Improving the Performance of Large Interconnection Networks using Congestion-Control Mechanisms

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

[1]Dep. of Computer Architecture and Technology, The University of the Basque Country, 20080 San Sebastian, Spain
miguel@si.ehu.es
[2]Department of Computer Science, The University of Adelaide, SA 5005 Australia
cruz@cs.adelaide.edu.au
[3]Computer Architecture Research Group, Universidad de Cantabria, 39005 Santander, Spain
monaster@unican.es

Abstract

As the size of parallel computers increases, as well as the number of sources per router node, congestion inside the interconnection network rises significantly. In such systems, packet injection must be restricted in order to prevent throughput degradation at high loads. This work evaluates three congestion control mechanisms on adaptive cut-through torus networks under various synthetic traffic patterns.

A range of network parameters (radix, number of injection channels, deadlock avoidance method) is used to cover the current network design space. Traffic is generated using bursts of data exchanges (instead of a Bernoulli process) to reflect the synchronized nature of data interchanges in parallel applications. Simulation results show that large networks perform their best when most network resources are dedicated to in-transit traffic. Besides, local congestion control mechanisms are nearly as effective as the more costly global ones for both uniform and non-uniform traffic patterns.

Keywords

Interconnection networks, congestion control, synchronized workload, adaptive virtual cut-through torus.

1 Introduction

The interconnection network is a key element of a tightly coupled multiprocessor system. It provides low latency and high bandwidth communication for a variety of workloads. As the standard microprocessor is being replaced by multithreaded ones or by chip multiprocessors, both the number of injectors[1] and the total

[1] Injection channels.

offered load per node has increased significantly. For example, the Alpha 21364 router [13] has 4 injectors (from the two on-chip memory controllers, the I/O external chip and the on-chip second level cache) and the BlueGene/L torus network [1] has 8 (one port per dimension plus two high-priority).

Besides, network bandwidth does not scale with the number of nodes. For example, in an 8x8 torus with bidirectional links, under uniform traffic, each node can inject up to 1 phit/cycle before reaching the bisection bandwidth limit. In comparison, a 32x32 torus has a limit of 0.25 phits/cycle/node. Consequently, large networks become congested more easily than small networks. Furthermore, networks with large radix (more than 20 routers per dimension) experiment additional performance degradation, due to an unbalanced utilization of network resources.

Both wormhole (WH) and virtual cut-through (VCT) routers can suffer from congestion at high loads, when their respective buffers become full. Once started, congestion keeps increasing due to the tree saturation effect, which builds up quicker in a WH network as messages spread amongst multiple nodes. VCT provides less contention at medium loads by storing each blocked message in a single router queue but, when that a blocked packet finally advances, only a single channel is released. In other words, less contention means that tree saturation takes longer to appear in a VCT network, but it is more persistent.

Although congestion is a common problem in all types of networks, up to now it has not been a critical issue in interconnection network design. The reasons for this are multiple. Firstly, the size of most systems (real or modeled) ranged from 64 to 512 nodes, all having single injection queues, whose HOLB (head-of-line blocking) was providing enough *hidden* control to prevent the node from flooding the network. Secondly, the software overheads were high, so that it was rare for a real system to provide loads close to the 100% utilization. Thirdly, networks with few resources (2 or 3 virtual channels and buffers of a few phits) kept congestion low because blocked messages spread along their paths, limiting network throughput due to contention. Thus, it is not surprising that most of the research effort went into reducing contention by increasing adaptivity and dealing with the problems this brought in, such as deadlock [5, 15, 16]. Finally, most simulation studies were carried out with loads normalized to the theoretical bisection limit for random traffic; it was uncommon to evaluate the network under very heavy loads. Actual application traffic is often described as a series of alternating phases of low (or null) network utilization followed by phases of intense network utilization [4, 14], where processes send a series of packets as fast as they can (for example, to complete an all-to-all collective operation). The latter are the phases that can easily saturate the network. Performance degradation has been observed only when the simulator models multiple injection channels and loads above the theoretical limit (as can be done with tools such as FlexSim [18], which incorporate virtual injection channels). Thus, some effort has gone into reducing congestion on wormhole adaptive routers [2, 19].

A representative router would provide minimal adaptive routing[2], to support multiple workloads, and use avoidance or recovery mechanisms to deal with network deadlocks. As the buffer capacity is usually large, our choice of flow control is cut-through. Deadlock can be avoided by applying the bubble condition as described in [15], which was the choice for IBM's BG/L torus network. Other routers such as the Alpha 21364 use virtual channels (VCs) to break the ring cyclic dependencies [6]. In the two cases, networks route packets in an adaptive, but still deadlock-free, fashion, by combining a deadlock-free escape sub-network with a minimal adaptive sub-network. For completeness, we will consider both router designs.

This work evaluates local and global mechanisms to control congestion in adaptive VCT tori of various sizes (8x8, 16x16, 32x32) with single and multiple injectors. Local mechanisms are virtually cost-free and work well under uniform traffic patterns. Global control could be more effective with highly non-uniform traffic patterns, when saturation cannot be detected locally, but has a significant cost in collecting and distributing global status information. On the other hand, if an additional network for collective communication (such as the tree-network of IBM's BG/L) is available, it could be a viable alternative.

The main contributions of this paper can be summed up in the following points:

1. We show that congestion control mechanisms are essential to maintain peak performance at loads beyond saturation, and have no effect for loads below that point.
2. In particular, we focus our analysis in large networks with more than one injector—configurations that are common in current parallel systems. These systems *incorporate* congestion control mechanisms, but as far as we know there are no studies comparing the different alternatives used by manufacturers.

[2] Adaptive routing that follows only minimum-distance paths.

3. We show that local congestion-control mechanisms are effective for uniform as well as non-uniform traffic patterns.
4. We show how Bubble routers provide congestion control in their escape sub-networks. Thus, they maintain reasonable performance levels at heavy loads. They benefit from additional control applied to the adaptive sub-network.

