Network Endpoint Congestion Control for Fine-Grained Communication

Nan Jiang
NVIDIA Research
Santa Clara, CA
tedj@nvidia.com

Larry Dennison
NVIDIA Research
Santa Clara, CA
ldennison@nvidia.com

William J. Dally
NVIDIA Research
Santa Clara, CA
bdally@nvidia.com

ABSTRACT

Endpoint congestion in HPC networks creates tree saturation that is detrimental to performance. Endpoint congestion can be alleviated by reducing the injection rate of traffic sources, but requires fast reaction time to avoid congestion buildup. Congestion control becomes more challenging as application communication shift from traditional two-sided model to potentially fine-grained, one-sided communication embodied by various global address space programming models. Existing hardware solutions, such as Explicit Congestion Notification (ECN) and Speculative Reservation Protocol (SRP), either react too slowly or incur too much overhead for small messages.

In this study we present two new endpoint congestion-control protocols, Small-Message SRP (SMSRP) and Last-Hop Reservation Protocol (LHRP), both targeted specifically for small messages. Experiments show they can quickly respond to endpoint congestion and prevent tree saturation in the network. Under congestion-free traffic conditions, the new protocols generate minimal overhead with performance comparable to networks with no endpoint congestion control.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.2 [**Computer-Communication Networks**]: Network Protocols

1. INTRODUCTION

Network congestion is a major performance inhibitor in large-scale high-performance computing (HPC) systems. Due to the lossless nature of HPC networks, a single point of congestion can spread through the network creating an effect commonly called tree saturation [30]. Tree saturation is created by the buffer flow control mechanism between adjacent switches. A full buffer at the input of a switch causes the upstream switch to halt transmission. This in turn causes packets to accumulate in the upstream switch. As the original source of congestion continues to persist, the upstream switch reaches capacity and causes additional switches to halt transmission. Eventually, a tree of congested packets fan out from the original point of congestion to the rest of the network. In a shared system, congestion caused by one application can impact other applications, leading to wide performance variability [5].

Many mechanisms have been developed to address network congestion. Today's HPC networks deploy adaptive routing algorithms to relieve *fabric* congestion caused by poor channel load-balancing [4, 8, 17]. This is particularly critical for high-radix topologies like the Dragonfly [25]. However, few HPC networks have hardware mechanisms for dealing with *endpoint* congestion, which requires admission control at the traffic sources. Without mechanisms to manage endpoint congestion, HPC systems today rely on software-level tuning to reduce the impact of congestion [26, 33].

One notable hardware approach to resolve endpoint congestion is the use of congestion notification. Some Infiniband networks deploy Explicit Congestion Notification protocol (ECN) to signal congestion in the network and reduce traffic injection rate [1]. ECN has been shown to work well for long-duration congestion scenarios [21]. However, congestion notification is a reactive protocol that responds to congestion after it has already occurred in the network. Thus it takes time for ECN to detect and throttle the congestion-causing traffic, leading to slow response time. Study has also shown ECN is highly sensitive to throttling parameters and that a single set of parameters cannot adequately handle all congestion scenarios [29].

More recently, a method for endpoint congestion control called Speculative Reservation Protocol (SRP) has been proposed for large-scale lossless networks [23]. SRP operates on the principle of congestion avoidance, actively combating the formation of endpoint congestion. It uses a lightweight reservation handshake between the traffic source and destination to ensure that no network endpoint is overloaded. To reduce the latency increase associated with the reservation handshake, SRP allows the traffic source to send lossy speculative packets to mask the reservation latency overhead. These speculative packets can be dropped by the otherwise lossless network if they begin to create congestion. SRP has been shown to work well for medium and large message transfers, where the size of the payload is large enough to amortize the cost of reservation control messages. However, HPC networks may not always be dominated by large message transfers.

SC '15, November 15-20, 2015, Austin, TX, USA
© 2015 ACM. ISBN 978-1-4503-3723-6/15/11…$15.00
DOI: http://dx.doi.org/10.1145/2807591.2807600

Most HPC applications today communicate using MPI two-sided communication model. These communications tend to generate large coalesced payloads optimized to achieve maximum MPI messaging bandwidth. However, one-sided communication has become an increasingly popular communication model for future programming systems [13], including many partitioned global address space languages [6, 7, 10, 35]. One-sided communication removes the synchronization bottleneck between source and destination, allowing for more concurrent and potentially more fined-grained communication than traditional models.

Furthermore, with heterogeneous systems dominating the HPC landscape, there is a growing trend to enable the throughput-optimized processors (e.g. GPUs) to access the network directly [28, 31]. The massively parallel nature of these processors tend to favor one-sided, fine-grained communication, with parallel threads individually issuing request to remote memory. This trend will likely shift the traffic characteristics of networks in heterogeneous HPC systems toward smaller messages transfers.

Resolving endpoint congestion caused by fine-grained communication is challenging, requiring both fast reaction time and low overhead. Existing congestion control protocols, such as ECN and SRP, cannot satisfy both requirements. In this work we develop two new endpoint congestion control protocols, Small-Message Speculative Reservation Protocol (SMSRP) and Last-Hop Reservation Protocol (LHRP), that are targeted specifically to endpoint congestion caused by small messages. Both new protocols share similar operating principles with SRP; using reservations to avoid endpoint congestion and allowing the transmission of lossy speculative messages to mask reservation overhead. However, the new protocols are designed to generated minimal overhead when operating with small messages, fulfilling the small message capability gap left by the SRP protocol.

The remainder of the paper is organized into the following sections. In Section 2 we describe the causes of network congestion and the previously studied SRP protocol for preventing endpoint congestion. In particular we analyze its inadequacies on fine-grained communication that motivate this work. The operation of our proposed congestion control protocols are described in detail in Section 3. Network simulator configuration used for evaluation is described in Section 4. In Section 5, we present the performance comparison of the protocols for a variety of synthetic traffic patterns. More in-depth studies relating to the behaviors of the new protocols are presented Section 6. Other works relating to congestion control in HPC networks are discussed in Section 7. We conclude the study in Section 8.

2. MOTIVATION

2.1 Causes of Congestion

A point of congestion occurs in a network when the bandwidth available cannot meet the bandwidth demand of traffic. Congestion can be categorized based on network admissibility of the traffic that generated the congestion. A traffic pattern is considered admissible if the amount of traffic destined for each endpoint is less than its ejection bandwidth, i.e. no over-subscription. Even though an admissible traffic pattern cannot overload any network endpoint, it can still cause *fabric* congestion due to insufficient network bisection bandwidth or poor routing of the traffic. On the other hand,

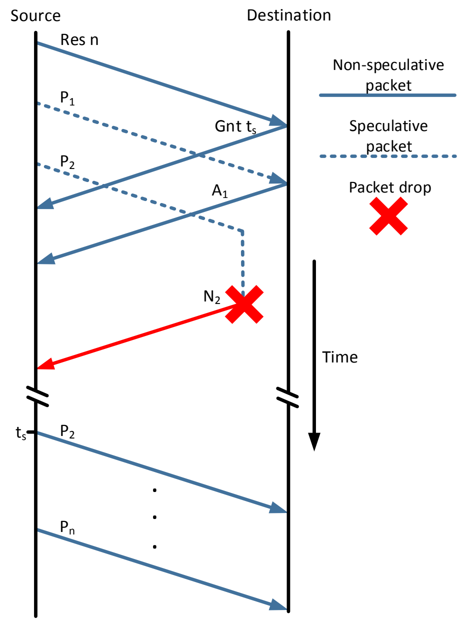


Figure 1: SRP operation diagram

inadmissible traffic by definition will oversubscribe some network endpoints, creating *endpoint* congestion. Because all traffic patterns can be decomposed into their admissible and inadmissible components, network congestion can also be separated into two types that cover all possible congestion scenarios.

