Integration of Admission, Congestion, and Peak Power Control in QoS-Aware Clusters *

Ki Hwan Yum Yuho Jin
Department of Computer Science
University of Texas at San Antonio
San Antonio, TX 78249 U.S.A.
email: yum@cs.utsa.edu

Eun Jung Kim Chita R. Das
Department of Computer Science
Texas A&M University
College Station, TX 77843 U.S.A.
email: {yuho,ejkim}@cs.tamu.edu

Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802 U.S.A.
email: das@cse.psu.edu

Abstract

Admission, congestion, and peak power control mechanisms are essential parts of a cluster network design for supporting integrated traffic. While an admission control algorithm helps in delivering the assured performance, a congestion control algorithm regulates traffic injection to avoid network saturation. Peak power control forces to meet pre-specified power constraints while maintaining the service quality by regulating the injection of packets. In this paper, we propose these control algorithms 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, which have been used in designing clusters. We use QoS-capable wormhole routers and QoS-capable network interface cards (NICs), referred to as Host Channel Adapters (HCAs) in InfiniBand^{TM} Architecture (IBA), to evaluate the effectiveness of these algorithms. The admission control is applied at the HCAs and the routers, while the congestion control and the peak power control are deployed only at the HCAs. A mixed workload consisting of best-effort, real-time, and control traffic is used to investigate the effectiveness of the proposed schemes.

Simulation results with a single router (8-port) cluster and a 2-D mesh network cluster indicate that the admission, congestion, and peak power 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/NIC to regulate traffic injection.

Index Terms: Admission Control, Congestion Control, Cluster Network, HCA/NIC, Quality of Service, Peak Power Control, Wormhole Router.

1 Introduction

Clustering servers is a cost-effective approach in designing scalable and high-performance computers that can support various scientific and commercial applications with diverse requirements. And, reversely, these applications pose significant design challenges in cluster networks in terms of Quality of Service (QoS), performance, and energy efficiency. To provide high bandwidth and low latency, clusters using switched network architectures are becoming more popular than broadcast-based networks in recent years. A cluster interconnect consisting of routers, links, and network interface cards (NICs) can be used to connect multiple PCs, multiple blades in a single server (Mellanox Nitro II [1]), multiple processors on a single board (Compaq Alpha 21364 [2]) or even multiple components on a single chip. InfiniBand^{TM} Architecture (IBA) [3], proposed as a new communication standard to design Sys-

*This research was supported in part by NSF grants CCF-0541360 and CCF-0541384. A preliminary version of this paper was presented at the IEEE International Conference on Cluster Computing, September 2002.

tem Area Networks (SANs), is a packet switch-based interconnection technology that connects processors and I/O devices directly under one unifying design.

QoS support in clusters has been studied extensively by many researchers [4, 5, 6, 7, 8, 9, 10, 11]. One of such approaches that provide QoS support in a communication network is to supply admission and congestion control to regulate the number of active connections in the network and the number of injected packets in those connections. Admission control determines the acceptance of a new connection request in the network, based on the requirement of the new connection and the current resource capacity availabe in the network. If the acceptance of the new connection jeopardizes QoS guarantees of already established connections, admission control denies the setup of the new connection. However, admission control alone may not be effective enough to guarantee Service Level Agreements (SLAs) of the 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, congestion control is typically used to monitor the network load, and intervene when the network 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.

Energy efficiency is another design challenge when building cluster networks, since it turned out that the cluster-based data centers consume significant power and the power usage is a major fraction of the total ownership cost [12]. The cost of thermal packages to manage operating temperature is also enormous in server racks of a data center [13]. [14] showed that the interconnection fabric consumes a significant portion of the total cluster power and that links are the major consumer in the cluster interconnects. Since application traffic tends to make cluster interconnects acquire more power, the total consumption of the network power may surpass the supplied power causing reliability problems [15]. To ensure power contraint satisfaction as well as high performance, therefore, cluster interconnects must have a peak power control mechanism.

The focus of the paper is on the design of admission and congestion control algorithms along with peak power control to supplement a wormhole-routed cluster interconnect for achieving both high and predictable performance. In this paper, we develop these control mechanisms using the wormhole router fabric proposed in [11]. However, unlike the NIC design of [11], 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:

- Although the Weighted Round Robin (WRR) scheduling has been used in the Internet routers and is also proposed for IBA, the actual implementation is rather intricate. This is primarily because mapping of the applications to appropriate weights depends on the selected frame size. We implement four variations of the WRR algorithm in the router and the HCA using two different frame sizes to support proportional bandwidth allocation. Performance implications of these implementations are analyzed.

- We develop a simple admission control algorithm to decide on the admission of real-time applications. The proposed admission control mechanism is orthogonal to the router and NIC design, and helps in further reducing the Deadline Missing Probability (DMP) and 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 uses the Completion Queue (CQ) in the HCA to determine the traffic load in the network.

- We propose Credit-based peak power control to meet pre-specified power constraints while maintaining the service quality, by regulating the injection of packets in the HCA. We take different approaches for different traffic types. For real-time traffic, our scheme determines the acceptance of a new connection based on the requirement of the power consumption of the connection and avaiable power budget. For best-effort traffic, we calculate power consumption of a packet based on the distance from its source to the destination. If the expected power consumption exceeds the power budget, we throttle the injection of the packet inside the HCA.

- We propose separate Completion Queue (CQ) scheme for end-to-end congestion control of best-effort traffic by providing multiple CQs in the HCA, with which we can take fine-grained control of best-effort traffic. The original scheme is to provide as many CQs as the number of nodes in the network, but we show that a Quad CQ configuration is sufficient for a mesh network by which we can reduce the overhead of implementing a large number of CQs in the HCA.

- We evaluate end-to-end QoS guarantees in clusters by integrating all the proposed techniques with 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 NICs/HCAs) and all the schemes. We use a mixed workload consisting of three types of traffic — short control messages, best-effort traffic, and real-time traffic (MPEG-2 video stream traces and ON/OFF sources). We conduct an in-depth analysis of the cluster performance using average message latency, Deadline Missing Probability (DMP) of MPEG-2 frames, and Deadline Missing Time (DMT) as the performance metrics. The first parameter quantifies performance implications for the best-effort traffic, control traffic, and ON/OFF traffic, while the other two parameters are indicators of MPEG-2 traffic behavior.

Simulation results of a single router (8-port) cluster and a 2-D mesh network cluster indicate that the integrated admission and congestion control is capable of delivering much better QoS compared to a cluster system without these control mechanisms. Specifically, both the schemes help in providing a very low and stable DMP and DMT for MPEG-2 streams over the entire workload, while the DMP and DMT values are higher and unstable without these controls. 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 [16, 17]. 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: determin-

istic 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 [18]. 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 [19] or a fluid flow approximation [20].

