Integrated Admission and Congestion Control for QoS Support in Clusters *

Ki Hwan Yum Eun Jung Kim Chita R. Das
Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802
{yum,ejkim,das}@cse.psu.edu

Mazin Yousif
Advanced Component Division
Intel Corporation
Hillsboro, OR 97124
mazin.s.yousif@intel.com

José Duato
Department de Informática de Sistemas y Computadores
Universidad Politécnica de Valencia
46071- Valencia, Spain
jduato@gap.upv.es

Abstract

Admission and congestion control mechanisms are integral parts of any Quality of Service (QoS) design for networks that support integrated traffic. In this paper, we propose an admission control algorithm and a congestion control algorithm for clusters, which are increasingly being used in a diverse set of applications that require QoS guarantees. The uniqueness of our approach is that we develop these algorithms for wormhole-switched networks. We use QoS-capable wormhole routers and QoS-capable network interface cards (NICs), referred to as Host Channel Adapters (HCAs) in InfiniBand™ Architecture (IBA), to evaluate the effectiveness of these algorithms. The admission control is applied at the HCAs and the routers, while the congestion control is deployed only at the HCAs.

Simulation results indicate that the admission and congestion control algorithms are quite effective in delivering the assured performance. The proposed credit-based congestion control algorithm is simple and practical in that it relies on hardware already available in the HCA to regulate traffic injection.

1 Introduction

Clustering servers is a cost-effective approach in designing scalable and high performance computers that can support various scientific and commercial applications. Given the variety and sophistication of services such as transfer of dynamic Web pages, multimedia objects, and e-commerce transactions offered by such applications, built-in Quality of Service (QoS) support in clusters is becoming critical.

The traditional *same-service-to-all* or *best-effort* model does not inherently possess enough granularity to provide customized services to these applications. The Internet Engineering Task Force (IETF), realizing the limitations of the *best-effort* model, has undertaken serious steps to meet the QoS demand in the Internet infrastructure. Since web/compute/database servers typically makeup the back-end infrastructure in *datacenters*, it is imperative to provide similar capabilities in clusters, which are being increasingly used as back-end machines. This is true, especially after the introduction of new cluster interconnects such as InfiniBand™ Architecture (IBA) [7]. Providing QoS guarantees in IBA has become a current research focus, as well as a major commercial interest [11].

QoS in networks can be provided by customized allocation of scarce network resources through policy rules. Two supplementary mechanisms are required to facilitate QoS guarantees. First, an appropriate scheduling technique is required to allocate link bandwidth to applications as per their requirements, while ensuring that all other applications get a fair share of bandwidth. Scheduling schemes such as Weighted Round Robin (WRR) [8], Fair Queuing [18], and VirtualClock [19] have been proposed for such proportional bandwidth allocation in packet-switched networks. Second, an admission control mechanism is required to regulate the number of users in the system such that the required QoS constraints of applications can be satisfied. This algorithm is a key component of any QoS policing mechanism since it determines the extent of resource utilization while delivering the promised QoS performance.

Admission control algorithms help to meet Service Level

*This research was supported in part by NSF grants MIPS-9634197, CCR-9900701, and CCR-0098149, and equipment grants from NSF and IBM.

Agreements (SLAs) of real-time applications. However, admission control alone may not be effective enough to guarantee the SLAs of real-time and best-effort applications because they may exhibit unpredictable behavior, resulting in short- or medium-term network traffic overload. Such traffic overload considerably degrades overall network throughput. Therefore, a congestion management algorithm is typically used to monitor the network load, and intervene when the traffic load reaches a certain threshold indicating possible network congestion. Since a congestion management scheme also brings its own set of constraints on the injection of traffic flows into the network, both admission control and congestion management are collectively needed to guarantee various QoS constraints. This is especially true in clusters running a diverse set of applications.

The focus of the paper is on the design of admission control and congestion management algorithms to supplement a wormhole-routed cluster interconnect for achieving both high and predictable performance. The motivation for using wormhole switches is that they have been adopted extensively in designing clusters [2, 6, 14] because of their ability to provide high performance. We develop the admission and congestion control mechanisms using the wormhole router fabric proposed in [16]. However, unlike the Network Interface Card (NIC) design of [16], we emulate a Host Channel Adapter (HCA) as proposed in the IBA framework to study the network interface (NI) performance.