The rest of this paper is organized as follows. Section 2 describes the adaptive router and the different congestion control mechanisms applied to it. Section 3 describes the simulation process and provides two alternatives to the temporal distribution of non-uniform synthetic traffic patterns. Section 4 presents the experiments performed for a range of traffic patterns and network configurations, and evaluates their results. Finally, section 5 summarizes the findings of this work.

2 Router design and congestion control

This section describes a range of congestion control mechanisms that can be applied to adaptive cut-through networks. To be more specific on their description, we will first present the architecture of the routers used in this study, and then describe in detail the mechanisms applied to them.

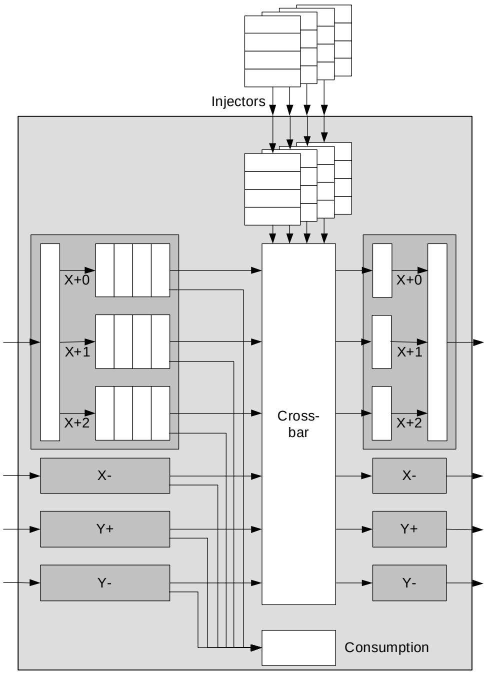


Figure 1. Architecture of the adaptive, VCT routers used in the experiments. Note that there are 4 injectors, each one with a dedicated physical link, connected to the crossbar. Management of virtual channels may be Bubble or Classic.

2.1 The adaptive routers

The router architecture used in this work is modeled as shown in Figure 1. This is a virtual cut-through (VCT) router with three virtual channels (VCs) per physical channel, to map both an oblivious (dimension-order routing) and a minimal adaptive virtual network. The deadlock-free oblivious sub-network is used as the escape path for any potentially deadlocked packet in the adaptive sub-network [8]. Such combination provides low-cost, deadlock-free adaptive routing.

Both the Alpha 21364 and the BG/L torus networks use this strategy, although they differ in their choice of the deadlock avoidance mechanism for the escape sub-network. The older Alpha network breaks cycles within a dimensional ring by using two virtual channels, as in the Torus Routing Chip [6]. The newer BG/L relies on Bubble Flow Control (BFC) [15], which prevents the node from injecting a packet if such action exhausts the local escape resources. From now on, we will use the term *Classic* to refer to an adaptive router like that of the Alpha 21364, and *Bubble* to refer to a router like that of the torus network of the BG/L.

The Bubble router uses only one VC for the escape sub-network, with no loss of functionality; in fact, as we will show later in this paper, it also provides partial congestion control. Therefore, it has two adaptive virtual channels, while its Classic counterpart has just one.

2.2 Congestion control mechanisms

Congestion control mechanisms limit injection when the network reaches a given level of congestion. They can be classified by the way congestion is estimated: locally or globally.

Local methods are simple because each node restricts its own injection based on the congestion level it observes in its own router. We will consider two different approaches:

- In-transit-priority restriction (**IPR**). For a given fraction **P** of cycles, priority is given to in-transit traffic, meaning that, in those cycles, injection of a new packet is only allowed if it does not compete with packets already in the network. P may vary from 0 (no restriction) to 1 (absolute priority to in-transit traffic). This is the method applied in IBM's BG/L torus network, in which P may take any value, although published evaluations of this network [3] have been carried out with P=1. Similarly, the Alpha 21364 network incorporates the "rotary rule" [13], which gives priority to in-transit traffic.

- Local Buffer Restriction (**LBR**). The bubble condition provides congestion control for the escape sub-network [11]. LBR consists on applying the same restriction to new packets that request an adaptive virtual channel. That is, a packet can only be injected into an adaptive virtual channel if such action leaves room for at least **B** packets in the transit buffer associated to that virtual channel. In other words, the parameter B indicates the number of buffers reserved for in-transit traffic. The adaptive bubble router as in [15] corresponds to the case of B=0 (no restriction).

Global methods estimate network congestion based on the level of congestion on the whole network, so that a mechanism is needed to gather and distribute this information. We use a global mechanism similar to the one described in [19], which estimates congestion based on buffer occupation. This Global Buffer Restriction (**GBR**) method collects the percentage of buffer utilization in the whole network and distributes this value to all nodes each **D** cycles. All nodes suspend injections if that utilization exceeds a given threshold **T**. The base case (no restriction) is in place when T=100%.

A different classification of these three mechanisms could be the following:

- **Utilization-based**: restriction is applied when resource utilization (for example, buffer occupation) exceeds a certain thresholds. LBR and GBR fall in this category.
- **Priority-based**. Higher priority is given to a certain class of packets, for example, those in transit (vs. those waiting to be injected). IPR falls in this category.

The utilization of a congestion control mechanism (or the lack of it), may have a significant impact on the performance of a parallel system. In [12], the evaluation of a 2D torus with radix 32 showed that head-of-line blocking at the injection queue is a factor that limits injection at saturation loads and introduces asymmetry in the use of network resources. Further evaluations of large networks with multiple injectors under uniform and hot-region traffic patterns was carried out as described in [9]. As the multiple injectors reduce the HOLB at the injection queues, at heavy loads the adaptive sub-network gets clogged with packets, regardless of network size. Network throughput drops to nearly that of the escape sub-network. The LBR mechanism proved to be quite effective for Bubble networks under uniform traffic, as local conditions at any given router are representative of the saturation level of the whole network. Is this mechanism still effective under non-uniform traffic patterns? How does it compare to other local or global mechanisms? One of the the goals of this study is to find answers to these questions.