While the visible symptoms are the same for both types of congestion, i.e. tree saturation and degraded performance, the optimal mechanisms needed to manage them are different. Fabric congestion can be alleviated by increasing network bisection bandwidth or by properly load balancing traffic among fabric channels through the use of better routing algorithms such as adaptive routing. In the case of endpoint congestion, there are no alternate paths for traffic to route through. Unless the over-subscription of all endpoints can be predicted, over-provisioning ejection bandwidth to prevent inadmissible traffic is unrealistic for a large-scale network. As a result, endpoint congestion requires admission control. The injection rate of traffic sources needs to be reduced to match the available bandwidth at the destination endpoint.

In this study we focus on network protocols that address the problem of endpoint congestion. We assume that the network fabric has sufficient bisection bandwidth and utilizes proper routing algorithms to prevent congestion in the fabric. Therefore the only source of sustained congestion occurs at the network endpoint.

2.2 Existing Endpoint Congestion Control Mechanisms

Current methods of endpoint congestion control in HPC networks rely on congestion notification or is left to the software layers. *Reactive* congestion control mechanisms such as ECN are designed to react to congestion after it has occurred. However, as network injection bandwidth continues to increase and network latency becomes limited by the speed of light, a large amount of congestion-forming traffic can be in-flight before congestion is even detected in the network. As a result, reactive congestion control is becoming

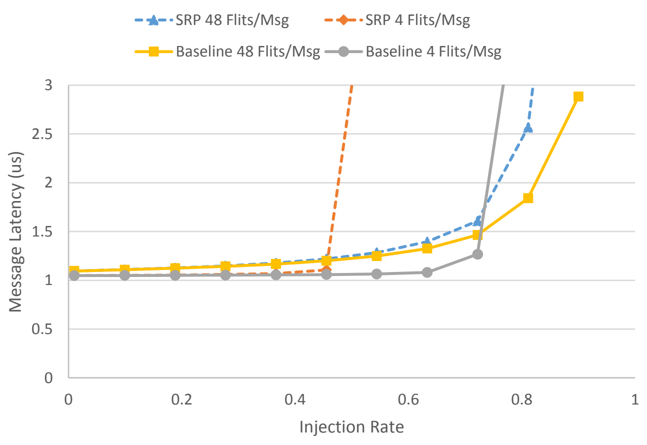


Figure 2: Comparison of SRP's performance on medium and small messages.

increasing ineffective as it suffers from slow reaction time and instability.

Alternatively, *proactive* congestion control mechanisms such as SRP seek to prevent congestion from occurring. The detailed operation of SRP is shown in Figure 1. In this example a network source sends a multi-packet message to a potentially congested destination. The source first sends a small, high priority reservation packet to the destination, *Res n*, indicating the amount of data it intends to transmit. When the destination receives the reservation packet, it returns a grant packet to the source indicating a transmission time, $Gnt\ t_s$.

After issuing the reservation, the source can begin to send data packets, P_1 and P_2, in a *speculative* mode without having received a grant. These speculative packets travel on a low-priority virtual channel (VC) [12], and are only allowed a limited queuing time inside the network before being dropped by an otherwise lossless network. If the network endpoint is congestion-free, the speculative packet will arrive intact at the destination as in the case of P_1. Successful speculative packet transmissions largely mask the latency of the reservation handshake. However, when the network endpoint is oversubscribed, the speculative packets may encounter large queuing delays in the network due to congestion, as in the case of P_2. After a short timeout, the congested speculative packets are dropped by the network and a negative acknowledgment (NACK) is returned to the source, N_2.

When the traffic source receives the grant packet or a NACK, it stops the speculative transmission of that message. When the source has reached its assigned transmission time, it will transmit the remainder of the message in *non-speculative* mode, and retransmit any previously dropped packets. Packets sent in non-speculative mode use a separate higher priority VC than speculative packets. This ensures that any queuing delay experienced by the non-reserved speculative packets do not affect non-speculative packets.

SRP has been shown in previous studies to offer significant improvements over reactive, notification-based protocols. It is able to react quickly to the onset of congestion through the discard of speculative packets. Successful transmission of speculative packets to a congestion-free endpoint also help to mask the latency overhead associated with reservation. The protocol generates little bandwidth overhead for medium to large message transfers ($> 512B$), as the reservation messages can be amortized over several data packets. However, for smaller message transfers, the bandwidth overhead of the protocol can significantly impact network throughput.

Figure 2 shows the latency-throughput plot of a 1056-node dragonfly network running uniform random traffic patterns with two different message sizes. Uniform random traffic does not generate systematic endpoint congestion and consequently a baseline network with no congestion control performs optimally. We used a protocol's performance under uniform random traffic to demonstrate its overhead. For a medium size message with 48-flit payload (each flit is 100 bits), SRP's performance tracks closely with the baseline performance until very high network loads. However, for the much smaller 4-flit message payload, SRP shows almost 30% reduction in saturation throughput compared to the baseline. This significant reduction in throughput is caused entirely by the bandwidth overhead of the reservation handshake. The large number of control packets associated with small messages both increases the overall network utilization and reduces the endpoint ejection bandwidth available to data packets.

The original SRP study sidestepped the overhead issue by allowing small messages to bypass the reservation protocol. This removes SRP's ability to control congestion caused by small messages, and leaves a system dominated by fine-grained communication vulnerable to endpoint congestion. Alternatively, coalescing multiple small message with the same destination into a single reservation can help to amortize the overhead, but can lead to longer latency for messages waiting for coalescing especially at low network loads.

3. FINE-GRAINED ENDPOINT CONGESTION CONTROL PROTOCOLS

Based on our analysis of the shortcomings of SRP, we have developed two new congestion control protocols designed for small message transfers.

3.1 Small-Message Speculative Reservation Protocol

SMSRP is based on the observation that reservation handshake for every message is not necessary if the endpoint is congestion-free. Unlike the SRP protocol that eagerly initiates reservation before every message transmission, SMSRP only issues a reservation after congestion has been detected at an endpoint through the drop of a speculative message.

Figure 3 shows the operation of SMSRP in detail. In this example a network source is sending two small messages, M_1 and M_2, to a possibly congested endpoint. Each message is small enough to fit into a single network packet. When a message is ready, the source immediately transmits it in *speculative* mode. Similar to speculative mode in SRP, speculative messages use a low priority VC and can be dropped by the network when congestion occurs. If the speculative transmission succeeds, such as in the case of M_1 as indicated by its positive acknowledgment A_1, no reservation is required. As a result, when the endpoint is congestion-free, SMSRP generates almost no overhead in the network. [1]

[1]In our experiments we assume that all data packets, even if lossless, require acknowledgments to ensure end-to-end reliability. This is practiced in some HPC networks [1], and are likely to continue in the future to ensure reliability in extreme scale systems.

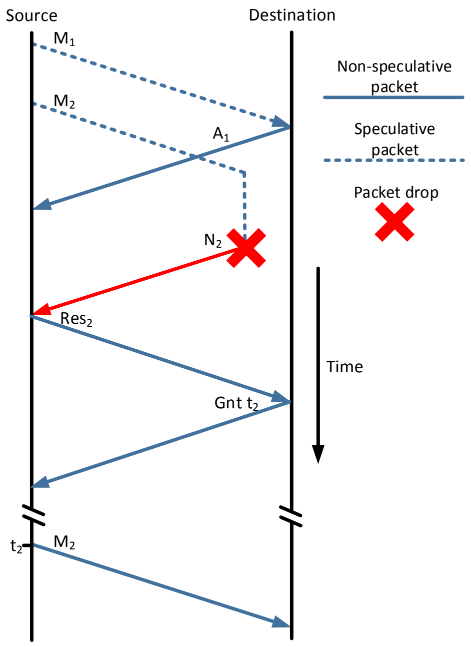


Figure 3: SMSRP operation diagram.

If the network endpoint is congested, after a period of queuing delay, the speculative message will be dropped by the network, as is the case for message M_2. The network switch sends a corresponding NACK back to the source. When the message source receives the NACK, it will initiate a reservation, Res_2, to the destination to acquire a retransmission time for the message. After receiving a grant, $Gnt\ t_2$, from the destination, the source waits until the assigned transmission time t_2 before retransmitting M_2 in *non-speculative* mode. Non-speculative messages are guaranteed to be lossless and use a separate, higher priority VC to ensure that they are not blocked behind speculative messages.