The main contributions of the paper are the following:

- We develop a simple admission control algorithm to decide on the admission of real-time applications. The proposed mechanism is orthogonal to the router and NIC design and helps in further reducing the Deadline Missing Probability (DMP) and the average Deadline Missing Time (DMT) of real-time applications.

- Next, we propose a novel and practical congestion management scheme using the concept of credit-based flow control. This congestion management algorithm, called *Credit-Based Congestion Control*, uses the Completion Queue (CQ) in the HCA to determine the traffic load in the network.

- We evaluate end-to-end QoS guarantees in clusters by integrating four main components: the admission control scheme, the congestion control scheme, the QoS-aware HCA and the QoS-aware network. Such a comprehensive study has not been undertaken in any prior research.

We develop a detailed simulator integrating the cluster interconnect (routers and HCAs) and the admission and congestion control algorithms. We use a mixed workload consisting of three types of traffic — short control messages, best-effort traffic, and real-time traffic (MPEG-2 video traces and ON/OFF sources).

Simulation results show that both the schemes help in providing very low and stable DMP and DMT for MPEG-2 streams over the entire workload. For the ON/OFF and best-effort traffic, the combined control mechanisms minimize average message latency significantly as the load increases. In summary, performance is the best with an integrated admission and congestion control, while admission control is more effective at lower load and congestion control is more effective at higher load.

Another advantage of the proposed *Credit-Based Congestion Control* algorithm is that it can be implemented using the hardware already available in the HCA. Moreover, our scheme can perform selective/per flow control and is shown to provide better performance than two recently proposed congestion control schemes [1, 15]. Although the admission and congestion control schemes are discussed in the context of wormhole networks, they should be applicable to packet-switched networks.

2 Related Work

Admission Control: An admission control algorithm determines whether a new real-time traffic flow can be admitted to the network without jeopardizing the performance guarantees given to the already established flows. Such an algorithm is essential irrespective of the underlying communication architecture to regulate the traffic flow. Admission control in packet-switched networks has been a rich area of research. There are two broad classes of admission control algorithms: deterministic and statistical admission control.

For real-time services that need a hard or absolute bound on the delay of every packet, a deterministic admission is used [5]. For such deterministic services, an admission control algorithm calculates the worst-case behavior of existing flows in addition to the incoming one before deciding if the new flow should be admitted. This model underutilizes network resources, especially with traffic burst.

Many of the new applications such as the media streams do not need hard performance guarantees and can tolerate a small violation in performance bounds. A statistical admission control scheme can be used for such applications. In this approach, an effective bandwidth that is larger than the average rate but less than the peak rate is commonly used. The bandwidth can be computed using a statistical model [12] or a fluid flow approximation [9].

For admission control in clusters, the MMR design uses the average and peak rates of requests [3]. However, this router uses PCS for real-time traffic and needs one virtual channel (VC) per connection (flow). The Switcherland router [4], based on the ATM protocol, uses a statistical admission algorithm. A flit reservation flow control scheme that uses control flits to reserve bandwidth and buffers prior

to the transfer of data flits has been proposed recently [10]. To our knowledge, there is no prior work on admission control in wormhole-switched networks.

Congestion Control: Congestion control is required to regulate traffic injection into a network to avoid network saturation, which may lead to performance penalty. In networks with QoS guarantees, congestion control mechanisms first attempt to regulate best-effort and misbehaving real-time traffic, and if required, then traffic from other service classes. In wormhole-switched networks, prior work on congestion control tends to limit message injection rate in each node when a specified network saturation point is reached [1, 13, 15]. Local or global information could be used to determine network saturation. For example, Lopez et al. [1] used the busy/free status of VCs to assess network congestion. Smai and Thorelli [13] counted on the global network state to detect network congestion. To achieve a global view of the network, each node communicates its traffic status with other nodes, which may lead to excessive communication overhead. Thottethodi et al. [15] suggested a self-tuned approach that determines appropriate threshold values to estimate network congestion.

Previous congestion control algorithms for wormhole-switched networks do not provide an end-to-end congestion control. They only consider the network/router status, not the NI, which is closer to the applications. Moreover, instead of penalizing the flow that caused congestion, a uniform reduction rate is typically applied to all the flows that pass through the congested point. Ideally, it should provide selective congestion control per flow/application as is done in the Internet TCP flow control. The proposed algorithm has this selective control ability.

3 Architecture

In this section we describe the cluster interconnect. It includes a QoS-capable router architecture, the Host Channel Adapter (HCA), and a rate-based scheduling algorithm, called VL arbitration, used in the router and the HCA.