For admission control in clusters, the Multimedia Router (MMR) [8] uses the average and peak rates of requests. However, this router uses Pipelined Circuit Switching (PCS) [21] for real-time traffic and needs one virtual channel (VC) per connection (flow). The Switcherland router [5], 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 [22].

Congestion Control: Congestion control is required to regulate traffic injection into a network to avoid network saturation, which may lead to performance degradation. 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 [16, 23, 17]. Local or global information could be used to determine network saturation. For example, [16] used the busy/free status of VCs to assess network congestion. [23] 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. [17] 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.

Peak Power Control: The power consumption behavior and models of different switch fabrics have been explored in [24]. Techniques for optimizing power dissipation in high speed links have been proposed in [25]. Analytical power models for interconnection networks have been developed based on transistor counts in [26]. [27] has presented an analytical power model to explore different switch configurations. While extended Dynamic Voltage Scaling (DVS) technique [28] to optimize link power in regular interconnection networks can conserve significant link energy, it degrades network latency severely especially at low to medium load [14]. Recently, Dynamic Link Shutdown (DLS) technique was proposed [14], which shuts links with low utilization down intelligently.

[29] developed the Muse prototype, which enables a data center to determine the number of active servers in a cluster in the view of the overall performance and energy cost. PowerHerd [30] has a distributed mechanism in each router to dynamically regulate power to ensure that the peak power constraint in the interconnection network is not exceeded.

3 System Architecture

In this section we describe the cluster interconnect. It includes a QoS-capable wormhole router architecture, the NIC or the HCA, and a rate-based scheduling algorithm, called VL arbitration, used in the router and the HCA. Also the energy model used in the cluster interconnect is explained.

3.1 Router Architecture

The pipelined wormhole router is shown in Figure 1. The first stage of the pipeline 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 bypass Stages 2 and 3, and 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 HCA 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. The VL arbitration is described in Section 3.3.

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 [11], 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 in this study 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. Note that the CA specification is very similar to the VIA design.

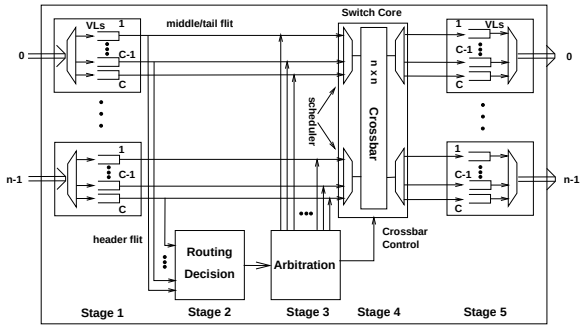


Figure 1. A 5-stage Pipelined Router Model

The HCA architecture proposed in IBA is shown in Figure 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 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 [3]. 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,

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

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 [11]. 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.

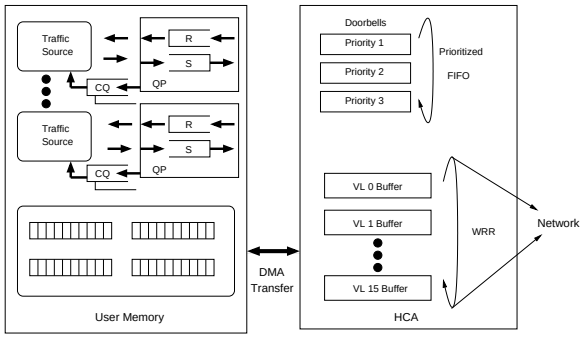


Figure 2. A Proposed IBA HCA with QoS Support

To make the CA design compatible to the QoS-aware router of the previous section, 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 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 Figure 1). The arbiter selects the flit to transmit from the set of candidate flits competing for that port.

IBA specifies a two-level scheme for VL arbitration. First, all the applications are classified into different priority classes (SLs) 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.

Although the WRR scheme is used for both best-effort and real-time traffic, it degenerates to the simple RR scheme for best-effort traffic. Here we discuss how we assign weights to real-time traffic. The weight value (0 - 255) specifies the amount of bandwidth allocation. Since IBA has limited number of VLs for data transfer, many connections may share the same VL. Thus, the weight of a VL implies the total amount of bandwidth allocated to all connections using the VL. The actual weight assignment to different connections is the intricate part of the WRR scheme. Given the total number of frames (F) and the bandwidth R_v allocated to a VL v, the WRR scheme computes the weights as follows.

Let r_i be the bandwidth requirement of connection i that is assigned to the VL v, and W_v be the weight of VL v. Then $R_v = \sum_{\forall \text{ connection } i \text{ in } VL_v} r_i$. The proportional bandwidth allocated to VL v (ρ_v) is given by $\frac{R_v}{\sum_{\forall j} VL_j \, R_j}$. The weight for VL v is given by $W_v = (\rho_v \times F)$. (The maximum frame size in IBA is 255×64, because there are 64 entries in the VL arbitration table and the maximum weight value of each entry is 255.) For example, let there be only 2 VLs, and $R_1 = 100$ and $R_2 = 200$. In this case, the weight assignment becomes

$W_1 = 1$, and $W_2 = 2$ for $F = 3$. If we use a larger frame size ($F = 300$), then $W_1 = 100$, and $W_2 = 200$.

If the proportional bandwidth assignments are all rational as in the previous example, ($\rho_1 = \frac{1}{3}, \rho_2 = \frac{2}{3}$), where a bandwidth allotment can be expressed by two integers ($\rho = \frac{a}{b}$), then the frame size F will be the least common multiple of the denominators. But with $R_1 = 101$ and $R_2 = 200$ in the previous example, this scheme gives $W_1 = 101, W_2 = 200$, and $F = 301$. These weight values can be used for allocating *exact* bandwidth without truncation error. But, with a fixed small frame size ($F = 6$), $R_1 = 100$ and $R_2 = 200$, we get $W_1 = 2$ and $W_2 = 4$, and for $R_1 = 101$, $R_2 = 200$, we also get the same weights, $W_1 = 2$ and $W_2 = 4$, indicating truncation error.

Next, let us examine the implementation of the WRR scheme. Each entry in the VL arbitration table contains a VL number and its weight value. A pointer circulates the table in a round robin fashion and points to the next entry eligible for scheduling. There are two ways to spread bandwidth with this approach. First, once one entry has a turn to send, it can contiguously transfer flits equal to its assigned weight. (This is identical to the IBA specification, although the unit of transfer in IBA is one packet.) Let us call this scheme a slow moving pointer. Second, only one flit is transfered at any time and then the pointer moves to the next available entry. This is termed as a fast moving pointer scheme.