3 Evaluation methodology

For many practical reasons, most performance studies of interconnection networks are carried out using synthetic traffic, running a simulator for a large number of cycles (simulated time) to get performance results with the network in steady state. Although this is not realistic, we consider the obtained results as indicators of the level of performance the network could provide under real traffic. For some SPLASH applications such as Radix or LU, it has been shown to be a reasonable approach [15].

The traffic workload is defined by its traffic pattern, and its temporal and message length distributions. The traffic pattern determines the distribution of destinations for each source node. The temporal distribution determines when a packet is generated. The message length distribution determines the size of each message. To limit the number of experiments, and taking into account that we are using VCT, in this study we only consider fixed-size messages that match the packet size.

3.1 Spatial traffic patterns

These are the traffic patterns, commonly seen in the literature, used in our experiments:

- **BR**: bit-reversal permutation. The node with binary coordinates (a_{k-1}, a_{k-2}, ..., $a_{1,}$ a_0) communicates with node (a_0, a_1, ..., $a_{k-2,}$ a_{k-1}).
- **SH**: perfect-shuffle permutation. The node with binary coordinates (a_{k-1}, a_{k-2}, ..., $a_{1,}$ a_0) communicates with node (a_{k-2}, a_{k-3}, ..., $a_{0,}$ a_{k-1}) – rotate left 1 bit.
- **TR**: transpose permutation. In a 2-D network, the node with coordinates (x, y) communicates with node (y, x).
- **TO**: tornado permutation. Each node sends packets $(k-1)/2$ hops to the right in the lowest dimension, where k is the network radix. [20]
- **UN**: uniform traffic. Each node selects destinations randomly in a packet-by-packet basis.

- **HR**: hot-region traffic. 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 choose their destinations uniformly over the entire machine. [3]

Of these, BR, SH, TR and TO are permutations (a given source node always sends packets to the same destination node) while in UN and HR each node select destinations randomly.

3.2 Temporal distribution of packet generation

In the literature we can find a range of options for the temporal distribution of packet injections. We can classify these distributions in two groups:

- **Independent traffic sources**. In this case, all nodes are "programmed" to inject packets using some probability distribution. Each node progresses independently of the others. Injection times may follow a Poisson or Bernoulli distribution (that are smooth over large time intervals) or on-off models that better characterize the self-similarity of traffic in some applications [17]. Many simulation-based studies of interconnection networks follow this approach.
- **Non-independent traffic sources**. The assumption of independent traffic sources ignores some key characteristics of real applications: most data exchange is reactive in nature, and many operations include (explicitly or implicitly) synchronization. We may simulate interchanges such as those required to implement an MPI_Alltoall() global operation, or client-server traffic (where a server node sends packets to respond to the reception of packets from clients).

A complementary study, analyzing the impact that the traffic sources model has on the evaluation of the IPR technique [10], indicates that the independent sources assumption yields, at heavy loads, results that may not be representative of network performance for parallel applications. To be self-contained, the following subsections explain first the reasons to choose a burst-synchronized workload and then describe the temporal distribution used in this work.

3.2.1 Independent traffic sources

Most studies assume a network in which all nodes are generating packets at the same given rate. Network performance is measured as packets delivered per node per cycle, but it is usually measured as number of packets delivered in a given interval divided by the interval length and the network size. In other words, this is the average network performance, which is expected to be even amongst the network nodes.

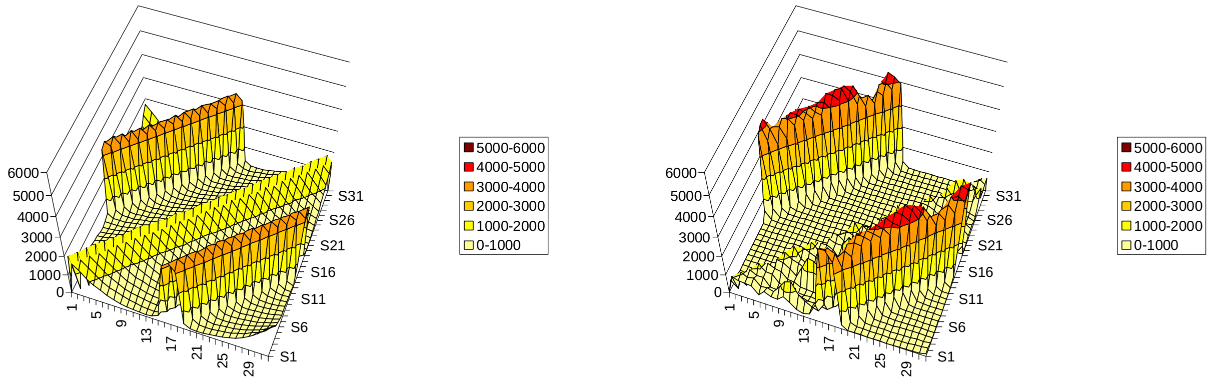
We should note that the time it takes a packet to be injected by any given node depends on the local router state, with or without restrictive mechanisms. Under UN traffic, the network load is evenly distributed, so that all nodes are able to inject packets at a similar rate. Under non-uniform loads, such as TR to name one, the occupancy of the output channels may vary widely from one router to another. Therefore, at high loads, nodes connected to busy routers have lower chances to inject than nodes in less used areas—which results in notable differences in the number of packets injected by each node. This is reflected in the average distance, which changes with the load. This distance, for a TR permutation in a 32x32 network, has a value of 16.5 hops—that matches what the simulator reports for loads below saturation. However, at loads beyond saturation, the simulator reports a value of 17.1. This is because nodes in two bands parallel to the diagonal are able to inject at a higher rate, as shown in the injection map of Figure 2a. In other words, it seems that the network is *unfair* for TR traffic[3].