3.1 Router Architecture

Since the focus of this paper is on admission and congestion control, we are using one of the recent router models proposed in [17]. The first stage of the pipelined wormhole router shown in Fig. 1 represents the functional units, which synchronize the incoming flits, demultiplex a flit so that it can go to one of the C virtual lanes (VLs)[1] to be subsequently decoded. If the flit is a header flit, routing decision and arbitration for the correct crossbar output is performed in the next two stages (Stage 2 and Stage 3), while middle flits and the tail flit of a message directly move to Stage 4. Flits get routed to the correct crossbar output ports in Stage 4. The router has a scheduler (arbiter/multiplexer) at the input port of the crossbar. In the traditional *best-effort* model, the scheduler can select one of the C VLs using FCFS or Round Robin (RR) principle. Finally, the last stage of the router performs buffering of flits flowing out of the crossbar, multiplexes the physical channel bandwidth amongst the C VLs, and carries out synchronization with input buffers of other routers or the network interface for the subsequent transfer of flits.

The VLs are statically assigned to different traffic classes during initial system configuration. A traffic class is allowed to use only the VLs assigned to it. To provide proportional bandwidth allocation, we decided to adopt WRR scheme at Stage 4 and 5, since WRR scheme has been used in many commercial routers and it is compatible with the IBA specification.

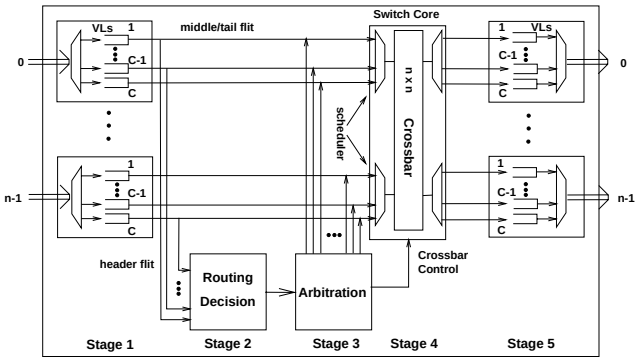


Figure 1. A 5-Stage Pipelined Router Model

3.2 Host Channel Adapter (HCA) Architecture

The importance of an NI in minimizing communication overhead is well documented in the literature. In a recent study [16], it was shown that QoS provisioning in the NIC is essential to transfer the benefits of the network to the application level. Therefore, we also design a QoS-capable NIC to analyze the entire communication substrate. Here we use the Channel Adapter (CA) specification of IBA and modify it to provide QoS in the NIC.

The Host Channel Adaptor (HCA) architecture proposed in IBA is shown in Fig. 2. A consumer (abstracted from an application) creates one or more Queue Pairs (QPs) and one or more Completion Queues (CQs) in a CA. A QP actually consists of two queues: one for sending messages and another for receiving messages. The consumer creates a Work Request (WR), which when passing through the IBA software stack gets converted to a Work Queue Element (WQE). The WQE subsequently gets deposited into a QP sending queue. Then the following sequence of events take place: the CA processes the WQE; the DMA engine in the CA transfers data from the host memory to one of the

[1] Virtual lanes as used in the InfiniBand terminology, and virtual channels are synonymous. We use both terms interchangeably in this paper.

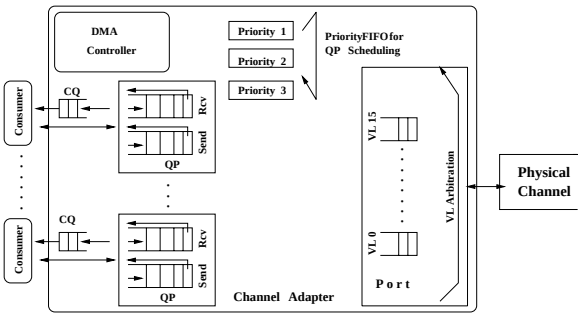


Figure 2. A Proposed HCA with QoS Support

Virtual Lanes (VLs) of the CA's port; then data gets pushed into the network. A WRR scheme is also used for arbitrating the VLs in the HCA port. When the CA completes executing a WQE, it places a Completion Queue Element (CQE) in its CQ. The sequence of events on the receiver and sender sides are similar [7]. Note that since the HCA is assumed to use the system bus instead of the standard PCI bus, only one DMA is required to bring data from the user memory to a VL in the HCA. (With a PCI bus interface, this transfer requires two DMAs: one for WQE transfer and the other for data transfer.)