Therefore, we have four options to implement a WRR scheme: (i) A Small Frame size with a fast moving pointer; (ii) A Small Frame size with a slow moving pointer; (iii) A Large Frame size with a fast moving pointer; and (iv) A Large Frame size with a slow moving pointer. A small frame size has the traffic reshaping ability to handle traffic burst [31], but suffers from truncation error. A large frame size, on the other hand, avoid truncation errors, but cannot mitigate traffic burst. We have analyzed all these four designs in this study.

We have implemented the WRR schemes with a small fixed frame number and the largest frame size allowed in the IBA specification (255×64). The small frame size is decided intuitively as ($k \times N$), where N is the number of VLs for real-time applications and k is a small constant. Let B be the total link bandwidth and B_r be the average allocated bandwidth for all real-time traffic. For a VL v of real-time traffic, $\rho_v = \frac{R_v}{B_r}$. Then, $W_v = (\rho_v \times F)$. In the case of a large frame, the admission control algorithm will not refresh the assignments of all weights, but will only update the weight for the connection, if B_r is fixed. If the allocated bandwidth for a certain SL dynamically changes during connection set up, we will not be able to get the above advantage. In case of a small frame size, the admission control algorithm re-computes the weights for each new connection setup.

Finally we discuss the assignment of Limit for transferring high priority traffic. In wormhole switching, bandwidth requirement is carried by the header flit of a message. For this study, since we use admission control, the bandwidth requirement for connection i is carried by a probe message as described in the next section. The bandwidth requirement in the probe is for the entire connection. While the bandwidth requirement (r_i) represents the average bandwidth of a stream i, the peak rate of the stream (p_i) is used for the assignment of *Limit of high priority*, denoted as L. Let B_p be the sum of peak bandwidth of all real-time traffic and is given by $B_p = \sum p_i$. If $B_p \geq B$, we assign $L = 255$. (The admission control algorithm will accept the connection only if the sum of average rate is less than the total bandwidth.) If $B_p < B$, then $L = \frac{B}{B - B_p}$, since $B = B_p + B_b$ and $B_b = \frac{B}{L}$ where B_b is the minimum allocated bandwidth for best-effort traffic. With this limit, we can prevent starvation of lower priority messages by assuring a minimum forwarding bandwidth (B_b).

3.4 Energy Model

The router energy model has 4 components: FIFO buffers, lookup tables, crossbar and output port arbiter. The main energy parameters used in this paper are summarized in Table 1. In this model each buffer is broken into a number of cells, where each cell has one sleep tran-

Component	Status	Energy (pJ)
Input/Output Buffer (per flit)	Read	31.125
	Write	27.075
Arbitration	Active	6.10086
Crossbar (per flit)	Active	68.475
Header Size	128 bits	N/A
Input/Output VL Buffer Size	1280 bits	N/A
Physical Link Energy Consumption (per bit)	2.5 Gbps	10.21

Table 1. System and Energy Parameters (180 nm Design)

sistor. We power down cells after reading them, since the FIFO access pattern is deterministic and the data is not needed again.

Our HCA energy model has a RISC processor, 8MB local memory, DMA controller, doorbell queues, and VL arbiter. We use DRAM data sheets [32] to obtain the energy numbers for the local memory and the doorbell queues. To evaluate the energy consumed by the RISC processor in the HCA, we use a StrongARM 1100 RISC core [33] based energy simulator and execute the kernel code. The physical links are capable of sending 2.5 Gbps data over a reasonable distance for cluster interconnects. The link includes the transmitter, receiver, and clock recovery at the receiver. The link energy consumption value for the 180 nm design is also shown in Table 1.

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 router. If rejected, a NACK is sent back to the traffic source without changing the currently used bandwidth of the physical link.

Upon receiving the probe packet, each router compares the available link bandwidth of the destination port for the packet with the requested bandwidth 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 solicited 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.

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.

After receiving the ACK message, the source starts to send its data packets. On the other hand, if it receives 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 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 [34, 11].

4.2 Congestion Control Algorithm

Network congestion occurs 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 preventive mechanism. Note that unlike the Internet congestion control, we do not allow dropping packets in the network. (Also, packet dropping is complex in wormhole-switched network.) If the network resources are not available, packets can be dropped or back-logged 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[2]. 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.[3] 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

[2]IBA also specifies end-to-end flow control only for reliable Connected service. The receiver informs the sender about availability of credits using dedicated flow control packets [3].

[3]For reliable service, CQE will be placed after getting an ACK from the receiver. In this case, we can have a different queue only for credits from HCA.

will be equal to the incoming rate of credits to the CQ. 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 ($\sum_i C_i \leq M$). For each established connection, initial credits are given according to the bandwidth requirement (b_i) as follows:

$$C_i = \frac{b_i}{B} \cdot M \quad \text{where } B \text{ is the channel bandwidth.}$$

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 $C_b \leq M - \sum_i C_i$.

Credits are generated by the HCA and consumed by a consumer. If the consumer does not have a credit in its CQ, 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 (message dropping) is used.

4.3 Separate CQs for End-to-End Congestion Control

IBA specification [3] makes it possible to have a multiple CQ configuration in the HCA. For connectionless best-effort traffic, we do not separate CQs for each application, while for connection based real-time traffic, CQs will be created per connection. The initial credits for real-time traffic in the HCA can be given per flow, as a ratio of bandwidth requirement to channel bandwidth. But there is no proper criterion on the number of credits for best-effort traffic. Instead of assigning credits to the flow like real-time traffic, a separate CQ for each destination can be constructed. When a whole packet enters the network from an HCA, a credit will be generated and placed into a CQ according to its destination. We call this scheme *Separate CQ*. Since the stored credits of each CQ reflects the amount of available resources of a different source-destination pair, Separate CQ enables Credit-based Congestion Control to provide more fine-grained traffic control than the Global CQ (a single CQ configuration).

The number of CQs in the Separate CQ configuration is the same as the number of nodes in the network. As the size of network increases, multiple CQs can be a waste of resources. In addition, its configuration is hard to adapt when a new node has to be inserted or deleted dynamically. Minimal routing on a mesh or torus topology uses only one out of four quadrants; partitions based on a source coordinate in both x-axis and y-axis. Based on such a property, we can build Quad CQ configuration that has 4 CQs in an HCA.

5 Peak Power Control

In this section, we propose Credit-based peak power control that regulates the injection of packets in the HCA. Fundamentally, before sending a packet or establishing a connection, we check the power budget availability (also called power credits) for the packet or the connection. Only packets/connections which earn power credits can enter the network. This scheme is implemented based on the Credit-based Congestion Control to prevent performance degradation under high load.

