Reducing head-of-line blocking in InfiniBand fabrics for Lustre filesystem with Congestion Control

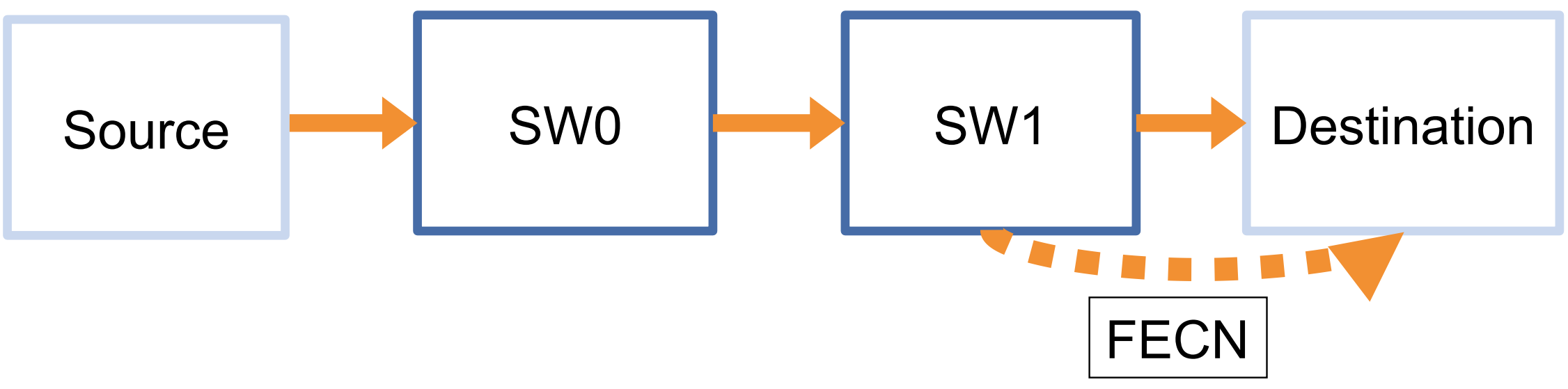
Perry Huang[1], Albert Chu[2]
[1]Institute for Scientific Computing Research, [2]Livermore Computing Division
Lawrence Livermore National Laboratory

Introduction

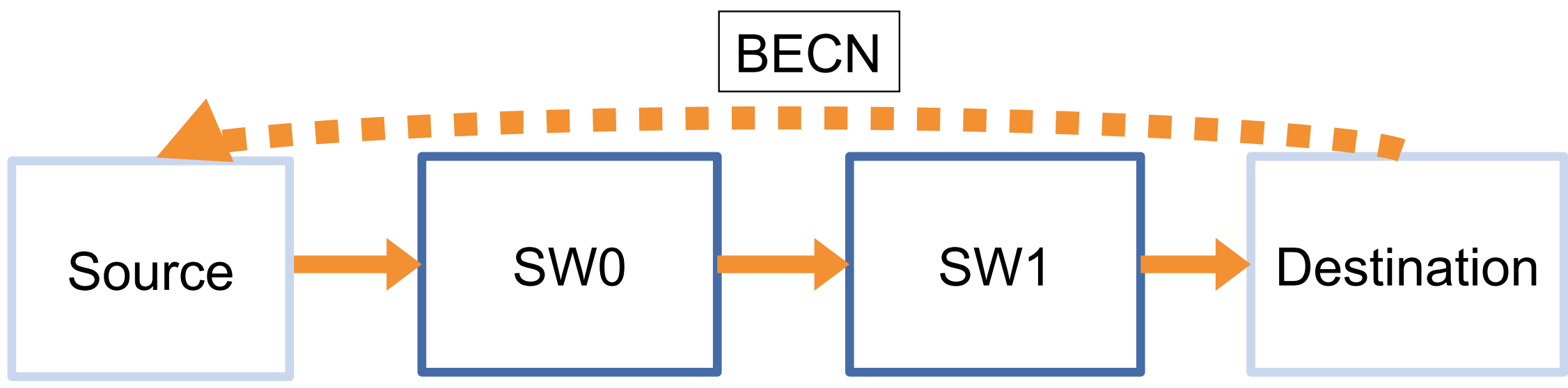
As our supercomputers set new boundaries for computing, we face obstacles in many aspects. InfiniBand (IB), the most popular interconnect technology that is sold as a commodity, has proven to be highly scalable for many systems. Its performance can be further improved upon with the recent introduction of the congestion control feature, if it is tuned properly. We have investigated the congestion control mechanism (CC) and have researched the use of the mechanism by trying to improve the performance of the Lustre filesystem running over InfiniBand. We also investigated the drawbacks of enabling CC and have determined our ideal optimizations.