The main attraction of SMSRP, aside from having low overhead in congestion-free network conditions, is that it can be added to a network already implementing SRP with little hardware changes. SMSRP's speculative drop policy and reservation handshakes are identical to SRP. The only change that is needed occurs at the source network interface: changing the ordering of reservation handshake and speculative transmission.

3.2 Last-Hop Reservation Protocol

The eager transmission of small messages in speculative mode reduces SMSRP's control overhead when the network is congestion-free. However, the disadvantage of SMSRP is that when reservations are needed to resolve congestion, the control messages must reach the reservation scheduler at the endpoint. As a result, the reservation handshake messages compete with data packets for ejection channel bandwidth. For small messages, the fraction of ejection bandwidth consumed by control packets can significantly contribute to additional over-subscription of the destination and can lead to more speculative messages to drop. Intuitively, the ejection channel bandwidth is a critical resource that should be preserved only for data packets.

LHRP avoids contributing to the endpoint over-subscription by moving the reservation scheduler from the endpoint to the last-hop switch just upstream of the endpoint. By allowing the switch to service reservation requests for endpoints connected to the switch, we eliminate the control overhead from the ejection channels. Handling reservations at the last-hop switch also opens other optimization opportunities. For small messages, instead of sending separate control packets for the reservation handshake, the reservation and grant information could be piggybacked on the data message itself and the NACK associated with any drop. This reservation piggyback optimization is not useful for SMSRP, because a data packet that has reached the reservation scheduler at the endpoint no longer requires a reservation.

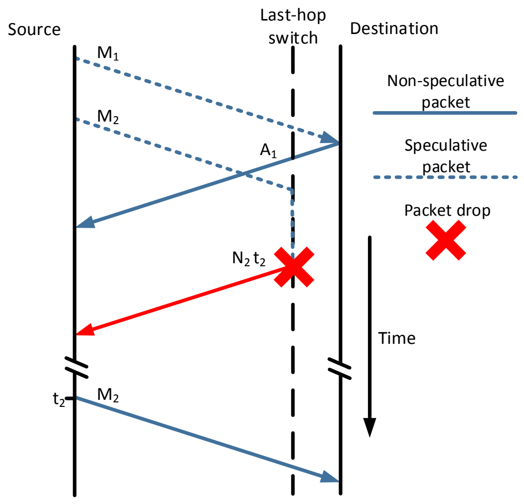


Figure 4: LHRP operation diagram.

The detailed operation of LHRP is shown in Figure 4. Like the example in the previous section, a network source is sending two small messages to a possibly oversubscribed endpoint. Both messages are immediately transmitted by the source in *speculative* mode. If the speculative transmission succeeds, as in the case of M_1, no reservation is required and the protocol does not generate additional overhead.

In the case that the endpoint is oversubscribed, congestion will first occur at the last-hop switch. And it is likely that a speculative message will encounter large queuing delay at the last-hop switch, as is the case for M_2. Unlike SRP and SMSRP, LHRP speculative packets are only eligible for drop at the last-hop switch. We modified the network switch to track the number of packets queued for each endpoint attached to the switch. When the queuing level for an endpoint exceeds a threshold, the switch will begin to drop speculative messages destined for that endpoint. The goal of the queuing threshold is to dynamically maintain a speculative drop rate such that the congested speculative messages do not back-up into adjacent switches, creating tree saturation in the rest of the network.

When a speculative message is dropped by the last-hop switch, it is also assigned a retransmission time by the reservation scheduler in the switch that is returned with the NACK, $N_2\ t_2$. When the source node receives the NACK, it also receives the retransmission time t_2. The source node then retransmits M_2 at time t_2 in a *non-speculative* mode that guarantees lossless transmission.

4. EXPERIMENTAL METHOD

We evaluate the new congestion control protocols using a modified version of the cycle-accurate network simulator

Table 1: Congestion control protocol simulation parameters.

Protocol	Parameter Description	Value
SRP and SMSRP	Speculative packet fabric timeout	$1\mu s$
LHRP	Last-hop queuing threshold	1000 flits
ECN	Inter-packet delay increment	24 cycles
	Inter-packet delay decrement timer	96 cycles
	Buffer congestion threshold	50% buffer capacity

Booksim [11, 22]. Experiments are conducted on a 1056-node dragonfly network with full bisection bandwidth. The network is constructed from 15-port switches, with 4 endpoints, 7 local channels, and 4 global channels per switch. Each dragonfly group consists of 8 switches, and the network has 33 groups. The latency of local channels within a group is set to $50ns$. The latency of global channels between groups is set to $1\mu s$ to simulate long delays expected in a large system spanning hundreds of cabinets. The bandwidth capacity of a channel is 100Gb/s. For routing, we implement the progressive adaptive routing algorithm PAR6/2 for the dragonfly network [20] in order to prevent fabric congestion. The network provides a sufficient number of VCs to each traffic class to prevent routing deadlock.

The network switches are implemented using combined input/output-queued (CIOQ) architecture [9] and use credit-based virtual cut-throughput flow control. At the switch input, each VC buffer is split into multiple virtual output queues (VOQ) to reduce head-of-line blocking. The total buffering of each VC is sufficient to cover a channel's credit round trip latency. The switch output queues have buffering for 16 maximum sized packets per VC. The switch crossbar has a 2× speedup over the network channels. Combined with VOQs, this speedup allows the switch to achieve nearly 100% throughput [32].

We assume the network switches are operating at 1GHz. Switch operations are simulated at the granularity of 100-bit flits (100Gb/s @1GHz). The minimum packet size is a single flit, used for control packets (reservation, grant, ACK, NACK). The maximum packet size is 24 flits. Messages bigger than the maximum packet size are segmented by the source network interface before injecting into the network. All data packets are acknowledged by the destination.

The number of traffic classes in the network depends on the congestion control protocol. The baseline network with no congestion control uses one traffic class for data messages, and one high priority traffic class for ACKs. The Infiniband-style ECN protocol we implemented has the same number of traffic classes as baseline. The LHRP protocol use an additional low priority traffic class for speculative messages. NACKs share the same traffic class as ACKs. Both SRP and SMSRP use two additional high priority traffic classes for reservation and grant. This is to prevent deadlock during the reservation handshake. Other protocol parameters are listed in table 1.

Traffic is generated by network endpoints in the granularity of messages. Message sizes are specified with each experiment. When transmitting messages, network endpoints use a mechanism similar to the Infiniband queue-pairs. The source endpoint creates a separate send queue for each destination and the destination endpoint creates a separate receive queue for each source. Multiple active send queues at a source arbitrate for the injection channel on a per packet, round-robin basis.

The network protocols are tested using three synthetic traffic patterns. Hot-spot traffic, where many network sources are sending to a few destinations, is used to generate endpoint congestion. For congestion-free network tests to measure protocol overhead, uniform random traffic is used. To generate network fabric congestion, we use worst-case traffic patterns for the dragonfly topology (WCn). WCn traffic is defined as, each node in group i sends all of its traffic to nodes in group $i + n \bmod G$, where G is the number of groups in the network. Unless otherwise specified all simulations are run for at least $500\mu s$ to ensure the networks have achieved steady state.