5.1 Peak Power Control

The peak power constraint is given to the network to avoid thermal emergency, when the system designer divides the total power budget for each part of the system. To induce peak power constraint, It is required for the reliable communication between routers to operate safely as well as for the tight power budget to overcome over-provisioning cooling packages. Since the routers try to consume more power to maximize their performance, the router circuit can get burned or have malfunctions without the peak power control [15]. Therefore, in order to guarantee power consumption of the network under the specified power limit, we regulate the injection rate

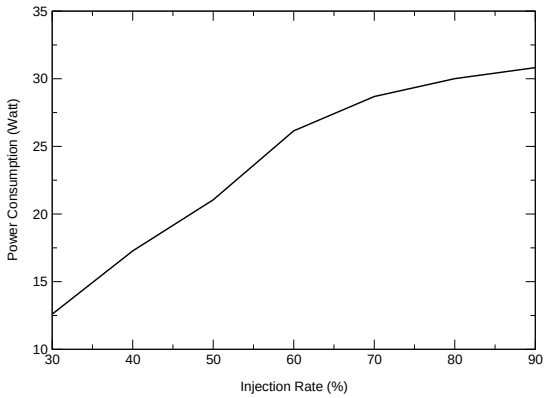


Figure 3. Power Consumption of a 4×4 Mesh Network

in the HCA. As shown in Figure 3, we can observe that the power consumption is proportional to the network load, which is in turn determined by the sum of the input load at each HCA. Note that, if we successfully control the peak input load that corresponds to the peak power, the network can be sustained without any malfunctions.

Our goal is to maintain the service quality while controlling the peak power consumption. According to the IBA specification [3], we can categorize network traffic into two classes; connection-based (real-time traffic) and connectionless (best effort traffic). Connection-based traffic usually requires QoS guarantees, which means that once a connection is established, we need to provide a certain SL. On the other hand, connectionless traffic does not have such a strict demand on performance. Thus, we should take different approaches to handle each type of traffic. To apply different schemes for each traffic class, the total power budget (P_{total}) is divided and distributed to each traffic class by its ratio.

$$P_{\text{total}} = P_{\text{RT}} + P_{\text{BE}} < P_{\text{peak}},$$

where P_{RT} is power consumption for real-time traffic and P_{BE} is power consumption for best-effort traffic. The power budget for the peak power constraint is the average power consumption over the thermal time constant, which is several seconds in chip-to-chip networks [35].

Real-time traffic throttling: Since each flow of real-time traffic has its bandwidth requirement and the requirement should be guaranteed, we cannot throttle the injection of packets for the admitted connection to control the power consumption. Thus we can only restrict the number of connections/flows to be admitted in the network so as to meet the peak power consumption. This scheme can be regarded as an extension of admission control. In the proposed admission control in Section 4, we only check the bandwidth availability. In our peak power control, we also check the power budget along with the bandwidth availability.

Before a new connection is established in the network, the probe packet checks whether the routers on the path from the source to the destination have both sufficient bandwidth and sufficient power budget for this new connection to guarantee QoS and to meet the power constraints, respectively. If all routers have both sufficient bandwidth and sufficient power budget, this connection can be established in the network. Otherwise, it gets rejected.

The bandwidth of the connection determines the power required for that connection. The real-time traffic, whose bandwidth is r bps, injects r/f flits per second, where f is the flit size in bits. The energy consumed for a single flit in a router (E_{router}), is obtained from our power model of the router. So a flit consumes $E_{\text{router}} \times r/f$ Watt at each router. The power budget of each router on the path of the connection is recalculated by subtracting the consumed power from the former power budget of the router. This can be denoted by the following formula. For each router i in the set of the routers on the path,

$$P_{\text{RT}}(i) = P_{\text{RT}}(i) - E_{\text{router}} \times r/f$$

where $P_{\text{RT}}(i)$ is the power budget for router i. The bandwidth of router i with the updated power budget, $P_{\text{RT}}(i)$, is equivalent to the available bandwidth that can be assigned to a new real-time connection in the router.

When we set the peak input load to k, the allo-

cated power budget of the router i for real-time traffic is $E_{\text{router}} \times kR/f$, where R is the maximum bandwidth. The sum of all routers' power budgets is equal to the total power budget of real-time traffic (P_{RT}).

Best-effort traffic throttling: Since best-effort traffic has no bandwidth requirement, it seems impossible to satisfy the peak power constraint by controlling the admission of connections. Instead of using the bandwidth requirement, our method estimates the consumed energy for a packet and traversal time in the network. An HCA updates its energy budget at both the packet departure time and the expected arrival time. If the HCA does not have enough energy budget for a new packet to deliver, the packet will be throttled. To achieve this, we need to convert the power budget (P_{BE}) into the energy budget (E_{BE}).

The power budget can be converted into the energy budget for every short period of time, T. (Note that $E_{\text{BE}} = P_{\text{BE}} \cdot T$ [30].) The total amount of best-effort power budget (P_{BE}) is equally divided and assigned to the power budget of each HCA. We need to estimate how much energy and time are required for a single packet delivery on a certain path. The wormhole switching requires energy (E_{packet}) and time (L_{packet}) to send a packet from a given source to a destination as shown in the following formulas assuming one packet consists of N flits.

$$E_{\text{packet}} = (E_{\text{router}} \cdot (D+1) + E_{\text{link}} \cdot D) \cdot N.$$

$$L_{\text{packet}} = ((D+1) \cdot S - 1) + N + W_{\text{HCA}} \cdot$$

E_{router} and E_{link} are energy values consumed by routers and links respectively. D is the distance between the source and the destination (number of hops). S is the sum of cycles for router and link operations. W_{HCA} denotes a queueing delay in the HCA and it can be estimated from the queue operations. The packet arrival time (L_{packet}) is estimated in terms of clock cycles.

When a packet departs from the HCA, the estimate of energy (E_{packet}) is subtracted from the energy budget of the HCA if it has a sufficient energy budget. After L_{packet} cycles, the value is restored to the former energy budget. Since each HCA does not accept a best-effort packet over the given energy budget, the sum of all energy budgets in the HCAs is less than the peak energy budget the system designer has set. Therefore, total allocated energy budget for best-effort traffic is satisfied by monitoring the energy budget and regulating the injection rate inside the HCA.

6 Experimental Platform

6.1 Simulation Testbed

For evaluating the proposed designs, we have developed flit-level simulation models for the QoS-aware routers and HCAs. The simulation models are flexible in that one can specify the number of physical channels (PCs), number of VLs per PC, link bandwidth, flit size, packet size, mean and variation of Variable Bit Rate (VBR) traffic, and many other architectural and workload parameters. It is also possible to configure any network topology using these routers.