In IBA, a CA may implement up to 16 VLs. $VL_0 - VL_{14}$ are referred to as Data VLs and are used for data transfer; VL_{15} is referred to as the management VL and is dedicated to control traffic. We extend the IBA framework to include a prioritized QP scheduling structure to support prioritized traffic transfer. This is similar to the prioritized doorbell scheme in the VIA domain [16]. In this scheme, there is a queue for each traffic class. An application posts a message in the appropriate queue after inserting the WQE in its QP. The CA firmware decides which QP to service in FCFS order based on their priority (traffic class) and programs the host DMA engine to transfer the message to the appropriate VL in the HCA port. Messages of the same class do not get reordered in this scheme. This prioritized QP scheduling helps in transferring the higher priority messages first to the VLs, where they are scheduled using the WRR scheme to be pushed to the network.

To make the CA design compatible to the QoS-aware router in Section 3.1, we implemented in the CA buffer an equal number of (C) VLs to enable virtual channel flow control in the CA. As messages are transferred into the CA by the host DMA, they are broken into *flits*. The CA buffer behaves as FCFS queues for the different VLs. The flits are injected into the network at the rate of one flit per cycle.

3.3 VL Arbitration

VL arbitration or scheduling refers to the selection of an outgoing link of a switch/router or channel adapter. In a multiplexed crossbar router implementation, we also need the arbiter at the input port of the crossbar (in stage 4 of the pipeline of Fig. 1). The arbiter selects the flit to transmit from the set of candidate flits competing for the same port.

IBA specifies a two-level scheme for VL arbitration. First, all the applications are classified into different priority classes and a priority scheduling is used for scheduling different classes. Next, a WRR scheme is used to schedule traffic of the same class. Additionally, the scheme provides a method to ensure forward progress of the low-priority VLs. Also, the weight calculation, prioritization, and minimum forward progress bandwidth should be programmable.

4 Admission and Congestion Control

4.1 Admission Control

The admission control algorithm decides whether a new real-time connection request should be accepted or rejected. Before a real-time traffic source starts its data transmission, it sends a probe packet to the destination. The probe packet includes the routing information and the solicited bandwidth. The first admission control check is performed at the corresponding HCA. If accepted, the solicited bandwidth of the request is added to the total currently used bandwidth of the physical link; then the probe packet is forwarded to the connected switch/router. If rejected, a NACK is sent back to the traffic source without changing the used bandwidth of the physical link.

Upon receiving the probe packet, each router tests the link bandwidth of the destination port for the packet to decide whether the link has sufficient bandwidth. If accepted, the router checks the destination node of the packet. If the destination is the same as the address of the present router, an ACK message is sent back to the source. Otherwise, the router forwards the probe packet to the next router using the underlying routing algorithm and destination address. In both cases, the requested bandwidth of the request is added to the total used bandwidth of the destination physical link and to that of the incoming physical link, where the probe packet resides. In addition, weight calculation for the WRR scheduling is also performed. After receiving the ACK message, the source starts to send its data packets.

If the request is rejected in the router, a NACK message, which also includes the address of the router that rejected the request, is sent back to the source. This NACK message travels back to the source using the underlying routing algorithm, which means that the forward and returning paths could be different. If the NACK message uses the same path as the probe packet in the reverse direction, there could be deadlock in wormhole routers. After receiving a NACK message, the source sends a release message that includes the same routing information, bandwidth requirement, and the address of the router that rejected the request. Each HCA or router that receives the release message carries out

the restoration procedure where the required bandwidth is subtracted from the total used bandwidth of the physical link(s) and the weights for WRR are recalculated. Then the router compares the address of the node that rejected the request in the release message with that of the neighboring router to decide if the release message should be sent further. If they are the same, it implies that the neighboring router had initiated the rejection, and so there is no need to send the release message further. Otherwise, it forwards the message to the neighboring router until all the reservations are released. Our simple scheme avoids deadlock by sending the release message from the source node, but incurs additional latency.

When the source finishes data transmission, it inserts the same bandwidth requirement that was used for connection setup in the header of the final data packet. The HCA and switches/routers release the reserved resources as this packet goes through them.

Bandwidth reservation using a probe packet is not a new concept. It is the best known scheme for providing hard QoS guarantees and has been used in packet-switched networks. What is new in this paper is how do we establish and tear down reservations in a wormhole-switched network. Such a reservation may not be required for statistical soft guarantees, which can be done using a QoS-aware network architecture [17, 16].