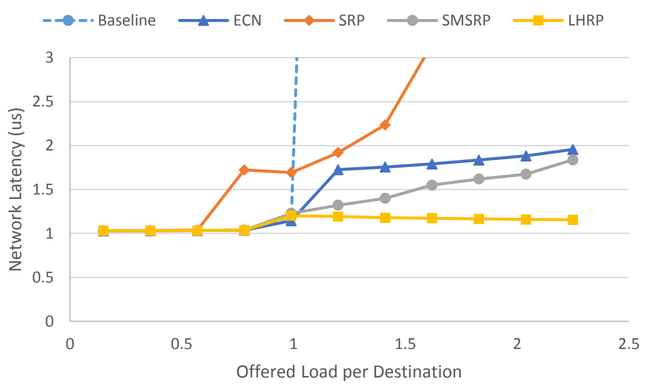
5. RESULTS

5.1 Hot-spot Performance

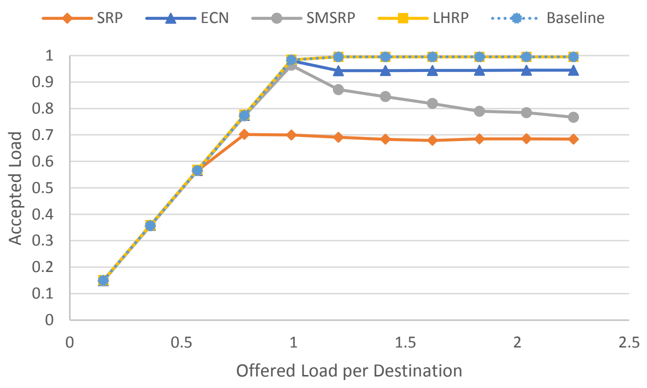
We evaluate the effectiveness of the new congestion control protocols using a 60:4 hot-spot traffic pattern with a message size of 4 flits. Under this traffic configuration, we randomly select 60 nodes in the network to send traffic to 4 destination nodes, while leaving other nodes in the network idle. We can vary the over-subscription factor of the destinations by varying the injection rate of the sources, up to 15× over-subscription. By using a hot-spot traffic pattern with multiple destinations, we are able to capture the effect of any scheduling conflicts such as when a source receives similar reservation times from multiple destinations.

Figure 5a shows the *network* latency of the different protocols as the traffic load increases. Network latency is defined as the time between source injection and destination ejection and does not include source queuing delay. It reflects the level of queuing within the network fabric, a valuable metric to measure the extent of tree saturation. When the traffic load per destination increases beyond 100%, the destinations saturate and congestion occurs. Consequently, in the baseline network with no congestion control the latency grows significantly. This is indicative of tree saturation, where all queues between source and destination are filled and the network is in gridlock. The network using Infiniband-style ECN protocol shows a small increase in network latency after the saturation point, but remains stable as traffic load increase further. This shows that at steady-state, ECN is able to adequately resolve network endpoint congestion. Note that the ECN network latency is higher than our new protocols. This is because ECN requires some congestion to operate: no source throttling occurs unless queues in the network periodically reach the congestion thresholds to trigger congestion notifications.

In a network using SRP, the network latency shows a large increase before 100% load. This is caused by the bandwidth overhead of the reservation handshakes which adds to destination traffic load. At even higher network load, the reservations themselves could create congestion, which SRP cannot prevent.



(a) Network latency



(b) Data throughput

Figure 5: Network performance of 60:4 hot-spot traffic with 4-flit messages.

Unlike SRP, our new protocols both have low network latency after the point of saturation, indicating that both networks remain tree saturation free despite endpoint congestion. As the traffic load increases further, the network using SMSRP shows an upward trend in latency. This is because at higher load, more speculative messages are dropped due to congestion. This in turn triggers sources to send reservation handshakes to the destination, further increase its load. At the limiting case, every SMSRP speculative message could be dropped and the protocol's performance becomes identical to that of SRP. However, the result shows with low levels of endpoint over-subscription, SMSRP remains an effective protocol.

The network using LHRP shows nearly no change in network latency as the traffic load is increased beyond saturation. This is because when LHRP speculative messages are dropped their reservations are piggybacked along NACKs. The LHRP network does not consume additional bandwidth for a separate reservation handshake. At the limiting case, if every LHRP speculative message is dropped, we expect only a moderate latency increase due to higher utilization of the last-hop switch. However, the traffic load at the destination endpoint will not increase because reservations are handled by the last-hop switch.

Performance differences between protocols can also be seen from a throughput perspective. Figure 5b shows the offered load vs. accepted data throughput for the different protocols. A network's saturation point can be defined as when the accepted throughput can no longer match the offered load. The SRP protocol shows saturation at 70% load. This occurs because 30% of the destination ejection bandwidth is being consumed by the reservation-related messages. In contrast, since both the ECN and baseline networks do not generate additional overhead, they saturate at near 100% load and maintains its data throughput with further load increase.

LHRP is also able to sustain the accepted data throughput at nearly 100% (its plot is overlapped with baseline). This is because the ejection channel bandwidth for the hot-spots are not consumed by control packets. While SMSRP also reaches saturation at 100% load, as the offered load increases further, its accepted data throughput decreases. The declining data throughput is caused by the increasing number of reservation handshakes triggered by dropped speculative messages. At the limit, if every speculative message is dropped, SMSRP data throughput approaches that of SRP.

5.2 Transient Congestion Response

While the previous section shows the effectiveness of the new protocols under steady-state traffic conditions, congestion caused by real applications is likely to be transient, occurring in bursts during the communication phase. A protocol's response time to the onset of congestion is a critical factor in determining its real system performance.

In the transient response experiments we test the protocols using a combination of uniform random and hot-spot traffic patterns. At the start of the simulation we launch a uniform random traffic pattern at 40% load among 992 nodes of the 1056-node network to serve as the victim traffic. After $20\mu s$ in simulated time, we launch a 60:4 hot-spot traffic on the remaining 64 nodes. The injection rate of each hot-spot source is at 50%, resulting in a 7.5× over-subscription of each hot-spot destination. Both traffic patterns use 4-flit messages. We can observe a protocol's congestion response time by measuring the transient impact of the hot-spot on the performance of the uniform random victim traffic.

Figure 6 shows the average message latency of the victim traffic before and after the onset of the hot-spots. These results are the average of 10 simulations using different random seeds. Multiple simulations are required to get an adequate sample size for each transient data point. As expected, the victim traffic becomes severely affected by the hot-spot traffic in the baseline network with no congestion control. When tree saturation is formed around the hot-spot, other traffic flows that share paths with the saturated traffic will also experience significant queuing delays. With adaptive routing, a network can delay the saturation impact for some time by routing traffic through less congested paths in the network. However, adaptive routing will also spread the congested traffic to the adaptive paths, and eventually cause many more paths to saturate in the network.

Similar to the baseline network, the ECN network also shows a latency spike after the onset of the endpoint congestion. This transient experiment demonstrates a fundamental limitation of ECN: network congestion must have occurred for ECN to become active. Packets sent during the ECN activation period will contribute to further increase in network congestion, causing performance impact on other traffic sharing the network. If the hot-spot traffic persists, over the course of several hundred micro-seconds, the initial congestion buildup will be cleared and the latency of the victim traffic returns to normal. This is consistent with our steady-state observations that ECN is effective in resolving

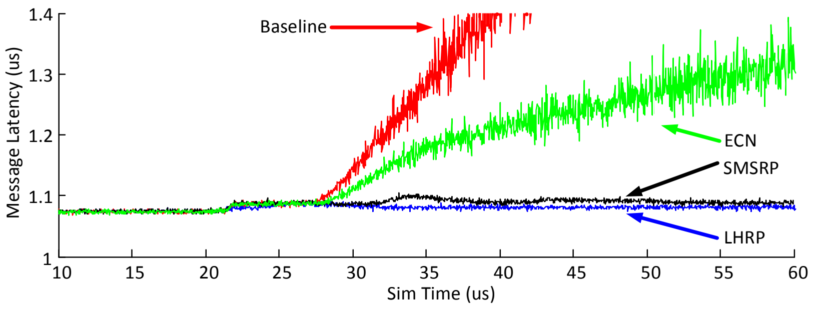


Figure 6: Network transient response to the onset of congestion.

