and lpath selection policies, and different levels of restrictive injection based on BFC.

We can see that the network response follows the same trends that in the case of uniform traffic, but with lower peak values, due to the concentration of traffic in the hot-spot sub-region. Again, it is necessary to limit congestion in the adaptive network in order to sustain peak network throughput for heavy loads.

5.3 Related work

Finally, we will briefly review other works that are related to the analysis presented above. To the best of our knowledge, no other work has analyzed the impact of multiple injection ports on network performance at full loads. Badak and Panda [2] analyzed the consumption bottleneck in wormhole torus network, by increasing the number of consumption channels from 1 to 2d. Although they also deal with the router interface, it covers a totally different design space: it considers small radix network, in which injection and consumption are limited by the node-to-router bandwidth, it uses wormhole flow control and it focuses on its impact on network delay. Even though they mentioned in the paper that for symmetry they increase injection bandwidth, there is no description of how this additional bandwidth is used.

Other recent works, such as [19] propose the use of multiple injection queues, one per destination, to obtain backpressure information and re-direct traffic over less congested areas. However, their experimental setup indicates that each node has an infinite source queue to model the network interface, thus modeling a single injection port per node; little is said about the mapping of those multiple queues into the injection port.

In [13], Duato et al. briefly review the effect of multiple injection ports in a fully adaptive wormhole router, for which adding more ports decreases latency and increases throughput. Obviously, the flow control technique has a significant impact on the way congestion builds up. However, they do not include in their plots data for injected loads beyond saturation "*for the sake of clarity*". Additionally, no mention is done to the policy used to allocate packets to ports.

Congestion control is a well-known issue in computer networks, but in the context of direct interconnection networks there is only a handful of papers. The adaptive wormhole router by Dally and Aoki [12] was one of the first to report throughput loss under uniform traffic. This work modeled a 16x16 mesh with 16 virtual channels per physical link. Throughput loss appeared only when using dynamic adaptive routing, a non-minimal adaptive strategy with a limited number of dimension reversals. This loss was reduced by limiting the number of virtual lanes that a packet at the injection port could access. On the same year, Boppana and Chalasani [8] also suggested the need to limit injection based on the number of messages of the same class queued in the node. A few years later, Baydal, Lopez and Duato extended Dally's limited injection to their fully adaptive wormhole router, using a range of metrics to estimate congestion [3,4,5]. Thottethodi et al. [20] proposed a global knowledge based congestion control mechanism instead of relying on network backpressure. This allows congestion to be detected earlier but involves considerable cost in order to gather global information.

We should note that all methods that limit injection based on local information, including BFC, have practically no cost, but they can be less effective for non-uniform traffic patterns.

Other routers provide congestion control by giving priority to blocked packets over new packets at the network interface. For example, the Alpha 21364 prioritizes packets according to their class and the input port they arrive: a policy called *rotary rule* [16] gives priority to packets arriving from an inter-processor port over packets generated by the local injection ports. The Chaos router [7] deals with congestion by both misrouting packets, and giving blocked packets at its central queue priority to advance ahead of new or incoming packets. However, it is difficult to estimate the contribution given by either misrouting or the priority mechanism. A more recent proposal [21] also applied lazy misrouting in order to reduce congestion in an adaptive VCT router; although effective, it has a considerable cost.

We should note that all these methods were proven effective in reducing congestion for the particular router they were proposed for. Most evaluations were done for medium network sizes with single injection queues. Besides, they are difficult to compare with, because each of them have small input buffers of different size which reflected the technological constrains of their times.

6. Conclusions

This work has provided an insight into the impact that the number of injection channels and their management has on the sustained performance of adaptive VCT networks of medium to large size. The key findings of this work are as follows:

1. An injection policy regulated only by VCT flow control is not a good strategy. Results have shown that for small and medium size networks, the head of line blocking of a single injection queue is a blessing in disguise, as it prevented the processing node from flooding the router in *all* directions. Once a channel is saturated, and a packet is blocked at the head of the injection queue, channels in other directions have a chance to drain their load. In larger networks, this is not enough to prevent new packets from accessing the scarce network resources, so that network congestion rises and channel utilization drops.
2. Adding multiple injection channels does not increase network throughput, for the analyzed traffic patterns. Instead, it increases the injection pressure at high loads and allows the processing nodes to flood their routers with packets, leading to higher congestion and significant throughput loss at heavy loads.
3. Large networks (and/or routers with multiple injection ports) need a mechanism to limit injection in order to sustain maximum throughput. For the studied traffic patterns, simple, local congestion control mechanisms may bring up this performance figure up to a 20-25%.