For our experiment, we simulated an 8-port router connected with HCAs and a 2-D mesh network designed using 5-port/6-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 packet consists of 40 flits except for the control packets, which are 10-flit long. Physical link bandwidth is 1.6Gbps (2.5 Gbps for Peak power control), 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 and Peak power control in wormhole-switched networks, we are using parameters compatible with recent router design.

6.2 Workload

Our workload includes packets from real-time VBR traffic or ON/OFF traffic, best-effort traffic, and control traffic. The VBR traffic is generated as a stream of packets between a pair of communicating (source-destination) processors. The traffic in each stream is

generated from real MPEG-2 traces [36], where there are 7 video traces with different bandwidth requirements. Each stream generates 30 frames/sec, and each frame is fragmented into 40-flit size packets (except possibly the last packet of a frame).

Once the input VL for a connection is determined, the destination processor is picked randomly using a uniform distribution of all nodes, and the destination VL is also drawn randomly from a uniform distribution of the VLs available for the VBR traffic.

Since the simulation with MPEG 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 packets between a pair of source and destination nodes. During the OFF period, the source does not generate any packets, while during the ON period, packets are generated according to the given injection rate λ_{onoff}. To avoid traffic burst, the generation is evenly scattered. The ON/OFF model with exponentially distributed ON and OFF times is commonly used to simulate real-time traffic [37, 38].

The best-effort traffic is generated with a given injection rate λ_{be}, and follows the Poisson distribution. Best-effort packets are assumed 40-flit long, and a destination is picked using a uniform distribution. The input and output VLs for a packet 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 rate of control traffic is very low (ten packets per 33.3 ms of simulation with MPEG-2 traffic and ten packets per (OFF period + ON period)), and only one VL (VL15) is assigned for this traffic.

The important output parameters measured in our experiment are Deadline Missing Probability (DMP) of delivered MPEG-2 frames, average Deadline Missing Time (DMT) of deadline missing frames, and average network latency for ON/OFF traffic, best-effort traffic, and control traffic. 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 ms 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 ms to the arrival time of the previous frame. Whenever a frame misses its deadline, we measure the deadline missing time and then calculate the average DMT. We use power consumption (watts) and packet latency to compare the simulation results on peak power control.

7 Performance Results

We discuss the performance results in four subsections. First, we compare the four implementations of WRR scheme. Secondly, we analyze the proposed credit-based congestion control scheme. Next, we evaluate the clusters with both admission and congestion control. Finally we investigate the effectiveness of the proposed peak power control. Most of the results for admission and congestion control are presented for a real-time to best-effort ratio of 80:20, while the workload for peak power control consists of 50% real-time and 50% best-effort traffic.

7.1 Comparison of Four WRR Implementations

We begin by comparing the performance results of the four WRR schemes described in Section 3.3. These are (i) A Small Frame size with a fast moving pointer; (ii) A Small Frame size with a slow moving pointer; (iii) A Large Frame size with a fast moving pointer; and (iv) A Large Frame size with a slow moving pointer. For 80% real-time traffic, 11 VLs are assigned. With the constant $k = 4$, the small frame size becomes 44. We gather the results from a single router cluster, which has both congestion control and admission control. The traffic mix includes control, best-effort and MPEG-2 traces. We use an input regulator to remove the traffic burst in MPEG streams. It is already known that with traffic burst, the first scheme will perform better [31]. With this experiment what we try to answer is the following question: *Which WRR implementation will be*

WRRs	Load	Fast Pointer		Slow Pointer	
		DMP	DMT(ms)	DMP	DMT(ms)
Small Frame Size ($F = 44$)	60%	0.030	0.031	0.034	0.034
	70%	0.027	0.038	0.029	0.036
	80%	0.021	0.040	0.022	0.089
Large Frame Size ($F = 255 \times 64$)	60%	0.034	0.031	0.035	0.035
	70%	0.024	0.037	0.032	0.034
	80%	0.023	0.089	0.024	0.104

Table 2. Performance Results of Four WRR Implementations (Single Router Cluster)

a good choice in terms of performance and complexity even without traffic burst? In Table 2, the first scheme among the four WRR implementations provides the best results in terms of DMP and DMT (DMT is around 0.04 msec compared to 0.09∼0.10 msec in other three cases for 80% input load), although we have used all three mechanisms(regulator, admission and congestion control) that prevent traffic burst. It turns out that the truncation error of a small frame size doesn't affect performance. We believe these differences will be more pronounced in a larger network and without traffic controlling mechanisms. However, the main advantage of WRR with a large frame is in reducing the complexity of weight computation. This is a better choice where the real-time applications include many short living connections and hence, admission control is invoked frequently. Since we use long lasting MPEG-2 streams, which do not need frequent weight calculation, we use the first WRR scheme (small frame size ($F = 44$) and a fast pointer) in the rest of the experiments.

7.2 Comparisons of Congestion Control Algorithms

We simulated the prior At-Least-One(ALO) [16] and the Self-Tuned [17] 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. The mechanism for gathering the global information is described in [17]. We use the same parameters given in [17] to simulate the scheme. Since these two schemes were developed for the network only (no NIC), 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. (If we use in-band (implying VL 15 for control traffic) for communication, then the Self-Tuned scheme results may be worse than those presented here.)

Figure 4 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 depict that we get the best results with 200 initial credits. As expected, higher number of initial credits injects more traffic into the network and thus increases delay. Without any congestion control, network throughput experiences a sudden drop due to saturation [16, 17]. But in our study, since there is an

HCA buffer, the source cannot inject any more messages when the HCA buffer is full. Therefore, the HCA buffer acts on an implicit congestion control and hence, the throughput drop is avoided.

Table 3 shows average packet queueing delay for each scheme in the same 4×4 mesh network. Without congestion control, packets experience low queueing delay at low loads, but as the load increases, the delay is rapidly growing. With the credit-based congestion control, the delay is kept low and slowly increases even at higher loads.

7.3 Results with Admission and Congestion Control

Figure 5 (a) plots the Deadline Missing Probability (DMP) and 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 Figure 5 (b). In the figures, **A,C** indicates a router with both admission and congestion control. **A** means a router with only admission control, and **C** means a router with only congestion control, while **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 ms.) The DMP and 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 admission and congestion control. Figure 5 (b) indicates that the average network 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.