4.2 Congestion Control Algorithm

Network congestion happens when more traffic is injected into the network than what the network resources can handle. The aim of any congestion control algorithm is to detect congestion occurrence at inception or as early as possible and then take appropriate corrective action. However, early congestion detection is extremely difficult, and possibly not reliable due to unpredictable traffic behavior. It seems that there are no well-accepted congestion control mechanisms due to various limitations.

Our goal for congestion control in clusters here is to regulate the injection rate of traffic sources according to the status of the available network resources. Therefore, we propose a congestion prevention mechanism. Note that unlike the Internet congestion control, we do not allow dropping packets in the network, since packet dropping is extremely complex in the wormhole-switched network. If the network resources are not available, packets can be dropped or backlogged at the injection points. The key issue is then how to find the status of the network resources at the HCA.

IBA specifies link-level credit-based flow control. The same scheme is also used in wormhole switching. This scheme can be implemented using relatively small size buffers, and hence the flow control information can be propagated faster. The flow control traverses backward up to the HCA of the source node.

In the HCA design as described in Section 3.2, a Completion Queue Element (CQE) is deposited in the completion queue (CQ) of the sender. It is possible to interpret a CQE as a credit to send a message to the HCA, which in turn implies that the network should be able to accept the message. If a consumer is allowed to inject messages into the HCA equal to the number of CQEs, congestion will not occur in the network We call this scheme *Credit-Based Congestion Control*. The protocol is simple and practical in that there is no need of any extra hardware for implementing it since the CQ is a part of the HCA. What is required is a judicious selection of the number of initial credits.

We need a certain number of initial credits at each HCA to start message injection into the network. This number could be different for each traffic class. Let C_i be the number of initial credits in the CQ for consumer i. Then, the first C_i messages of a consumer i can be injected into the HCA without any constraint. After that, source i can inject additional messages into the HCA, only after the HCA has injected messages into the network, and has returned credits to the CQ. Therefore, in the steady state, the injection rate of a consumer i will be equal to the incoming rate of credits to the completion queue. With this approach, the traffic burst can be controlled with the number of initial credits (C_i).

It is important to assign proper initial credits to each consumer. Further, it should be obvious that the total number of initial credits cannot exceed the size of the buffer (M) in the HCA ($\sum_{\forall i} C_i \leq M$). For each real-time connection, initial credits are given according to the bandwidth requirement (b_j) as follows: $C_j = \frac{b_j}{B} \cdot M$ where B is the channel bandwidth. Since best-effort traffic does not have any specific bandwidth requirement, we heuristically assign the initial credits for best-effort traffic(C_b) such that $C_b \leq M - \sum_j C_j$.

Credits are generated by the HCA and consumed by a consumer. If the consumer does not have a credit in its CQ to send a message, it has the two options. Either, the consumer waits until a credit is available in the CQ; or it drops the message and retries posting the message later. The former approach is used to handle congestion control for best-effort traffic; for real-time traffic, the latter approach is used. In our implementation, we assume that *Credit-based Congestion Control* does not restrict the injection of control traffic, which has the highest priorty.

5 Experimental Platform

5.1 Simulation Testbed

For our experiment, we simulated an 8-port router connected with HCAs and a 2-D mesh network designed using 5-port routers and HCAs. We used 16 VLs per PC as has been proposed in the IBA specification. The flit size is 128 bits, and each message consists of 40 flits except

for the control messages, which are 10-flit long. Physical link bandwidth is 1.6Gbps, and flit buffers are 10-flit deep. Note that there is a difference in the packet size between our simulator and the IBA specification. Since our interest here is to explore the feasibility of QoS support in wormhole-switched networks, we are using parameters compatible with recent router design.

5.2 Workload

Our workload includes messages from real-time VBR traffic or ON/OFF traffic, best-effort traffic, and control traffic. The VBR traffic from real MPEG-2 video traces is generated as a stream of messages between a pair of communicating processors. Since the simulation with MPEG-2 traces is extremely time consuming, we also use an ON/OFF source to simulate real-time traffic. The ON/OFF traffic is generated as a stream of messages between a pair of source and destination nodes. During the OFF period, the source does not generate any messages, while during the ON period, messages are generated according to the given injection rate λ_{onoff}. To avoid traffic burst, the generation is evenly scattered.