In short, congestion control is becoming a critical issue for current parallel systems. We have identified the network configurations that are more susceptible to suffer network degradation. Further research is required to find simple and effective injection policies that are starvation free and provide maximum sustained throughput.

References

[1] NR Adiga, GS Almasi, Y Aridor, M Bae, Rajkishore Barik, et al., "An Overview of the BlueGene/L Supercomputer", Proc. of SuperComputing 2002, Baltimore, Nov. 16-22, 2002

[2] D. Basak and D.K. Panda "Alleviating Consumption Channel Bottleneck in Wormhole-routed k-ary n-Cube Systems". IEEE Trans. on Parallel and Distributed Systems, vol. 9, no. 5, pp. 481-496, 1998.

[3] E. Baydal and P. López: A Robust Mechanism for Congestion Control: INC. Euro-Par 2003: 958-968

[4] E. Baydal, P. López and J. Duato: Congestion Control Based on Transmission Times. Euro-Par 2002: 781-790

[5] E. Baydal, P. López and J. Duato: A Simple and Efficient Mechanism to Prevent Saturation in Wormhole Networks. IPDPS 2000: 617-62

[6] M. Blumrich, D. Chen, P. Coteus, A. Gara, M. Giampapa, P. Heidelberger, S. Singh, B. Steinmacher-Burrow, T. Takken, P. Vranas. "Design and Analysis of the BlueGene/L Torus Interconnection Network" IBM Research Report RC23025 (W0312-022) December 3, 2003.

[7] Bolding, M. L. Fulgham, L. Snyder, "The Case for Chaotic Adaptive Routing. IEEE Trans. Computers 46(12): 1281-1291 (1997)

[8] R.V. Boppana and S. Chalasani, " A Comparison of Adaptive Wormhole Routing Algorithms." 20th Annual Int'l Symp. on Computer Architecture (ISCA), pp. 351-360, May 1993.

[9] C. Carrión, R. Beivide, J.A. Gregorio and F. Vallejo, "A Flow Control Mechanism to Prevent Message Deadlock in k-ary n-cube Networks", Proceedings of the Fourth International Conference on High Performance Computing (HiPC'97). Bangalore, India. Diciembre 1997.

[10] W.J. Dally: "Virtual Channel Flow Control", IEEE Trans. on Parallel and Distributed Systems, vol. 3, no. 2, pp. 194-205, 1992.

[11] W.J. Dally, C.L. Seitz, "The Torus Routing Chip" *Distributed Computing* vol **1** pp. 187-196. 1987

[12] W.J. Dally and H. Aoki, "Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels", IEEE Trans. on Parallel and Distributed Systems, vol. 4, no. 4, pp. 466-475, 1993.

[13] J. Duato, S. Yalamanchili and L. Ni, "Interconnection Networks: an engineering Approach. Revised Printing", Morgan Kaufmann, 2003.

[14] J. Duato. "A Necessary and Sufficient Condition for Deadlock-Free Routing in Cut-Through and Store-and-Forward Networks". IEEE Trans. on Parallel and Distributed Systems, vol. 7, no. 8, pp. 841-854, 1996.

[15] J. Miguel-Alonso, J.A. Gregorio, V. Puente, F. Vallejo and R. Beivide. "Load Unbalance in k-ary n-cube Networks". In Proc. Europar 2004 Pisa September 2004, pp. 900-907.

[16] S. Mukherjee, P. Bannon, S. Lang, A. Spink and David Webb, "The Alpha 21364 Network Architecture", IEEE Micro v. 21, n. 1 pp 26-35.

[17] V. Puente, C. Izu, J.A. Gregorio, R. Beivide, and F. Vallejo, "The Adaptive Bubble router", Journal on Parallel and Distributed Computing, vol 61, no. 9, pp.1180-1208 September 2001.

[18] S. L. Scott and G. Thorson, "The Cray T3E networks: adaptive routing in a high performance 3D torus", Proc. of Hot Interconnects IV. 1996

[19] A. Singh, W. J. Dally, A. K. Gupta, B. Towles, "Adaptive channel queue routing on k-ary n-cubes". SPAA 2004: 11-19.

[20] M. Thottethodi, A.R. Lebeck and S.S. Mukherjee, "Self-Tuned Congestion Control for Multiprocessor Networks". In Proc 7th Int. Symp. on High Performance Computer Architecture (HPCA-7), pp. 107-118, January 2001.

[21] M. Thottethodi, A.R. Lebeck and S.S. Mukherjee, "BLAM: a High-Performance Routing Algorithm for Virtual Cut-through Networks". In Proc Int. Parallel and Distributed Processing Symposium IPDPS 2003: 45