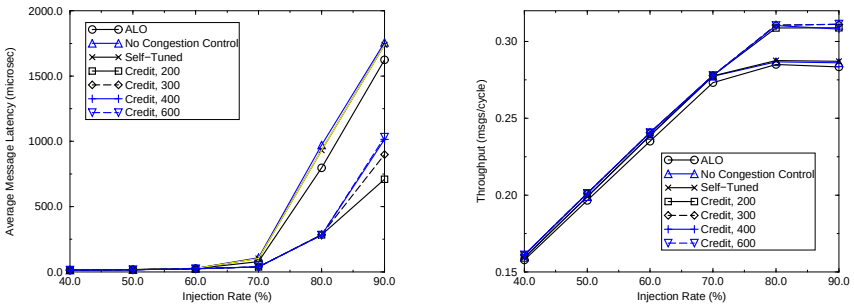
Figure 6 shows the effect of admission control and congestion control in a 5 × 5 mesh network with ON/OFF real-time traffic. The general trend in all these graphs is that the performance is the best with both admission and congestion control, followed by only admission control, and then congestion control 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 and is not controlled by admission control. On the contrary, as depicted in 6 (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.

Table 4 shows real-time traffic connection rejection rates for four combinations of the schemes. Without admission control (No A, C and No A, No C), no connection is rejected and, therefore the rate is 0. With the proposed admission control, the rejection rates are still kept low (10.4% $\sim$ 25.3%), which implies most of the requested connections are accepted.

7.4 Separate CQs and Peak Power Control Results

Figure 7 shows average latency and power consumption of different CQ configurations. We use only congestion control for this experiment, because multiple CQ configurations are developed for fine-grained congestion control. Average packet latency in the multiple CQ configuration is reduced by 2.5 times over the global CQ configuration. Quad CQ is slightly better than separate CQ, which implies that each CQ in Quad CQ configuration models the congestion status of one quadrant properly. We cannot observe any prominent difference between separate and quad CQ configurations for power consumption.

We set the peak power constraint to the average



(a) Message Latency (b) Throughput

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

Injection Rate	40	50	60	70	80	90
No Congestion Control	1.115	1.489	2.273	5.260	231	1050
Credit 200	1.116	1.539	2.261	5.098	207	619
Credit 300	1.116	1.539	2.257	5.098	206	805
Credit 400	1.118	1.537	2.257	5.098	204	919
Credit 600	1.118	1.537	2.257	5.098	204	937
Self-Tuned	452	543	629	755	844	
ALO	251	455	1566	2422	3075	3455

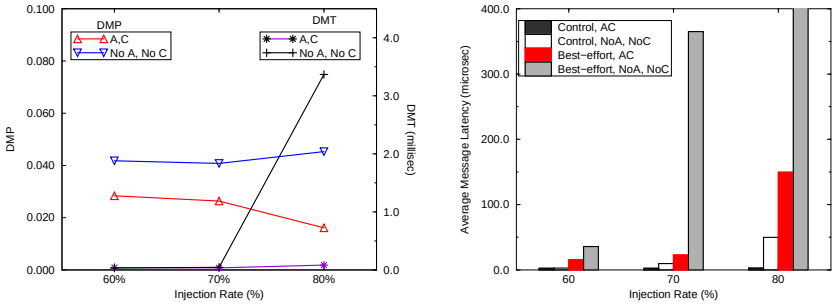
Table 3. Packet Queueing Delay in a 4×4 Mesh Network with 100% Best-Effort Traffic (in μsec)

power value (21 Watts) consumed at the injection rate of 50% as shown in Figure 3. We compare the Credit-based peak power control with the Credit-based congestion control and PowerHerd [30] in terms of performance and power consumption. We use the same peak power value for the comparison between our schemes and PowerHerd.

The total average latency of mixed traffic is plotted in Figure 8 (a), while the average latencies of best-effort and real-time traffic are shown in Figure 8 (b) and (c), respectively. In these graphs, 4 different schemes are evaluated: a cluster without any power control (**None**), a cluster with the credit-based congestion control (**CC**), a cluster with the peak power control (**PC**), and a cluster with PowerHerd (**PH**). Note that **None** and **CC** do not have any peak power control. Although **PH** is very effective in controlling the peak power, it incurs severe performance degradation as shown in Figure 8. Our scheme (**PC**) shows the best performance for both best-effort and real-time traffic.

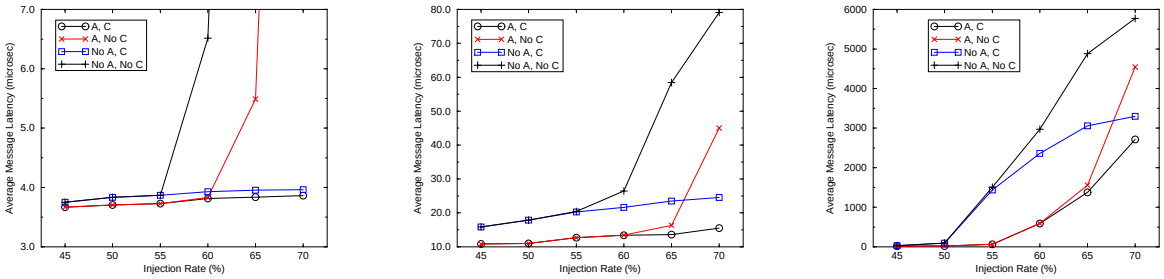
Figure 9 (a) shows the average power consumption for mixed traffic that consists of 50% real-time and 50% best-effort traffic. This graph indicates that congestion control alone (**CC**) does not help to reduce the power consumption of the network, since it controls the injection rate only to prevent network saturation. It tries to distribute the load evenly to all the routers to maximize the utilization, thus causing the power consumption to exceed the peak power constraint (21 Watts).

We also conduct experiments with 100% best-effort and 100% real-time traffic to show the net effect of our peak power control, since we have different power control schemes for each traffic class. From Figure 9 (b), we can tell that our scheme is more effective in refraining the power consumption than PowerHerd, while still providing better performance as shown in Figure 8 (b). Note that, for real-time traffic, our scheme decides whether new real-time connections are accepted or not. Once accepted, a real-time connection may consume more bandwidth than what it originally requested.



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

Figure 5. Performance Results of a Single Router Cluster with MPEG-2 Video Traffic



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

Figure 6. Average Message Latency in a 5×5 Mesh Network with On/Off Real-Time Traffic

Therefore, as shown in Figure 8 (c), the peak power control (**PC**) consumes more power than PowerHerd (**PH**), while still refraining the power consumption under the predetermined value.

To show runtime peak power satisfaction under high injection rate, we measure the average power consumption every 500 cycles (1.88 μs). Figure 10 shows that the network with the proposed peak power control refrains the power consumption to satisfy the given peak power constraint, which is 21 Watts.