When local congestion control mechanisms are added to the Bubble network this unfairness is magnified, as shown in Figure 2b: nodes in the busiest areas reduce their injection rates while the nodes in the less used areas continue to pour their packets at even higher rates; consequently, the average distance rises to 23.5.

Note that starvation is the extreme case of unfairness in which a node never gets the chance to inject. Any routing mechanism that favors in-transit traffic (imposing restrictions to the injection of new packets) may suffer from starvation, if it is in the path of intensive traffic generated by an independent source. A router that is starvation-free may still be unfair; the time it takes to inject a packet, although bounded, would be different for each node depending on its router workload.

[3] This behavior is not caused by the simulator or router architecture; for example, simulations of an adaptive WH network using FlexSim under TR traffic exhibit the same fairness problem.

Note that, as IPR favors packets that travel longer paths, the network exhibits lower throughput (packets delivered per node per cycle), in spite of having higher channel utilization. Does IPR degrade performance, or not?



(a) (b)

Figure 2. Maps of injected packets for a network of 32x32 Bubble routers, under TR traffic beyond saturation. Each surface point represents the number of packets injected by a node: the darker, the larger. (a) No injection restriction, and (b) IPR, for P=1.

One way to deal with unfairness is to measure throughput as the lowest injection rate that matches the desired workload [7]. This measuring methodology is correct in the context of infinite, independent sources of traffic, such as in local area networks. Applications running in a parallel system do not work that way. As we stated before, their processes are somehow *coupled*, because they work to perform a given task in a cooperative way. It is true that worst-case performance for data exchanges is important (as shown in [14]) because it may halt progress of computation nodes, which are not able to perform additional operations, or communicate any further, until the data exchange has been completed. However, we cannot conceive a realistic scenario in which, *in the same parallel application*, a process is sending packets to its selected destination *ad infinitum* while other nodes do the same at a *much* smaller rate.

3.2.2 Burst-synchronized workloads

The previous sub-section explains the rationale behind using non-independent traffic sources. Most (if not all) applications have some synchronization barrier, perform collective operations or other mechanisms that make all the processes advance at a similar rate. To reflect this synchronized nature of application workload we have implemented a traffic generation mechanism similar to that described in [4]: burst (or bulk) synchronized traffic.

We assign the same workload to each source of traffic, modeling a system of *b* data exchanges following a given traffic pattern. In other words, each node generates *b* packets in a single burst to be transmitted to the other nodes using the selected spatial pattern. The burst ends when all packets of all the traffic-generating nodes have been consumed. Then we measure the (simulated) time it takes for all these operations to complete.

With this new measuring mechanism, maps of injected packets are meaningless: all nodes inject exactly *b* packets per burst[4]. If an injection restriction mechanism favors messages traversing long distance, the corresponding injecting nodes will deliver their workload sooner than the other nodes, and then they will not interfere with the remaining traffic. If we still want to force the network to work in saturated mode for a long period of time, we only need to make *b* large enough.

Note that, in this context, latency is not comparable to that measured under the assumption of independent sources. In the latter case, at loads beyond saturation, per-packet latency is not stable, because the network cannot reach a steady state, and only network throughput is reported. Now per-burst latency is a manifestation of the throughput supplied by the network: the higher the network throughput, the shorter the time it takes to deliver the assigned workload.

3.3 Simulation parameters

The previous section has described the traffic generation method used in this work. To make the experiments reproducible we need to describe the rest of the parameters used in the simulations:

- **Network size**: we are interested in large parallel systems, so most experiments are run in a 2D torus with radix 32. To include small and medium system, we will also consider radix 8 and 16. In all cases, we restrict the experiments to full-duplex links.
- **Router**: all routers have 3 virtual channels, each one with an input queue with capacity for 8 packets. Packet length is fixed to 16 phits. If the router is Bubble, one VC is configured to form the escape network, while the other two are used in an adaptive fashion. For Classic routers, two VCs form the

[4] For the same reason, starvation is not an issue in this context.

escape network, and the remaining one is adaptive. Unless otherwise specified the number of injectors is 4.

- **Traffic patterns**: For the spatial distribution, we use the patterns described in sub-section 3.1. For the temporal distribution, the simulator works in burst mode, for 5 consecutive burst of 1K packets each. This burst size is large enough to keep the network saturated for long periods of time. Reported times are those of completing 5 bursts.

4 Performance of congestion control mechanisms

This section reports the impact that injection restriction mechanisms have on the state-of-the-art interconnection networks. Firstly, we focus on a Bubble torus network of 32x32 nodes, under increasing restrictive injection for each of the three methods described in section 2.2. Secondly, we will extend this evaluation to Classic networks and to networks with smaller radix in order to show that the results are not specific of a particular kind of network.

4.1 Experiments with a 32x32 Bubble torus network

This section evaluates the three congestion control techniques for a Bubble network of 32x32 nodes. The rest of the parameters of the network are those described in Section 3.3.

Figure 3 shows the effect of applying the local buffer restriction (LBR) mechanism. For most traffic patterns, a bubble size of 1 packet has a very positive effect; a new packet can only be injected in an adaptive channel when it does not completely fill-up the corresponding buffer. This increases the chance of packets using the adaptive channel as the VCT condition will hold. Values of B greater than 2 exhibit minor gains for uniform or almost-uniform patterns (UN and HR), but there are not clear benefits for permutations—in fact, excessive restriction in SH is counterproductive.