The best-effort traffic is generated with a given injection rate λ_{be}, and follows the Poisson distribution. Best-effort messages are assumed 40-flit long, and a destination is picked using a uniform distribution. The input and output VLs for a message are assigned using a uniform distribution of the available VLs. Control traffic is typically used for network configuration, congestion control, and transfer of other control information. This traffic has the highest priority in our model. We assume the generation rate of control traffic is very low (for example, ten messages per 33.3 msec of simulation with MPEG-2 traffic) and VL_{15} is assigned for this traffic.

The important output parameters measured in our experiment are the Deadline Missing Probability (DMP) of delivered MPEG-2 frames, the average Deadline Missing Time (DMT) of deadline missing frames, and average network latency for best-effort, control, and ON/OFF traffic. The DMP is the ratio of the number of frames that missed their deadlines to the number of total number of delivered frames. The deadline for each frame is determined by adding 33.3 msec to the previous deadline, since the frame rate is 30 frames/sec for MPEG-2 video streams. However, if a previous frame missed its deadline, a new deadline is set by adding 33.3 msec to the arrival time of the previous frame. Whenever a frame misses its deadline, we measure the delay and then calculate the DMT.

6 Performance Results

Most of the performance results are presented for a real-time to best-effort ratio of 80:20. Only a selected set of results are presented due to space limitation.

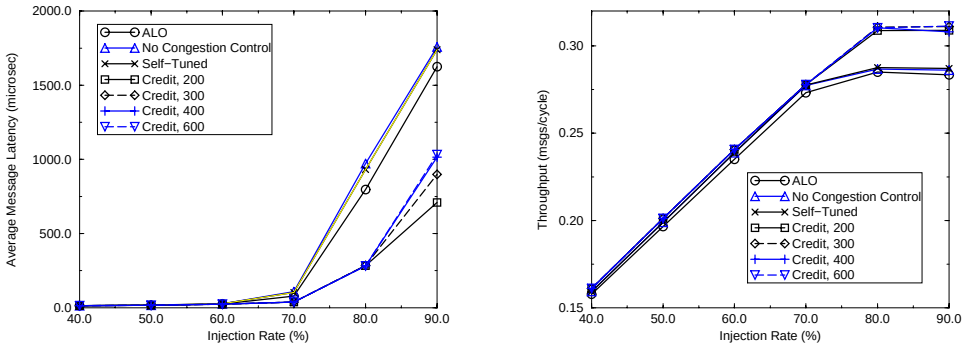
6.1 Comparisons of Congestion Control Algorithms

We simulated the prior At-Least-One (ALO) [1] and the Self-Tuned [15] congestion control schemes to compare with our scheme. In the ALO congestion control, the global network congestion is estimated locally at each node. If at least one VL is free in every useful physical channel or if at least one physical channel has all its VLs free, then the packet injection is allowed. Otherwise the new packets are throttled. The Self-Tuned congestion control technique uses the global knowledge of the number of *full* network buffers to estimate the network congestion. We use the same parameters given in [15] to simulate the scheme. Since these two schemes were developed for the network only, they only monitor the buffer status in the router. We modify these schemes to include the status of the HCA buffer. We assume that an exclusive side-band is used for communicating the congestion and throughput information in the Self-Tuned scheme, because in-band communication using VL_{15} would worsen the results.

Fig. 3 shows latency and throughput variation of the congestion control schemes in a 4×4 mesh network. We have simulated the credit-based congestion control scheme with four different initial credits. Since the ALO and the Self-Tuned schemes used best-effort traffic for their results, we compare the schemes with only best-effort traffic. The network without any congestion control exhibits the lowest performance in terms of latency and throughput. In general, the credit-based congestion control scheme is capable of providing lower latency and better throughput than the ALO and the Self-Tuned schemes for the entire load. Especially, the improvements become more evident at higher load. The number of initial credits affects the message latency, and the results show that we get the best performance with 200 initial credits. As expected, higher number of initial credits injects more traffic into the network and thus increases delay. Unlike [1, 15], the existence of an HCA buffer prevents a sudden throughput drop without any congestion control in our study, since the source cannot inject messages when the buffer is full.