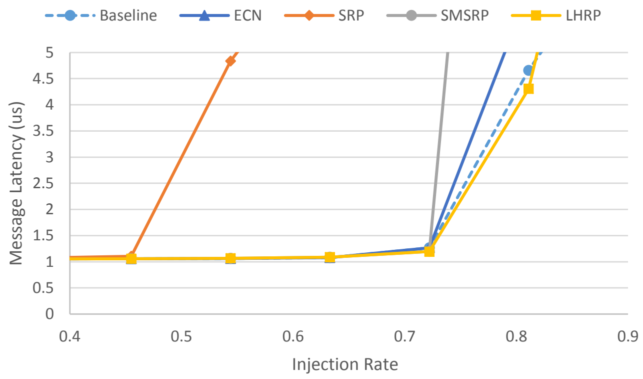


Figure 7: Network performance of uniform random traffic with 4-flit messages.

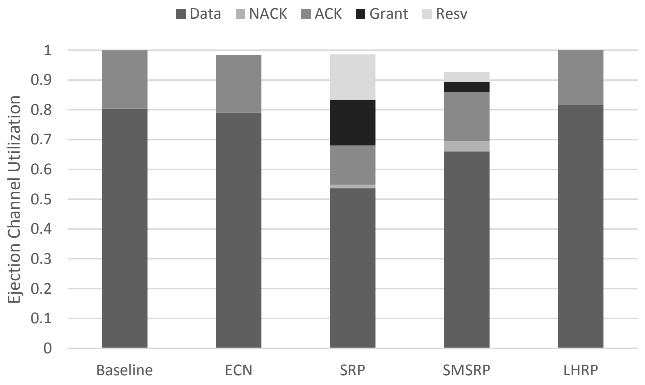


Figure 8: Ejection channel utilization for uniform random traffic at 80% injection rate.

endpoint congestion over a long period of time. However, if the hot-spot traffic burst only occurs for a short period of time, with a duration smaller than the response time, ECN is entirely ineffective in protecting the network from transient congestion scenarios.

By contrast, the victim traffic in the LHRP and SMSRP networks show minimal impact from the hot-spot traffic. On average, the message latency in these two networks is increased by less than 50 ns. This indicates that tree saturation has not formed, and the victim traffic is free to utilize all paths in the network. On closer inspection, the LHRP network shows slightly lower latency than the SMSRP network. This result is consistent with our previous observation that LHRP is more effective in resolving congestion than SMSRP.

5.3 Congestion-Free Network Performance

Overhead generated by a congestion control protocol can be measured by its performance under congestion-free traffic conditions. Figure 7 shows the latency-throughput curves of the different protocols running uniform random traffic with message size of 4-flits. Under congestion-free conditions, the baseline network with no overhead has the optimal performance. In the ECN network, by properly setting buffer congestion thresholds, we can ensure that congestion notifications are not falsely triggered under congestion-free traffic until very high load. As a result, ECN network shows similar saturation throughput as the baseline network. Similar to the result shown in Section 2, SRP performs poorly with small messages. The SRP network saturates around 50% load because of the overhead of the reservation handshakes.

Both LHRP and SMSRP show significantly higher saturation throughput than SRP. This shows that both protocols generate far less overhead when the network is congestion-free. SMSRP shows a slightly lower saturation throughput compared to the baseline. This is caused by overhead generated when speculative messages are dropped in a highly loaded network. LHRP's performance is nearly identical to the baseline, an indication that the protocol creates almost zero overhead even at very high loads.

An in-depth analysis of protocol overhead is shown in Figure 8, demonstrating the utilization breakdown of the network ejection channels for uniform random traffic at 80% load. Both the baseline network and ECN network show that the ejection channel has a data utilization rate of 80%. The remaining 20% of the channel is consumed by ACKs, as we require positive acknowledgment from every packet. In the SRP network, almost 30% of the ejection channel bandwidth is consumed by the reservation-related messages. This overhead is the main culprit that causes the SRP network to saturate almost 30% earlier than the other networks.

In the SMSRP network, NACKs consume 3.5% of ejection bandwidth, showing that some speculative messages are dropped in the network. These NACKs lead to similar levels of reservation handshake messages, creating more network overhead. These control messages are prioritized over data messages for the ejection channel, leading to lower data throughput and earlier network saturation. The SMSRP overhead analysis is consistent with the hot-spot throughput result: as network load increased beyond saturation, the data throughput of the network decreased due to contention with control packets.

The LHRP utilization breakdown looks nearly identical to the baseline, dominated by data and ACKs packets. The data shows that NACKs consumed only 0.2% of the ejection bandwidth, indicating very low speculative message drop rate. Furthermore, a dropped speculative message in LHRP does not generate additional control overhead because the reservation time for the message is piggybacked on the NACK.

Overall evaluation shows that while both new protocols offer significant improvement over SRP, the LHRP protocol consistently out performs SMSRP and creates less overhead. In the following section we provide a more in-depth analysis of the LHRP protocol in various network scenarios.

6. DISCUSSION

6.1 Switch Over-Subscription

A distinguishing feature of the LHRP protocol is that speculative messages are only allowed to drop at the last-hop switch. Since the reservation scheduler for an endpoint is located in the last-hop switch, a packet drop here can acquire a reservation time to be piggybacked with the NACK. However, if a traffic pattern has very high levels of endpoint over-subscription or if a switch has multiple over-subscribed endpoints, it is possible for congestion to occur before the last-hop switch. In general, if the aggregate over-subscription of all endpoints connected to a switch is greater than the number of fabric ports, the switch cannot drop speculative messages fast enough and congestion will form at the fabric channels leading to the switch. In a large-scale network constructed from high-radix switches with 48 to 64 ports [4] [17] [34], we expect traffic scenarios that can overload such a switch to be rare.

Alternatively, the switch over-subscription problem can be resolved by allowing the LHRP protocol to drop speculative messages in the fabric before the last-hop switch. This would operate similar to SRP and SMSRP, where a speculative message can be dropped anywhere in the network fabric after experiencing long queuing delays. Since LHRP does not use separate control messages to acquire reservations, speculative messages dropped outside of the last-hop switch cannot acquire reservations. When the message source receives a NACK without a reservation time, it has to retransmit the message again in speculative mode. If the severe congestion is transient, the second speculative retransmission may succeed or it may acquire a reservation. On the other hand, the message source can detect sustained, severe congestion at the destination, through repeated reservation-less NACKs. It may then coalesce the dropped messages into a single guaranteed reservation similar to the SRP protocol in order to guarantee forward progress. We discuss combining LHRP and SRP protocols in the same network in a later section.

Figure 9 shows the network latency of LHRP under very high levels of endpoint over-subscription. In this experiment the networks are running a 60:1 hot-spot traffic with message size of 4 flits. The network is constructed from 15-port switches. Seven of the ports are used for local fabric channels. For the LHRP protocol without fabric drop, network latency increases significantly when the destination over-subscription is greater than 7×. At this traffic load, all local fabric channels of the hot-spot's last-hop switch are fully subscribed, any additional increase in load causes congestion to occur beyond the switch. By allowing LHRP to drop speculative messages before the last-hop based on queuing delay, the result shows that LHRP is able to maintain lower network latency for much higher levels of over-subscription.

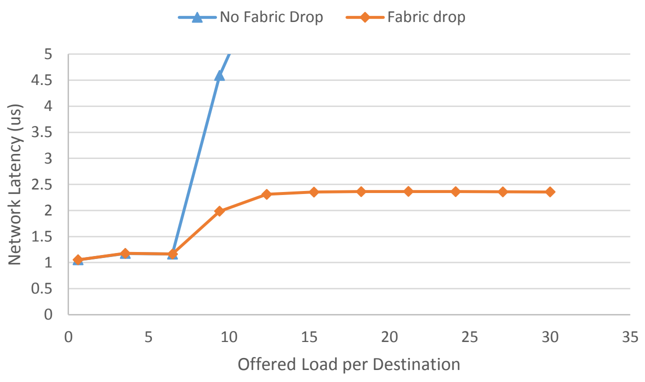


Figure 9: Effect of allowing LHRP to drop speculative messages in the fabric for hot-spot traffic with very high level of over-subscription.