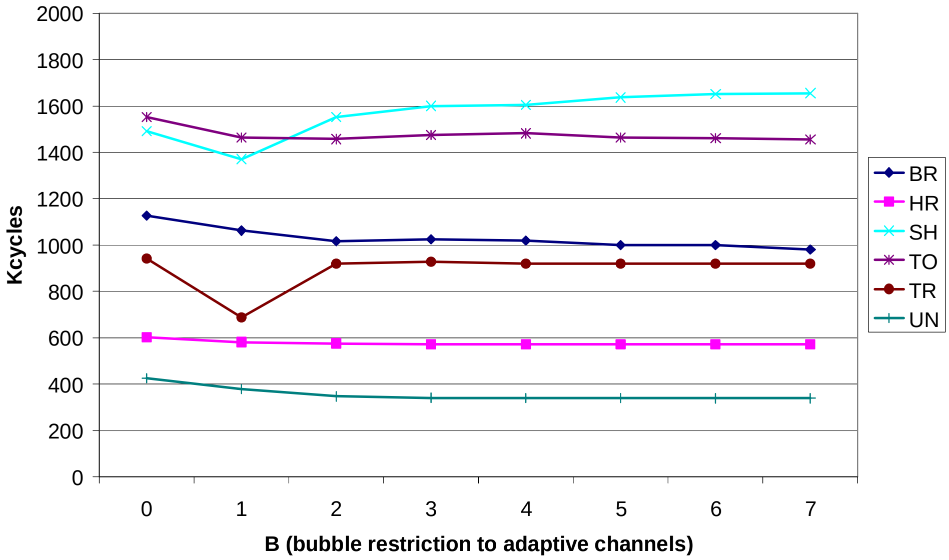


Figure 3. Effect of LBR restriction under different values of B, for a range of traffic patterns.

Figure 4 shows the effect of applying the in-transit priority restriction (IPR) mechanism. For most patterns, a low value of P has little effect on performance, as injection may be delayed a few cycles, but this is not enough to reduce network congestion. It can be observed that prioritization of in-transit traffic, applied at the maximum level (P=1) is very positive for all traffic patterns—except, again, for SH. Note that IPR with P=1 shows similar (slightly better) completion times that LBR with B=1. Both cases prevent each node from exhausting the last resources of the adaptive sub-network, so that packet progress is not prevented by the VCT flow control.

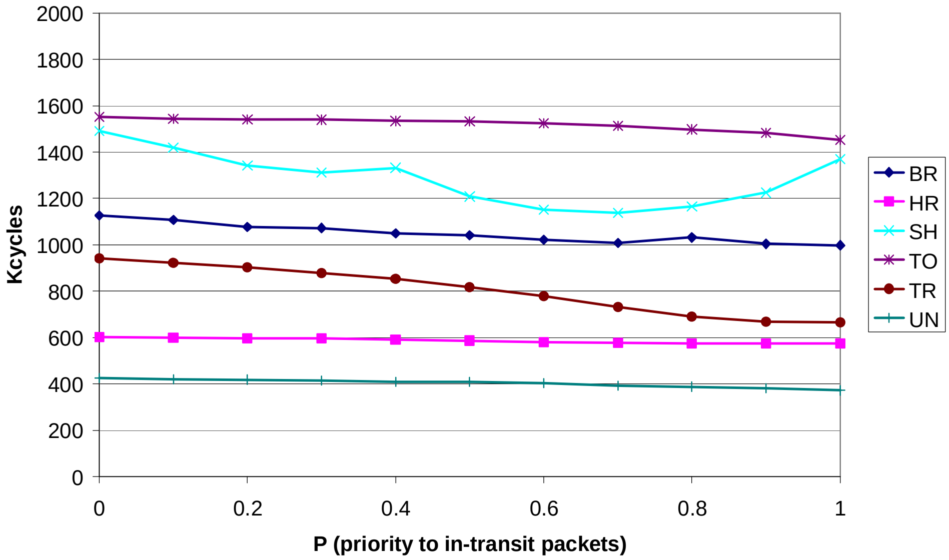
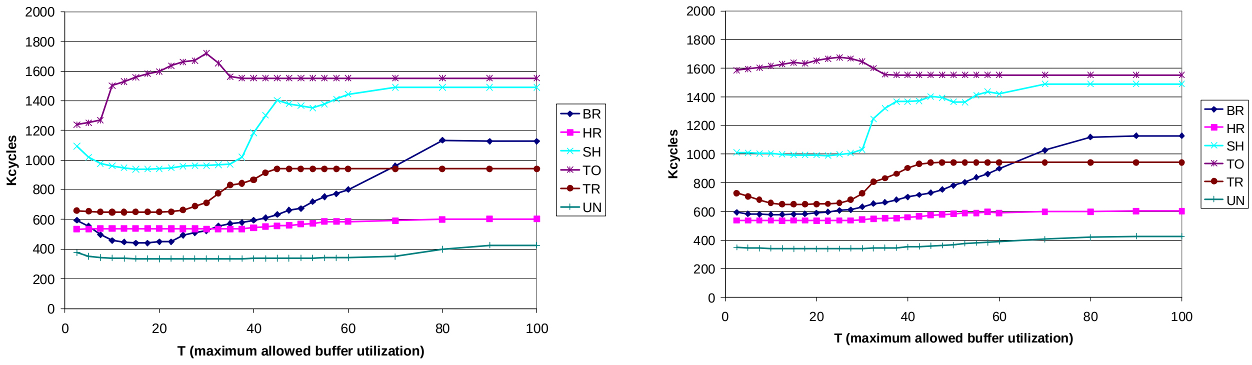


Figure 4. Effect of IPR restriction under different values of in-transit-priority P.

Figure 5 shows the effect of applying the global buffer restriction (GBR) mechanism. Figure 5a corresponds to GBR with D=1, which means that the global buffer utilization available to all the routers with a delay of just one cycle. Although this is not realistic, it provides an upper bound of the potential benefits of using GBR—more realistic values of D are discussed later.

Global buffer restriction can be very beneficial when the adequate threshold T is selected. Note that in a VCT network with large buffers, a low buffer occupation (a few packets of the 24 per network direction) is sufficient to keep the physical links busy; the remaining buffer space allows the network to cope with transitory fluctuations. Therefore, a buffer occupation in the range 10-20% provides substantial reductions in packet burst completion for most patterns when compared with the base case. Only for the tornado pattern (TO) GBR-1 shows itself as a potentially counter-productive measure. On the other hand, improvements for BR are spectacular.