6.2 Results with Admission and Congestion Control

Fig. 4 (a) plots the Deadline Missing Probability (DMP) and the Deadline Missing Time (DMT) of a single router cluster with uniform traffic. Also the average latency of control and best-effort traffic is plotted in Fig. 4 (b). In the figures, **A,C** indicates a router with both admission and congestion control, and **No A, No C** implies a router without admission and congestion control. It is seen that the DMP and the DMT remain very small with admission and congestion control over the entire workload. (The DMP is only 0.002 and the DMT is around 0.04 msec.) The DMP and



(a) Average Message Latency

(b) Throughput

Figure 3. Message Latency and Throughput in a 4×4 Mesh Network with 100% Best-effort Traffic

the DMT values without admission and congestion control are higher and unstable. Note that the cluster without admission and congestion control is simulated with controlled injection rates of 60, 70, and 80%. This is an implicit input control. In a real environment, the input rates are not controlled and therefore, the DMP and the DMT values will be much higher without these mechanisms. Fig. 4 (b) indicates that the average message latency with these control mechanisms is smaller for both control and best-effort traffic. In particular, the best-effort traffic latency is orders of magnitude smaller.

Fig. 5 shows the effect of admission control and congestion control in a 5×5 mesh network with ON/OFF real-time traffic. The general trend in all these graphs is that the performance is the best with both admission and congestion control, followed by only admission control, and then congestion control only at low load. However, as the load increases, the performance of the network with *only* congestion control becomes better than that of the network with *only* admission control. This is because the network is congested at higher load even with admission control. The effect of admission control is less prominent for control traffic, since control traffic has higher priority than real-time traffic. On the contrary, as shown in 5 (b), admission control plays a major role for QoS assurance in real-time traffic. All the results emphasize one point clearly: admission and congestion control are essential to provide QoS assurance for all traffic classes. The results are the worst without any of these control mechanisms.

7 Concluding Remarks

This paper presents admission and congestion control mechanisms for wormhole-routed cluster interconnects to provide QoS guarantees in clusters. While QoS in clusters has become a recent research focus, and the IBA Trade Association has defined a generic QoS specification, there is no unified work for regulating QoS parameters in wormhole routed networks that are currently used in many clusters. We believe that our work makes a significant contribution in this aspect. Moreover, the algorithms are equally applicable to other networked architectures.

The important conclusions of this work are the following: First, the admission control algorithm, which uses a probe packet to reserve channel bandwidth prior to sending message flits, guarantees MPEG-2 stream delivery with a very small and stable DMP over the entire workload as compared to a cluster without any admission control. Second, the *Credit-Based Congestion Control* algorithm effectively administers the injection of flits into the HCA, and thus provides better throughput and lower message latency compared to two former schemes. Further, since its implementation is simple and requires no additional hardware, our approach is commercially attractive. Finally, an integrated admission and congestion control mechanism can provide significant performance improvement resulting in better QoS guarantees.

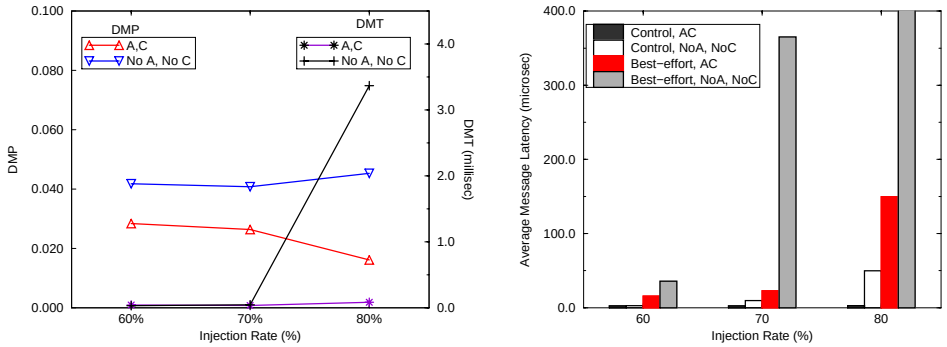
References

[1] E. Baydal, P. Lopez, and J. Duato. A Simple and Efficient Mechanism to Prevent Saturation in Wormhole Networks. In *Proc. of Intl. Parallel and Distributed Processing Symp.*, pages 617–622, 2000.

[2] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic, and W.-K. Su. Myrinet: A Gigabit-per-second Local Area Network. *IEEE Micro*, 15(1):29–36, February 1995.