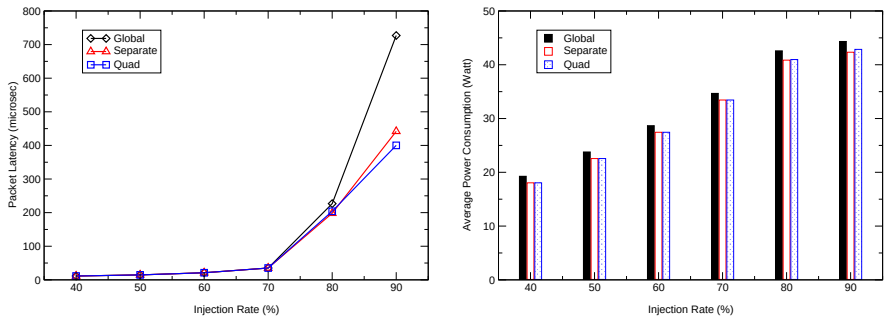
8 Concluding Remarks

This paper presents admission, congestion and peak power control mechanisms for wormhole-routed cluster interconnects to provide QoS guarantees in clusters and System Area Networks (SANs). While QoS in clusters/SANs 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, although the algorithms are developed for QoS-capable wormhole routers and QoS-capable NIC/HCAs, they are equally applicable to other networked architectures.

The important conclusions of this work are the following: First, the smaller frame implementation of the WRR scheme seems a reasonable choice for the workloads studied in this paper. Second, 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. The impact of our unified admission and congestion control becomes

Injection Rate	40	45	50	55	60	65	70
A, C	16.4%	21.0 %	25.3%	12.7%	15.7%	17.6%	10.4%
A, No C	16.4%	21.0 %	25.3%	12.7%	15.7%	17.6%	10.4%
No A, C	0	0	0	0	0	0	0
No A, No C	0	0	0	0	0	0	0

Table 4. Connection Rejection Rate in 5×5 Mesh Network with On/Off Real-Time Traffic



(a) Average Latency (b) Average Power

Figure 7. Different CQ Configurations with Best-Effort Traffic Only

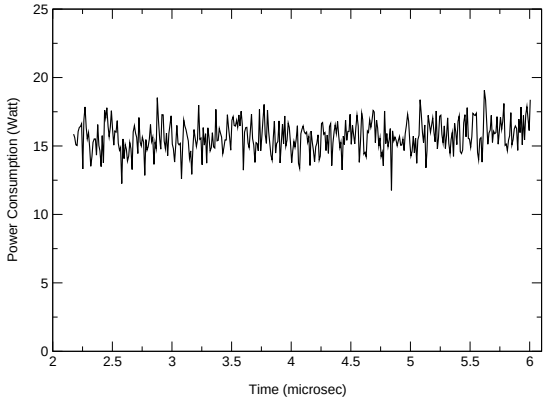


Figure 10. Runtime Power Consumption in a 4×4 Mesh Network under the Input Load 70% of Mixed Traffic

more pronounced with higher workloads and with non-uniform traffic. Third, 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. An integrated admission and congestion control mechanism can provide significant performance improvement resulting in better QoS guarantees. Four, further enhancement in the HCA, multiple CQ/quad CQ configurations, can improve the performance by providing fine-grained control for best-effort traffic. Finally, the peak power control mechanism in this paper is effective in controlling the power consumption while providing competitive performance. Our peak power control works like admission control for real-time traffic and like congestion control for best-effort traffic.

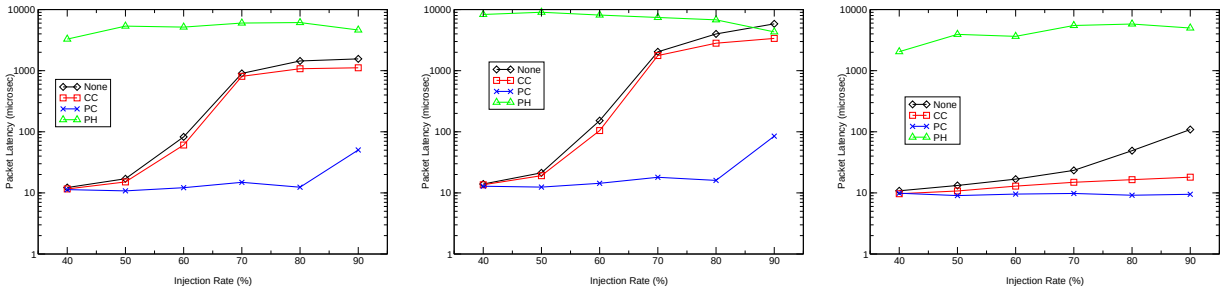
References

[1] Mellanox Technologies Inc., "Mellanox Performance, Price, Power, Volumn Metric (PPPV)." Available from `http://www.mellanox.com/products/shared/PPPV.pdf`.

[2] Digital Equipment Corporation, Maynard, MA, *Alpha Architecture Technical Summary*, 1992.

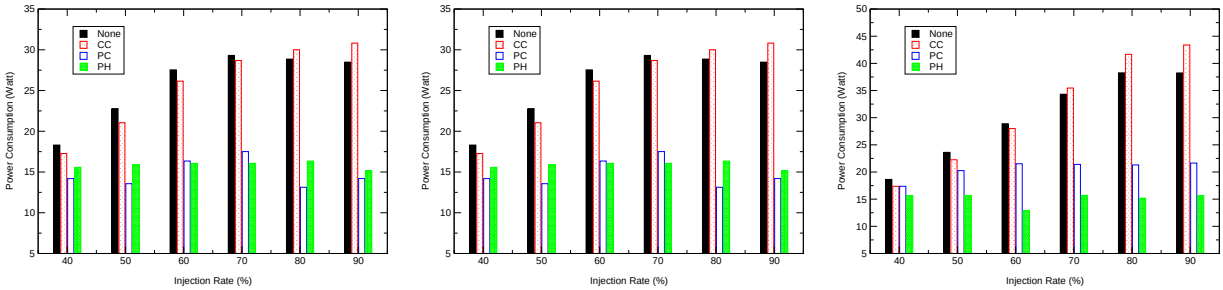
[3] InfiniBand Trade Association, "InfiniBand Architecture Specification, Volume 1, Release 1.0," October 2000. Available from `http://www.infinibandta.org`.

[4] J. H. Kim and A. A. Chien, "Rotating Combined Queueing (RCQ): Bandwidth and Latency Gurantees in Low-Cost, High-Performance Networks,"



(a) Mixed Traffic (b) Best-Effort Traffic (c) Real-Time Traffic

Figure 8. Average Latency in a 4×4 Mesh Network



(a) Mixed Traffic (b) Best-Effort Traffic Only (c) Real-Time Traffic Only

Figure 9. Average Power Consumption in a 4×4 Mesh Network

in *Proceedings of the International Symposium on Computer Architecture*, pp. 226–236, May 1996.

[5] H. Eberle and E. Oertli, "Switcherland: A QoS Communication Architecture for Workstation Clusters," in *Proceedings of the International Symposium on Computer Architecture*, pp. 98–108, June 1998.