(a)

(b)

Figure 5. Effect of GBR restriction for a range of values of threshold T. (a) GBR-1. (b) GBR-64.

In a real network it would be necessary to collect the buffer utilization at all nodes, aggregate these values, and broadcast the resulting global utilization to all the nodes. As the diameter of this network is 32, it would be very difficult to do this in less than 64 cycles. Figure 5b presents the results for GBR-64. Completion times are higher that those obtained with GBR-1, but the improvement is still very good, except for the TO permutation. Note that performance depends on T, and the optimal value of T in our VCT network does not change much with the traffic pattern. We can suggest 15% occupancy to be an adequate threshold for all patterns. This would be different in a wormhole network with shallow buffers as in [19], in which there is not a generic value of T that performs well in all (or almost all) contexts. For this reason, that work proposes a self-tuning mechanism to dynamically adjust T.

To finalize the comparison of congestion control mechanisms in a large Bubble network, Figure 6 summarizes the lower completion times attainable with each mechanism (LBR, IPR, GBR-1 and GBR-64), as well as the base case (no restriction).

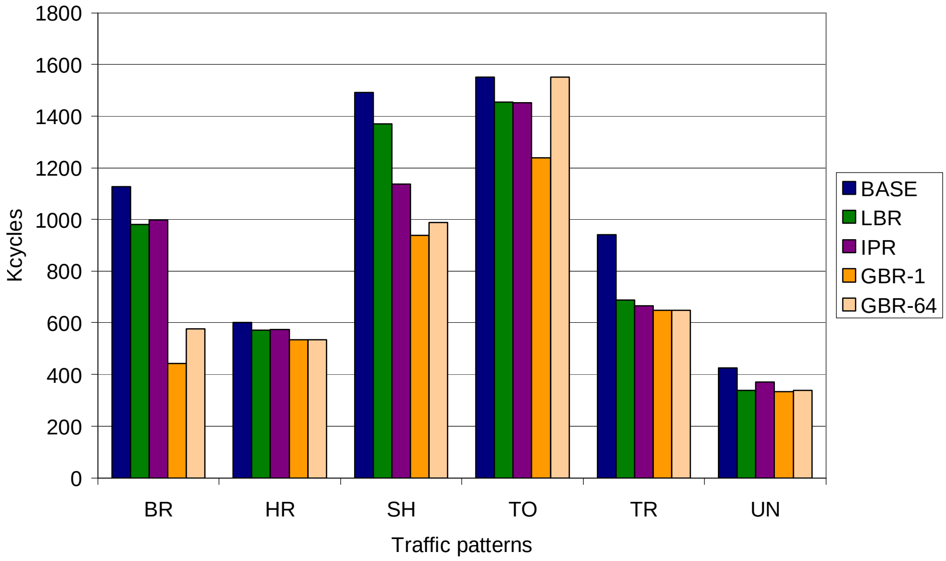


Figure 6. Comparison of best cases for different traffic patterns under different mechanism of injection restriction. The BASE case represents the performance without a restriction mechanism.

The conclusions drawn from this first set of experiments with a 32x32 Bubble torus are:

1. Injection control mechanisms are crucial to keep good levels of performance in large Bubble networks with multiple injectors. This is applicable to all the traffic patterns we have studied. Potential gains are higher for permutations.
2. GBR exhibits its maximum potential when global information is available immediately—a non-realistic restriction. When the delay in distributing this information is proportional to network diameter, it is still beneficial but in terms of cost it does not compete with the local counterparts.
3. Local injection control mechanisms provide good levels of performance with negligible implementation costs. It is a safe bet to use IPR with P close to 1, because it provides performance benefits for all traffic patterns. However, the optimal choice of B (either 1 or 2) for LBR depends on

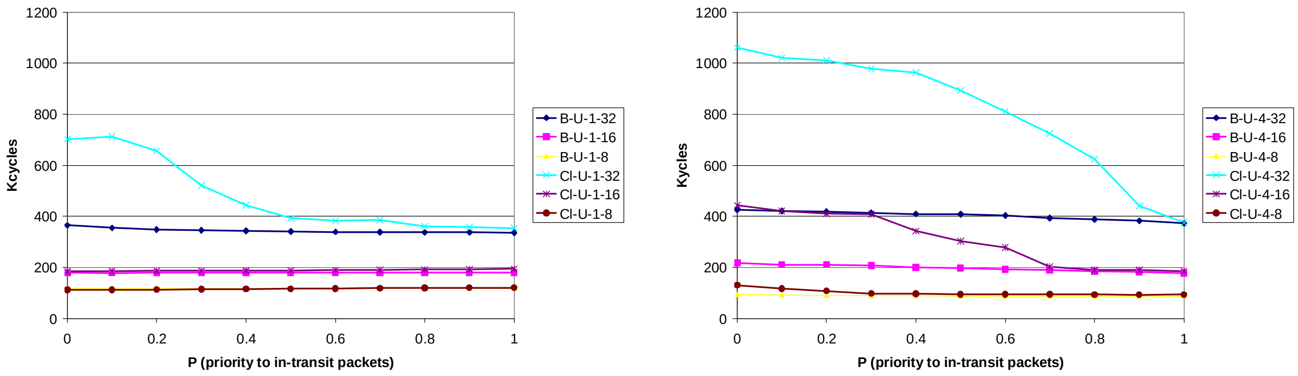
the traffic pattern. On the other hand, LBR is easier to implement, as it does not require modifications in router arbitration.

4.2 Other router configurations

The previous section has evaluated a large Bubble network with multiple injectors. In this section, we will extend the evaluation to networks of Classic routers as defined in Section 2.1. This architecture has been very popular to build the communication subsystems of multicomputers [13, 16]. In addition, we will show that congestion can also appear in networks of small radix.