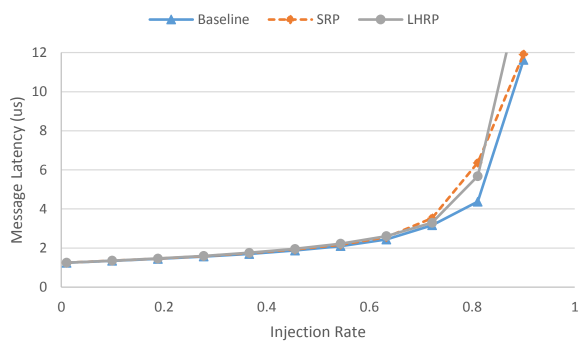
6.2 Large Message Performance

Given LHRP's successful congestion control performance and low overhead on small messages, we explore the possibility of applying the protocol to larger message transfers as well. For messages larger than the maximum packet size, the traffic source segments the message into multiple packets before injection. Each packet can individually traverse the network using the LHRP protocol before being reassembled at the destination.

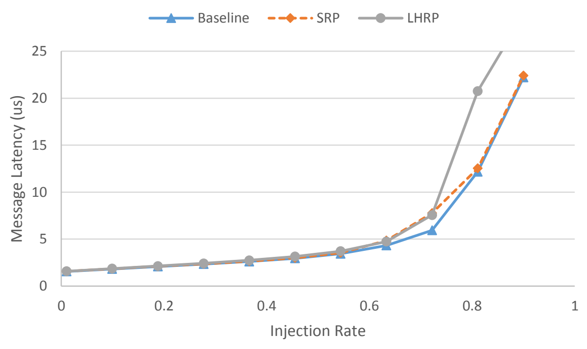
Figure 10 shows network performance of uniform random traffic for message sizes of 192 flits and 512 flits (8 packets and 22 packets). The latency shown in the graph is measured from the time a messages is generated by the source to the time when all packets belonging to the message have been received by the destination. LHRP is compared against a baseline network with no congestion control and a network using the SRP protocol.

For these larger message sizes, SRP shows low overhead and matches the performance of the baseline network in terms of saturation throughput. For 192-flit messages shown in Figure 10a, LHRP also shows low overhead and its performance follows closely with both SRP and the baseline until very high network loads. This result demonstrates that for messages sizes with 192 flits or less, LHRP protocol can be adequate for transferring multi-packet messages as well.

However, for the 512-flit messages show in Figure 10b, LHRP has a 8% lower saturation throughput compared to SRP and baseline. For a multi-packet message, the speculative drop of any one packet will cause an increase in the latency of the whole message. By allowing every packet in a large message to individually undergo the LHRP protocol, which always transmit speculatively at first, the probability that part of the message will encounter speculative drop at high network loads increases. This causes higher message latency and more overhead generated due to NACK and retransmission. In contrast, SRP can issue a single reservation for the entire multi-packet message. At high load this reduces the number of speculative messages in-flight, decreases speculative drop rate compared to LHRP, and results in less overhead. Therefore, beyond a certain message size SRP is preferable over LHRP due to lower overhead at high load.



(a) 192-flit message latency



(b) 512-flit message latency

Figure 10: Network performance of LHRP for large message uniform random traffic.

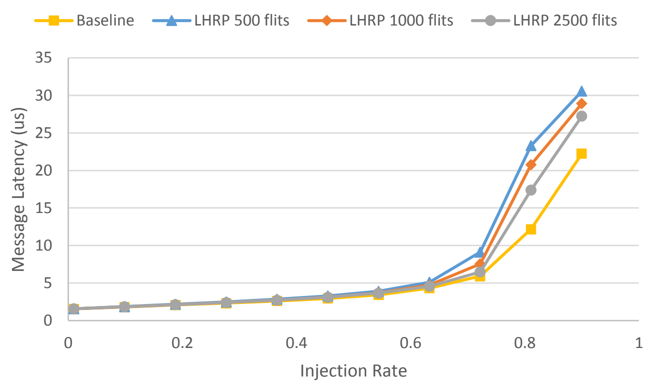
6.3 Last-Hop Queuing Threshold

The single parameter that controls the speculative drop rate of LHRP is the queuing threshold at the last-hop switch. When the number of flits queued for an endpoint reaches the threshold, the LHRP switch will begin to drop speculative packets to restore the queuing level. LHRP's saturation throughput for uniform random, large message traffic can be improved by increasing the queuing threshold parameter. Figure 11a demonstrates the effect of the queuing threshold parameter on the performance of LHRP for uniform random traffic with a message size of 512 flits. The result shows that an increase in threshold causes an increase in the network saturation throughput, and LHRP approaches the saturation throughput of the baseline network. The higher queuing threshold results in fewer speculative packet drops at high load, reducing LHRP's overhead. At the limiting case, with an infinite queuing threshold, no speculative packet is dropped and the LHRP becomes the baseline network.

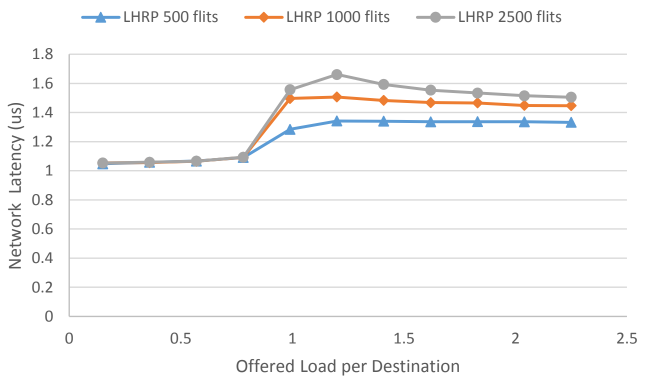
On the other hand, higher queuing threshold worsens LHRP congestion control performance. Figure 11b shows LHRP's performance for 60:4 hot-spot traffic with a message size of 4 flits. Higher queuing threshold results in higher network latency beyond the point of saturation, signifying more queuing delay in the network. If other traffic is sharing the network with the hot-spot traffic, its performance will be more degraded.

6.4 Comprehensive Endpoint Congestion Control

Instead of fine tuning the LHRP queuing threshold to make the protocol suitable for both small and large messages, we can instead create a comprehensive endpoint con-



(a) Message latency of uniform random traffic with 512-flit messages.

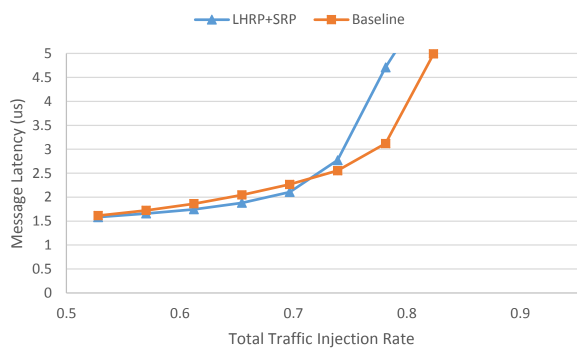


(b) Network latency of 60:4 hot-spot traffic with 4-flit messages.

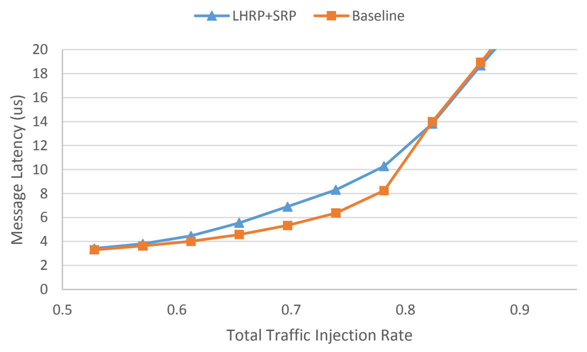
Figure 11: Effect of different last-hop queuing threshold on the performance of LHRP under various traffic conditions.

gestion control protocol that uses LHRP for small messages and relies on SRP for large messages. Figure 10 shows that the SRP protocol consistently outperforms LHRP on large messages. The two protocols can be easily made compatible in the same network. The SRP reservation handshake can utilize the endpoint reservation scheduler in the last-hop switch like LHRP. The speculative packets of both protocols can adopt either the last-hop drop policy of LHRP or the queuing delay drop policies of SRP. During injection, the source network interface can select which protocol to use in the network based on message size.