[6] J. Rexford, J. Hall, and K. G. Shin, "A Router Architecture for Real-Time Point-to-Point Networks," in *Proceedings of the International Symposium on Computer Architecture*, pp. 237–246, May 1996.

[7] J.-P. Li and M. Mutka, "Priority Based Real-Time Communication for Large Scale Wormhole Networks," in *Proceedings of International Parallel Processing Symposium*, pp. 433–438, May 1994.

[8] J. Duato, S. Yalamanchili, M. B. Caminero, D. Love, and F. J. Quiles, "MMR: A High-Performance Multimedia Router-Architecture and Design-Tradeoffs," in *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture*, pp. 300–309, January 1999.

[9] J. Pelissier, "Providing Quality of Service over InfiniBand Architecture Fabric," in *Proceedings of Symposium on High Performance Interconnects (Hot Interconnects 8)*, August 2000.

[10] K. H. Yum, E. J. Kim, C. R. Das, and A. S. Vaidya, "MediaWorm: A QoS Capable Router Architecture for Clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, pp. 1261–1274, December 2002.

[11] K. H. Yum, E. J. Kim, and C. R. Das, "QoS Provisioning in Clusters: An Investigation of Router and NIC Design," in *Proceedings of the International Symposium on Computer Architecture*, pp. 120–129, June 2001.

[12] The New York Times, "There's money in housing internet servers," April 2001. Available from `http://www.internetweek.com/story/INW20010427S0010`.

[13] C. D. Patel, R. Sharma, C. E. Bash, and A. Beitelmal, "Thermal Considerations in Cooling Large Scale High Compute Density Data Centers," in *Proceedings of the Eighth Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pp. 767–776, May 2002.

[14] E. J. Kim, K. H. Yum, G. M. Link, N. Vijaykrishnan, M. Kandemir, M. J. Irwin, M. Yousif, and C. R. Das, "Energy Optimization Techniques in Cluster Interconnects," in *Proceedings of International Symposium on Low Power Electronics and Design (ISLPED'03)*, pp. 459–464, August 2003.

[15] L.-T. Yeh and R. C. Chu, *Thermal Management of Microelectronic Equipment*. ASME Press, 2002.

[16] E. Baydal, P. Lopez, and J. Duato, "A Simple and Efficient Mechanism to Prevent Saturation in

Wormhole Networks," in *Proceedings of 14th International Parallel and Distributed Processing Symposium*, pp. 617–622, 2000.

[17] M. Thottethodi, A. R. Lebeck, and S. S. Mukherjee, "Self-Tuned Congestion Control for Multiprocessor Networks," in *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture*, pp. 107–118, January 2001.

[18] D. Ferrari and D. C. Verma, "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 368–379, 1990.

[19] H. Saito and K. Shiomoto, "Dynamic Call Admission Control in ATM Networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 982–989, September 1991.

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

[21] P. T. Gaughan and S. Yalamanchili, "Pipelined Circut-Switching: A Fault-Tolerant Variant of Wormhole Routing," in *Proceedings of Fourth IEEE International Symposium on Parallel and Distributed Processing*, December 1992.

[22] L.-S. Peh and W. J. Dally, "Flit-Reservation Flow Control," in *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture*, pp. 73–84, January 2000.

[23] A. Smai and L. Thorelli, "Global Reactive Congestion Control in Multicomputer Networks," in *Proceedings of 5th International Conference on High Performance Computing*, pp. 179–186, 1998.

[24] T. T. Ye, L. Benini, and G. D. Micheli, "Analysis of Power Consumption on Switch Fabrics in Network Routers," in *Proceedings of the 39th Conference on Design Automation*, pp. 524–529, June 2002.

[25] G.-Y. Wei, S. Sidiropoulos, D. Liu, J. Kim, and M. Horowitz, "A Variable-Frequency Parallel I/O Interface with Adaptive Power-Supply Regulation," *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 1600–1610, November 2000.

[26] C. S. Patel, S. M. Chai, S. Yalamanchili, and D. E. Schimmel, "Power Constrained Design of Multiprocessor Interconnection Networks," in *Proceedings of International Conference on Computer Design*, pp. 408–416, October 1997.

[27] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik, "Orion: A Power-Performance Simulator for Interconnection Networks," in *Proceedings of the 35th International Symposium on Microarchitecture (MICRO)*, November 2002.

[28] L. Shang, L.-S. Peh, and N. K. Jha, "Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks," in *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture*, pp. 91–102, February 2003.

[29] J. S. Chase, D. C. Anderson, P. N. Thakar, and A. M. Vahdat, "Managing Energy and Server Resources in Hosting Centers," in *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, pp. 103–116, 2001.

[30] L. Shang, L.-S. Peh, and N. K. Jha, "PowerHerd: Dynamic Satisfaction of Peak Power Constraints in Interconnection Networks," in *Proceedings of International Conference on Supercomputing*, pp. 98–108, June 2003.

[31] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted Round-Robin Cell Multiplexing in a General-Purpose ATM Switch Chip," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 1265–1279, October 1991.

[32] T. G. Tip, "RDRAM Power Estimation and Thermal Considerations," October 2001. http://www.rambus.com/rdf/presentations/2_A3_Thermal_Yip2.pdf.

[33] T. Simunic, L. Benini, and G. D. Micheli, "Cycle-Accurate Simulation of Energy Consumption in Embedded Systems," in *Proc. DAC*, 1999.

[34] K. H. Yum, A. S. Vaidya, C. R. Das, and A. Sivasubramaniam, "Investigating QoS Support for Traffic Mixes with the MediaWorm Router," in *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture*, pp. 97–106, January 2000.

[35] K. Skadron, T. F. Abdelzaher, and M. R. Stan, "Control-Theoretic Techniques and Thermal-RC Modeling for Accurate and Localized Dynamic Thermal Management," in *Proceedings of International Symposium on High-Performance Computer Architecture*, pp. 17–28, February 2002.

[36] M. B. Caminero, F. J. Quiles, J. Duato, D. S. Love, and S. Yalamanchili, "Performance Evaluation of the Multimedia Router with MPEG-2 Video Traffic," in *Proceedings of the Third International Workshop on Communication, Architecture and Applications on Network Based Parallel Computing (CANPC'99)*, pp. 62–76, January 1999.

[37] S. Jamin, P. B. Danzig, S. J. Shenker, and L. Zhang, "A Measurement-Based Admission Control Algorithm for Integrated Services Packet Networks," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 56–70, February 1997.

[38] J. Qiu and E. Knightly, "QoS Control via Robust Envelop-Based MBAC," in *Proceedings of IEEE/IFIP IWQoS*, May 1998.