We have performed a new set of experiments, for networks of sizes 32x32, 16x16 and 8x8, with either 1 or 4 injectors. We focus on the local injection restriction technique IPR because the LBR technique does not naturally fit in the Classic router.

Figure 7 shows the completion time for burst-synchronized uniform workload. For small network sizes (8x8), Bubble and Classic perform equally well, and there is no gain in adding an injection restriction mechanism, regardless of the number of injection channels. This is because the injection restriction provided by the HOLB at the injection queue is enough to avoid network entering in saturation [9].



(a) (b)

Figure 7. Performance of Bubble (B) vs. Classic (Cl) routing under uniform (U) traffic and different network sizes (8x8, 16x16, 32x32), using IPR: (a) results for 1 injector, and (b) results for 4 injectors.

For medium-size networks (16x16) and 1 injection channel, injection restriction still is not necessary; however, the performance of the Classic network with 4 injectors drops drastically—unless traffic restriction is used. Bubble also benefits from restrictions, but as the escape sub-network suffers lower congestion, due to constrains set by the bubble condition [9], the improvement obtained for Bubble is not as noteworthy as in the Classic case. As happened with smaller networks, HOLB is enough to keep congestion under control—unless we reduce its effect by using multiple injectors, situation in which injection restriction techniques are necessary.

The large-size network (32x32) exposes that network saturates even with 1 injection channel, but performance degradation can be much worse when using 4 injection channels. Both Bubble and Classic networks benefit from injection restriction, although Classic, with suffers more from congestion (as its large completion time for the base case shows) improves in a more spectacular way than Bubble.

Figure 8 shows the Classic router performance (for a network of 32x32 nodes, using 4 injection channels) for all traffic patterns; it can be compared directly with Figure 4, for its Bubble counterpart. The plots show that, in large networks, reasonable performance can be obtained from a Classic router *only* if we tightly control injection. Note that the effect of the parameter P on completion times is more steep that for Bubble. This is because congestion builds up first in the adaptive sub-network. As the Classic router has only one adaptive virtual channel, congestion is higher and the chances to inject a new packet for a given value P are lower than in its bubble counterpart.

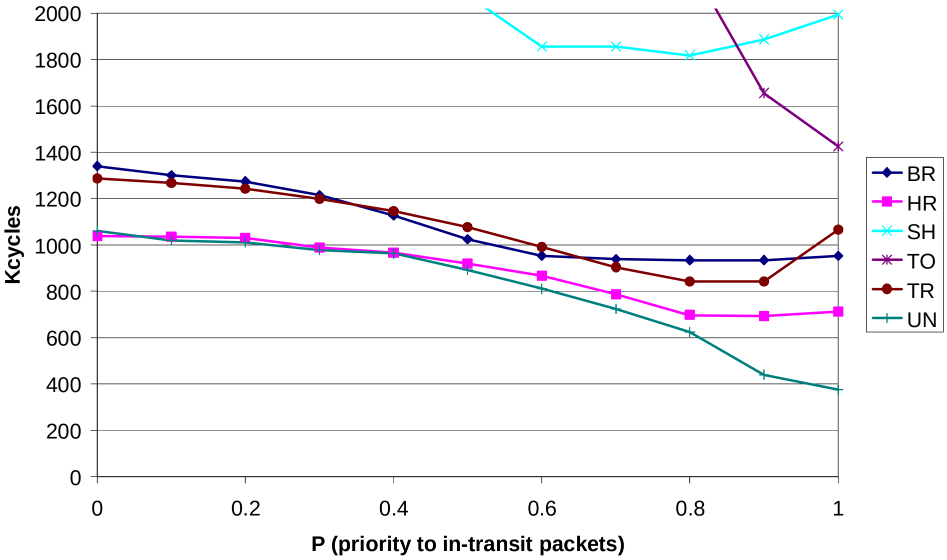


Figure 8. Effect of IPR restriction in a Classic network for different traffic patterns[5].

[5] Data for SH traffic range from 2.5 to 1.8 Mcycles. Data for TO traffic range from 2.6 to 1.4 Mcycles.

For the sake of completeness, Figure 9 compares performance obtained with both routers, with and without IPR, for a 32x32 network. Best results are always obtained when applying IPR with priority values for in-transit traffic P > 0.8. Actually, in most cases, the optimum value of P is 1.0.

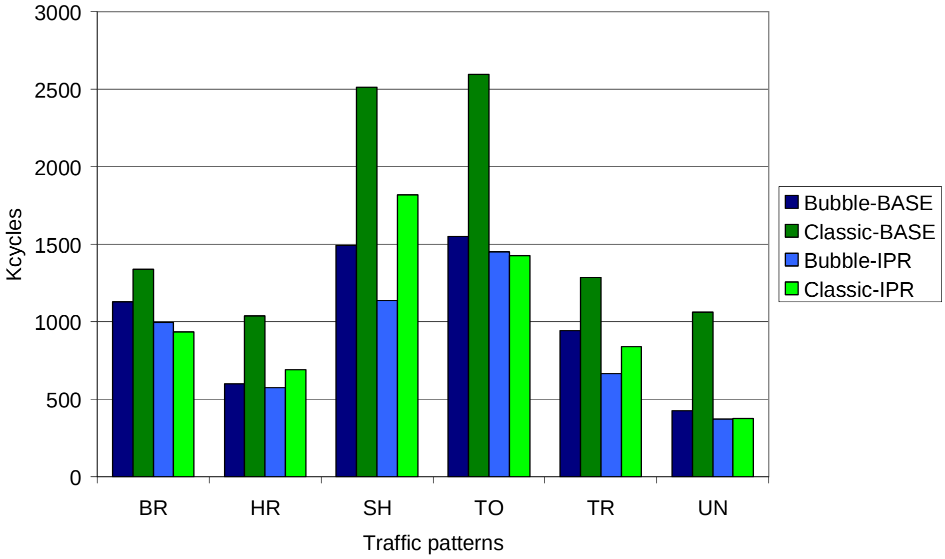


Figure 9. Comparison of Bubble and Classic routing without/with a local injection-restriction mechanism (IPR) for different traffic patterns