Figure 12 demonstrates the feasibility and shows the overhead of the comprehensive congestion control protocol. The network is setup using LHRP for messages smaller than 48 flits and SRP for larger messages. The network is running a uniform random traffic pattern with 50% of the data transfered as 4-flit messages and the other 50% of the data transfered as 512-flit messages. Using the comprehensive protocol, small messages in this mixed traffic environment show a minor 5% drop in saturation throughput compared to the baseline network with no congestion control. The loss of saturation throughput is caused by the interaction between speculative small messages and the burstiness of large messages, which causes a slightly higher speculative drop rate. For the large messages, their saturation throughput is unaffected by the small messages and matches closely the baseline network. The result shows there is no signifi-



(a) 4-flit message latency



(b) 512-flit message latency

Figure 12: Performance of combining LHRP and SRP protocols in a network running uniform random with a 50/50 mixture of small and large messages by data volume.

cant interference between LHRP and SRP sharing the same reservation scheduler at the last-hop switch. With further refinement, we expect the comprehensive congestion control protocol will be suitable for all possible network endpoint congestion scenarios.

6.5 Impact of Network Fabric Congestion

The protocols we developed in this work focus exclusively on congestion created at the network endpoints. To combat *fabric* congestion, we rely on previously developed adaptive routing algorithms for the dragonfly topology [20, 24]. In particular, the core mechanisms of LHRP and adaptive routing are complementary and works well together. Adaptive routing uses congestion information such as channel credit count for routing decisions and works independently of the endpoints. Similarly, LHRP only interact with reservation scheduler at the last-hop switch and is largely independent of the paths taken by packets.

Figure 13 shows the interaction of LHRP and adaptive routing in the presence of both endpoint and fabric congestion. In this experiment we use the WC-Hot$n$ traffic pattern: for each group i in the dragonfly network, nodes in group i sends all of its traffic to the same n nodes in group i+1 mod G, where G is the number of groups in the network. For example in the case of WC-Hot2 traffic pattern, 32 nodes in group 0 send all traffic to the same 2 nodes in group 1. This pattern is repeated in other groups of the network. The resulting traffic generates up to 16× over-subscription at the destination endpoints and also overloads the minimal global channel connecting the source and destination groups. All

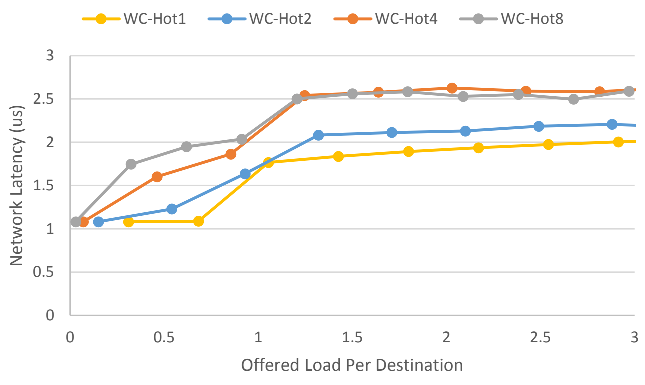


Figure 13: Network latency of a LHRP network running a combination of both hot-spot and worst-case traffic pattern for a dragonfly.

traffic have message size of 4 flits.

Figure 13 shows the network latency as we increase the load of the hot-spot endpoints. The result shows, across all four traffic variations, that after the endpoints enters saturation the networks remain stable and congestion-free. For WC-Hot1 traffic, the network latency plateaus at 2μs, higher than those shown in previous hot-spot simulations in Figure 5a. This higher latency is caused by packets routing adaptively between groups in the dragonfly network. When the load on the single hot-spot within each group increase beyond 100%, the load on the minimal global channel between traffic source and destination also increase beyond 100% triggering adaptive routing. Adaptively routed packets in a dragonfly network takes a longer path resulting in higher network latency. For other traffic patterns with more hot-spot endpoints per group, the traffic load on the minimal global channel between source and destination groups is greater, leading to more adaptive routing. The additional adaptively routed traffic results in even higher network latency observed in these traffic patterns. Nevertheless, in all cases the networks remain stable with further load increase and are free from sustained tree saturation caused by either endpoint or fabric congestion. This demonstrates that LHRP and adaptive routing are working well together in the same system in managing various forms of network congestion.

7. RELATED WORK

The development of SMSRP and LHRP is in part motivated by the SRP protocol proposed in [23]. Together they share many similar operating principles such as endpoint reservation and speculative packet transmission. However, as stated in the original SRP study and demonstrated in our results, SRP works poorly for traffic dominated by small messages due to reservation overhead. The protocols we have developed serve to fill that small-message performance gap and can work in complement with SRP. We have shown that by combining LHRP with SRP in the same network, we can create a comprehensive solution that can work well for any endpoint congestion.

The channel reservation protocol (CRP) proposed by Michelogiannakis, *et al.* is another protocol based on principles similar to SRP [27]. In a CRP network, a reservation is acquired at each hop of the network in addition to the endpoint. CRP is designed to resolve both fabric and endpoint congestion using reservations. The challenge of implementing CRP is that multiple network resources must consent to a common reservation schedule. This significantly complicates reservation scheduling, making CRP prone to reservation gaps when a schedule cannot be agreed upon. Furthermore, CRP also suffers from poor performance with small messages. In this study we take the approach that network fabric congestion and network endpoint congestion should be treated separately. By targeting only endpoint congestion we are able to keep the protocol operations simple and maintain a low overhead, while leaving the task of fabric congestion control to another optimized mechanism such as adaptive routing [20, 24].

The other main class of congestion control mechanisms, such as ECN, are based on congestion notification. Our experiments have shown that ECN is slow to respond to congestion and is potentially unstable. This result is shown in other works as well [23, 29]. Many studies have proposed ways of improving ECN such as changing the way congestion is detected [14] [19], using notification history to predict the rate of congestion build up [3], or using multi-bit notifications to indicate the level of congestion [2] [18]. However, fundamentally these improved ECN algorithms remain reactive in nature. As network bandwidth in HPC systems continues to increase, the amount of congestion-causing traffic that could be in-flight before congestion is detected is ever growing. It is our view that in the future, reactive mechanisms simply cannot match the responsiveness and effectiveness of the proactive mechanisms embodied by our new protocols.

An alternative solution to endpoint congestion control is based on the idea of congestion isolation. This type of protocol stems from the observation that the performance impact of congestion is a result of head-of-line (HoL) blocking by the congested packets. By identifying and separating the congested traffic into its own set of isolation queues, this would alleviate HoL blocking, eliminating the performance impacts. Several works done by Escudero-Sahuquillo, *et al.* proposed such mechanisms for an Infiniband-style network [15] [16]. Congestion isolation protocols can work well in traffic with limited over-subscription and deterministic routing. However, in large-scale networks with multiple network paths and non-deterministic routing, isolating congested traffic to only part of the network is difficult. Furthermore, if there are a large number of destination hot-spots and the network runs out of isolation queues, tree-saturation could once again form in the network.

8. CONCLUSION

Network endpoint congestion caused by small message traffic is difficult to address as it requires both fast reaction time and low overhead. In this study we introduced the Small-Message Speculative Reservation Protocol (SMSRP) and the Last-Hop Reservation Protocol (LHRP), two new endpoint congestion control protocols designed specifically for small message traffic. Both protocols take a proactive approach to prevent congestion formation through the use of endpoint reservations and use lossy speculative transmission to mask the overhead associated with the reservation. SMSRP further reduces protocol overhead by eagerly transmitting messages speculatively and issuing reservations only after congestion has been detected at the destination through the drop of a speculative message. LHRP optimizes for overhead by moving the endpoint reservation scheduler from the endpoint to the the last-hop switch, which preserves the ejection channel bandwidth for data traffic. Furthermore, moving the reservation scheduler to the last-hop switch allows dropped speculative packets to acquire reservations that are piggybacked on negative acknowledgments. This saves LHRP from needing to issue separate reservation messages.