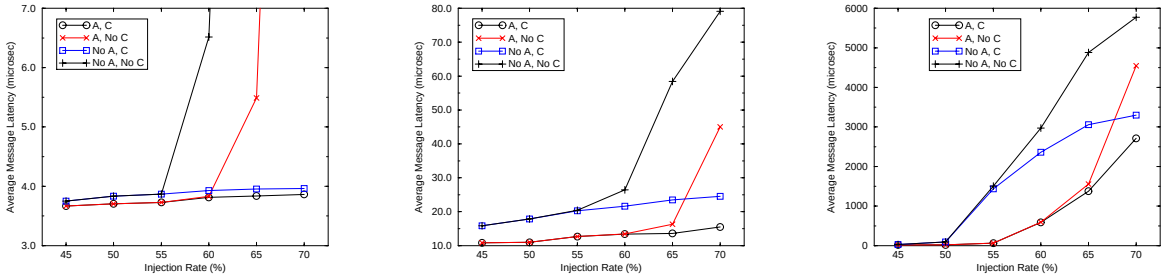
[3] J. Duato, S. Yalamanchili, M. B. Caminero, D. Love, and F. J. Quiles. MMR: A High-Performance Multimedia Router-Architecture and Design-Tradeoffs. In *Proc. of HPCA*, pages 300–309, January 1999.

[4] H. Eberle and E. Oertli. Switcherland: A QoS Communication Architecture for Workstation Clusters. In *Proc. of ISCA*, pages 98–108, June 1998.



(a) DMP and DMT (b) Average Message Latency

Figure 4. Performance Results of a Single Router Cluster with MPEG-2 Video Traffic



(a) Control Traffic (b) Real-time Traffic (c) Best-effort Traffic

Figure 5. Average Message Latency in a 5×5 Mesh Network with On/Off Real-time Traffic

[5] D. Ferrari and D. C. Verma. A Scheme for Real-Time Channel Establishment in Wide-Area Networks. *IEEE JSAC*, 8(3):368–379, 1990.

[6] M. Galles. Scalable Pipelined Interconnect for Distributed Endpoint Routing : The SGI SPIDER Chip. In *Proc. of Hot Interconnects*, pages 141–146, August 1996.

[7] InfiniBand Trade Association. InfiniBand Architecture Specification, Volume 1, Release 1.0, October 2000. Available from `http://www.infinibandta.org`.

[8] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis. Weighted Round-Robin Cell Multiplexing in a General-Purpose ATM Switch Chip. *IEEE JSAC*, 9(8):1265–1279, October 1991.

[9] F. P. Kelly. Effective Bandwidths at Multi-Class Queues. *Queueing Systems*, 9:5–16, 1991.

[10] L.-S. Peh and W. J. Dally. Flit-Reservation Flow Control. In *Proc. of HPCA*, pages 73–84, January 2000.

[11] J. Pelissier. Providing Quality of Service over InfiniBand Architecture Fabric. In *Proc. of Hot Interconnects*, August 2000.

[12] H. Saito and K. Shiomoto. Dynamic Call Admission Control in ATM Networks. *IEEE JSAC*, 9(7):982–989, September 1991.

[13] A. Smai and L. Thorelli. Global Reactive Congestion Control in Multicomputer Networks. In *Proc. of Intl. Conf. on High Perf. Computing*, pages 179–186, 1998.

[14] C. B. Stunkel, D. G. Shea, B. Abali, M. G. Atkins, C. A. Bender, D. G. Grice, P. Hochschild, D. J. Joseph, B. J. Nathanson, R. A. Swetz, R. F. Stucke, M. Tsao, and P. R. Varker. The SP2 High-Performance Switch. *IBM Systems Journal*, 34(2):185–204, 1995.

[15] M. Thottethodi, A. R. Lebeck, and S. S. Mukherjee. Self-Tuned Congestion Control for Multiprocessor Networks. In *Proc. of HPCA*, pages 107–118, January 2001.

[16] K. H. Yum, E. J. Kim, and C. R. Das. QoS Provisioning in Clusters: An Investigation of Router and NIC Design. In *Proc. of ISCA*, pages 120–129, June 2001.

[17] K. H. Yum, A. S. Vaidya, C. R. Das, and A. Sivasubramaniam. Investigating QoS Support for Traffic Mixes with the MediaWorm Router. In *Proc. of HPCA*, pages 97–106, January 2000.

[18] H. Zhang and S. Keshav. Comparison of Rate-Based Service Disciplines. In *Proc. of SIGCOMM '91*, pages 113–121, September 1991.

[19] L. Zhang. VirtualClock: A New Traffic Control Algorithm for Packet-Switched Networks. *ACM Trans. on Computer Systems*, 9(2):101–124, May 1991.