The conclusions of this section are that injection-restriction mechanisms are *always* required to keep congestion under control, independently of the router architecture in place. In some particular circumstances, such as small networks with a small number of injectors, the head-of-line blocking may provide enough congestion control—thus hiding this issue. However, none of these circumstances concur in current, state-of-the-art interconnection networks. In general, Classic networks are more prone to congestion than Bubble networks: for all the traffic patterns considered, the performance of Bubble without congestion control is higher than that of Classic. When IPR is applied at the maximum level (P=1), performance of Classic and Bubble are similar for most patterns, with Bubble showing itself as a clear winner for some permutations.

5 Conclusions and future work

Many studies of interconnection network performance have been carried out using small networks, and limited node-to-router bandwidth. While valid in the past, these scenarios are not any more representative of current machines, which incorporate networks with thousands of nodes and have several injection channels per node. The ability to work with larger networks has brought to light an issue many times hidden in past studies: significant performance drops when network traffic goes beyond the saturation point. This effect was not visible because small networks saturate at higher loads than large networks, and also because the head-of-line blocking at the single injector provided a rudimentary mechanism of injection control. In current networks, HOLB is not enough to keep network utilization inside operational limits, and the implementation of an explicit congestion control mechanism (in the form of injection restriction) must be put in place.

In this paper, we have shown that restrictive injection mechanisms eliminate performance degradation for loads beyond saturation, keeping adequate levels of throughput at high loads. Those methods can be local or global, utilization or priority based. Best results are obtained with global methods when network status is immediately available to all routers—a clearly unrealistic scenario. Local mechanisms are much cheaper to implement, and still offer good performance levels. Of those, a priority-based mechanism (giving priority to in-transit packets against new injections) is the easiest to tune: a large value of the in-transit priority works well almost all the traffic patterns we have studied.

We are not aware of similar performance studies, in terms of scope and results. However, the most powerful of current multicomputers, IBM's BG/L, incorporate a set of design choices for its torus network that are compatible with our findings: utilization low-radix network (using a 3D torus to reach the desired number of nodes), virtual channel management using adaptive bubble routing, and implementation of a local, priority based restrictive injection mechanism. Now, we can state that these choices are well justified in terms of their effectiveness in keeping congestion under control.

For the future, we plan to extend our studies to 3D networks of large radix (32x32x32 or even larger) to better understand the behavior of systems with many thousands of nodes. Another line of work will be to extend this analysis to real workloads.

6 Acknowledgements

This research 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.

7 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] E. Baydal and P. López. “A Robust Mechanism for Congestion Control: INC”. In Proc. Euro-Par 2003. Klagenfurt (Austria), 26-29 Aug. pp 958-968.

[3] 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.

[4] T. Callahan and S.C. Goldstein, “NIFDY: A Low Overhead, High Throughput Network Interface”, in Proc. 22nd Annual Int. Symp. on Computer Architecture (ISCA), June 2005, Santa Margherita Ligure, Italy.

[5] 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.

[6] W.J. Dally, C.L. Seitz, “The Torus Routing Chip”. Distributed Computing, vol 1 pp. 187-196. 1987

[7] W.J. Dally, B. Towles. “Principles and Practices of Interconnection Networks”. Morgan-Kaufmann, 2004.

[8] 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.

[9] C. Izu, J. Miguel-Alonso, J.A. Gregorio. "Packet Injection Mechanisms and their Impact on Network Throughput". Technical report EHU-KAT-IK-01-05. Department of Computer Architecture and Technology, The University of the Basque Country. At http://www.sc.ehu.es/acwmialj/papers/ehu_kat_ik_01_05.pdf

[10] C. Izu, J. Miguel-Alonso, J.A. Gregorio. "Traffic Sources and their Influence on the Evaluation of Interconnection Network Performance". Technical report EHU-KAT-IK-03-05. Department of Computer Architecture and Technology, The University of the Basque Country. At http://www.sc.ehu.es/acwmialj/papers/ehu_kat_ik_03_05.pdf

[11] C. Izu, C. Carrion, J.A. Gregorio and R. Beivide. “Restricted Injection Flow Control for k-ary n-cube Networks” In Proc. of the ISCA 10th Int. Conf. on Parallel and Distributed Computing Systems (PDCS'97), New Orleans, October 1997, pp. 511-518.

[12] 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.

[13] 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, 2001

[14] F. Petrini, D. Kerbyson and S. Pakin. “The Case of the Missing Supercomputer Performance: Achieving Optimal Performance on the 8,192 Processors of ASCI Q”. In IEEE/ACM SC2003, Phoenix, AZ, November 2003.

[15] 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.

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

[17] J. Shin, T.M. Pinkston. “The Performance of Routing Algorithms under Bursty Traffic Loads”. Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Applications. Las Vegas (USA), Jun. 2003.

[18] SMART group at the U. of Southern California. FlexSim 1.2. Available at http://ceng.usc.edu/smart/FlexSim/flexsim.html

[19] M. Thottethodi, A.R. Lebeck, S.S. Mukherjee. “Exploiting Global Knowledge to Achieve Self-Tuned Congestion Control for K-Ary N-Cube Networks”. IEEE Trans. on Parallel and Distributed Systems, Vol. 15, No. 3, March 2004, pp 257-272.

[20] B. Towles and W.J. Dally, "Worst-case traffic for oblivious routing functions," in *Proceedings of the ACM Symposium on Parallel Algorithms and Architectures*, pp. 1–8,Winnipeg, Manitoba, Canada, August 2002.