Experiments show that SMSRP and LHRP are highly effective in preventing network tree saturation caused by small-message hot-spot traffic patterns and have a fast response time to the sudden onset of congestion. Under congestion-free network conditions the new protocols incur significantly less overhead compared to the existing reservation-based solution, speculative reservation protocol (SRP). In particular when the traffic is non-congesting, LHRP has nearly no performance penalty. By combining LHRP, which works well for small messages, and SRP, which is optimized for large messages, we are able to create a single solution that is able to excel under any endpoint congestion scenarios.

9. ACKNOWLEDGMENTS

This research was supported in part by the Department of Energy's DesignForward program as part of the Exascale Initiative. We would also like to thank the anonymous reviewers for their helpful feedbacks and suggestions.

10. REFERENCES

[1] Infiniband trade association, infiniband architecture specification, volume 1, release 1.2.1, http://www.infinibandta.com.

[2] M. Alizadeh, B. Atikoglu, A. Kabbani, A. Lakshmikantha, R. Pan, B. Prabhakar, and M. Seaman. Data center transport mechanisms: Congestion control theory and ieee standardization. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, 2008.

[3] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan. Data center tcp (dctcp). *ACM SIGCOMM computer communication review*, 41(4).

[4] R. Alverson, D. Roweth, and L. Kaplan. The gemini system interconnect. In *Proceedings of the 2010 18th IEEE Symposium on High Performance Interconnects*, HOTI '10.

[5] A. Bhatele, K. Mohror, S. H. Langer, and K. E. Isaacs. There goes the neighborhood: Performance degradation due to nearby jobs. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, 2013.

[6] B. L. Chamberlain, D. Callahan, and H. P. Zima. Parallel programmability and the chapel language. *International Journal of High Performance Computing Applications*, 21(3).

[7] B. Chapman, T. Curtis, S. Pophale, S. Poole, J. Kuehn, C. Koelbel, and L. Smith. Introducing openshmem: Shmem for the pgas community. In *Proceedings of the Fourth Conference on Partitioned Global Address Space Programming Model*.

[8] D. Chen, N. A. Eisley, P. Heidelberger, R. M. Senger, Y. Sugawara, S. Kumar, V. Salapura, D. Satterfield, B. Steinmacher-Burow, and J. Parker. The ibm blue

gene/q interconnection fabric. *IEEE Micro*, 32(1), 2012.

[9] S.-T. Chuang, A. Goel, N. McKeown, and B. Prabhakar. Matching output queueing with a combined input/output-queued switch. *Selected Areas in Communications, IEEE Journal on*, 17(6).

[10] U. Consortium et al. Upc language specifications v1. 2. *Lawrence Berkeley National Laboratory*, 2005.

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

[12] W. J. Dally. Virtual-Channel Flow Control. *IEEE Transactions on Parallel and Distributed Systems*, 3(2), 1992.

[13] J. Dinan, P. Balaji, D. Buntinas, D. Goodell, W. Gropp, and R. Thakur. An implementation and evaluation of the mpi 3.0 one-sided communication interface. *Concurrency and Computation: Practice and Experience*, 2013.

[14] J. Duato, I. Johnson, J. Flich, F. Naven, P. Garcia, and T. Nachiondo. A new scalable and cost-effective congestion management strategy for lossless multistage interconnection networks. In *High-Performance Computer Architecture. 11th International Symposium on*, 2005.

[15] J. Escudero-Sahuquillo, P. García, F. Quiles, J. Flich, and J. Duato. Fbicm: Efficient congestion management for high-performance networks using distributed deterministic routing. In *Proceedings of the 15th International Conference on High Performance Computing*, HiPC'08, 2008.

[16] J. Escudero-Sahuquillo, E. G. Gran, P. J. Garcia, J. Flich, T. Skeie, O. Lysne, F. J. Quiles, and J. Duato. Combining congested-flow isolation and injection throttling in hpc interconnection networks. In *Proceedings of the 2011 International Conference on Parallel Processing*, ICPP '11, 2011.

[17] G. Faanes, A. Bataineh, D. Roweth, T. Court, E. Froese, B. Alverson, T. Johnson, J. Kopnick, M. Higgins, and J. Reinhard. Cray cascade: a scalable hpc system based on a dragonfly network. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '12, 2012.

[18] J.-L. Ferrer, E. Baydal, A. Robles, P. Lopez, and J. Duato. Congestion management in mins through marked and validated packets. In *Proceedings of the 15th Euromicro International Conference on Parallel, Distributed and Network-Based Processing*, 2007.

[19] J.-L. Ferrer, E. Baydal, A. Robles, P. Lopez, and J. Duato. A scalable and early congestion management mechanism for mins. In *Parallel, Distributed and Network-Based Processing, 18th Euromicro International Conference on*, 2010.

[20] M. Garcia, E. Vallejo, R. Beivide, M. Odriozola, and M. Valero. Efficient routing mechanisms for dragonfly networks. In *Parallel Processing (ICPP), 2013 42nd International Conference on*, Oct 2013.

[21] E. Gran, M. Eimot, S.-A. Reinemo, T. Skeie, O. Lysne, L. Huse, and G. Shainer. First experiences with congestion control in infiniband hardware. In *Parallel Distributed Processing, 2010 IEEE International Symposium on.*

[22] N. Jiang, D. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D. Shaw, J. Kim, and W. Dally. A detailed and flexible cycle-accurate network-on-chip simulator. In *Performance Analysis of Systems and Software (ISPASS), 2013 IEEE International Symposium on*, April 2013.

[23] N. Jiang, D. U. Becker, G. Michelogiannakis, and W. J. Dally. Network congestion avoidance through speculative reservation. In *Proceedings of the 2012 IEEE 18th International Symposium on High-Performance Computer Architecture*, HPCA '12, 2012.

[24] N. Jiang, J. Kim, and W. J. Dally. Indirect adaptive routing on large scale interconnection networks. *SIGARCH Comput. Archit. News*, 37(3), June 2009.

[25] J. Kim, W. J. Dally, S. Scott, and D. Abts. Technology-driven, highly-scalable dragonfly network. Beijing, China, 2008.

[26] M. Luo, D. K. Panda, K. Z. Ibrahim, and C. Iancu. Congestion avoidance on manycore high performance computing systems. In *Proceedings of the 26th ACM International Conference on Supercomputing*, ICS '12, 2012.

[27] G. Michelogiannakis, N. Jiang, D. Becker, and W. J. Dally. Channel reservation protocol for over-subscribed channels and destinations. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, 2013.

[28] L. Oden and H. Froning. Ggas: Global gpu address spaces for efficient communication in heterogeneous clusters. In *Cluster Computing (CLUSTER), 2013 IEEE International Conference on*, Sept 2013.

[29] G. Pfister, M. Gusat, W. Denzel, D. Craddock, N. Ni, W. Rooney, T. Engbersen, R. Luijten, R. Krishnamurthy, and J. Duato. Solving hot spot contention using infiniband architecture congestion control. In *High Performance Interconnects for Distributed Computing*, 2005.

[30] G. Pfister and V. A. Norton. Hot spot contention and combining in multistage interconnection network. *IEEE Trans. on Computers*, C-34, October 1985.

[31] S. Potluri. Toc-centric communication: A case study with nvshmem, 10 2014.

[32] B. Prabhakar and N. McKeown. On the speedup required for combined input-and output-queued switching. *Automatica*, 35(12).

[33] P. Sack and W. Gropp. Faster topology-aware collective algorithms through non-minimal communication. In *Proceedings of the 17th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '12, 2012.

[34] S. Scott, D. Abts, J. Kim, and W. J. Dally. The blackwidow high-radix clos network. In *Proceedings of the 33rd annual international symposium on Computer Architecture*, 2006.

[35] Y. Zheng, A. Kamil, M. Driscoll, H. Shan, and K. Yelick. Upc++: A pgas extension for c++. In *Parallel and Distributed Processing Symposium, 2014 IEEE 28th International*, May 2014.