Congestion Control Mechanism

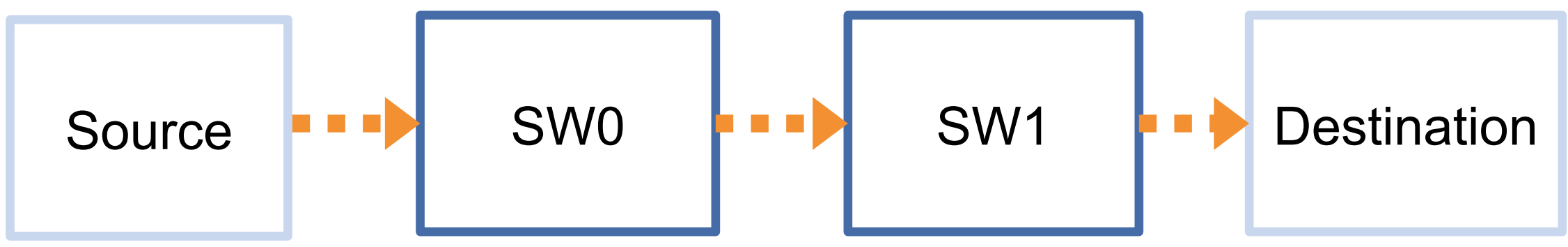
1. Congestion occurs and the first switch that can detect will set a bit in the packet headers, the *Forward Explicit Congestion Notification* (FECN) bit

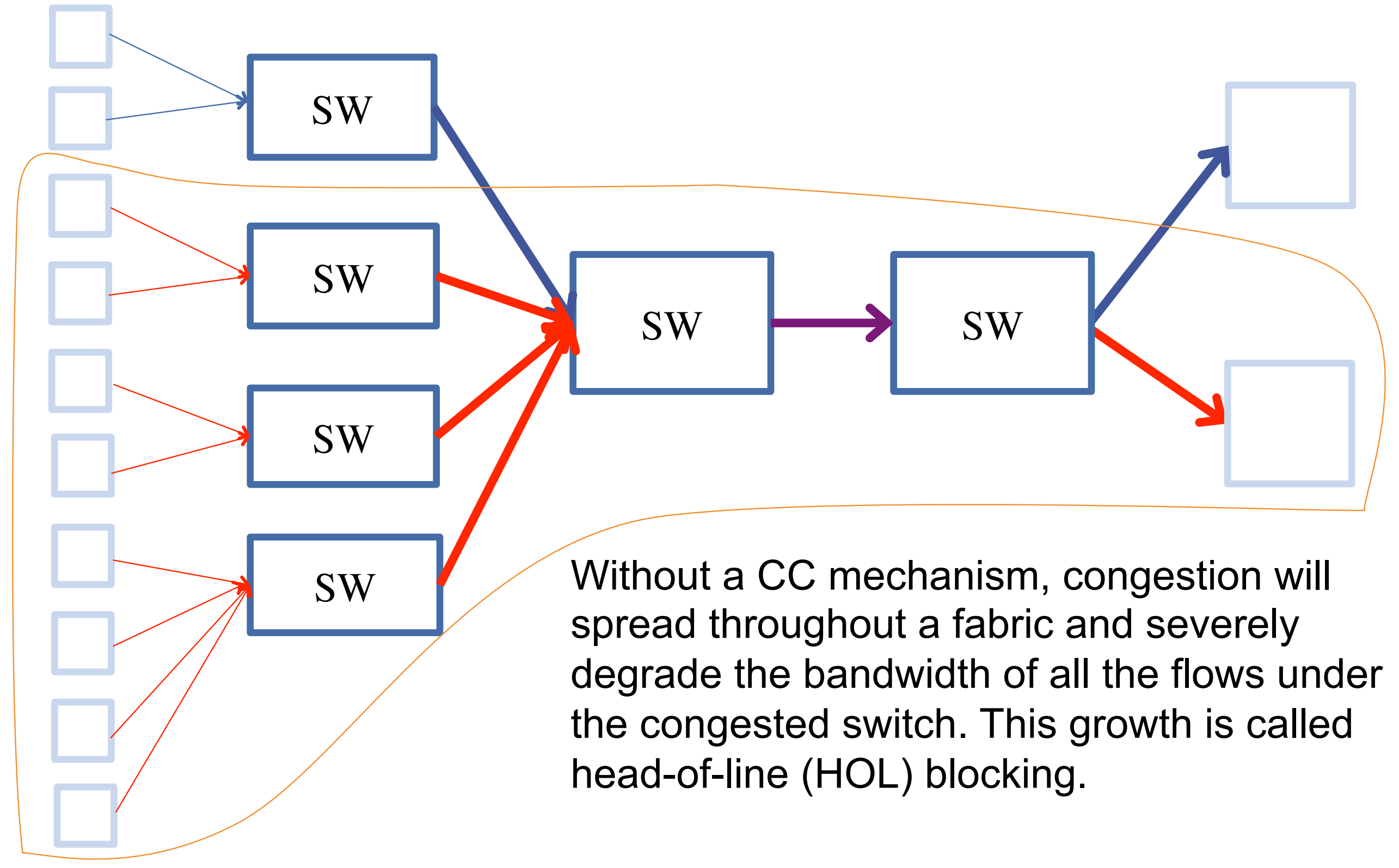


2. The packet reaches the destination node, which will respond by replying with a packet set with the *Backward Explicit Congestion Notification* (BECN)



3. Upon receipt of BECN, the source node will reduce injection rate, which will improve performance by reducing congestion



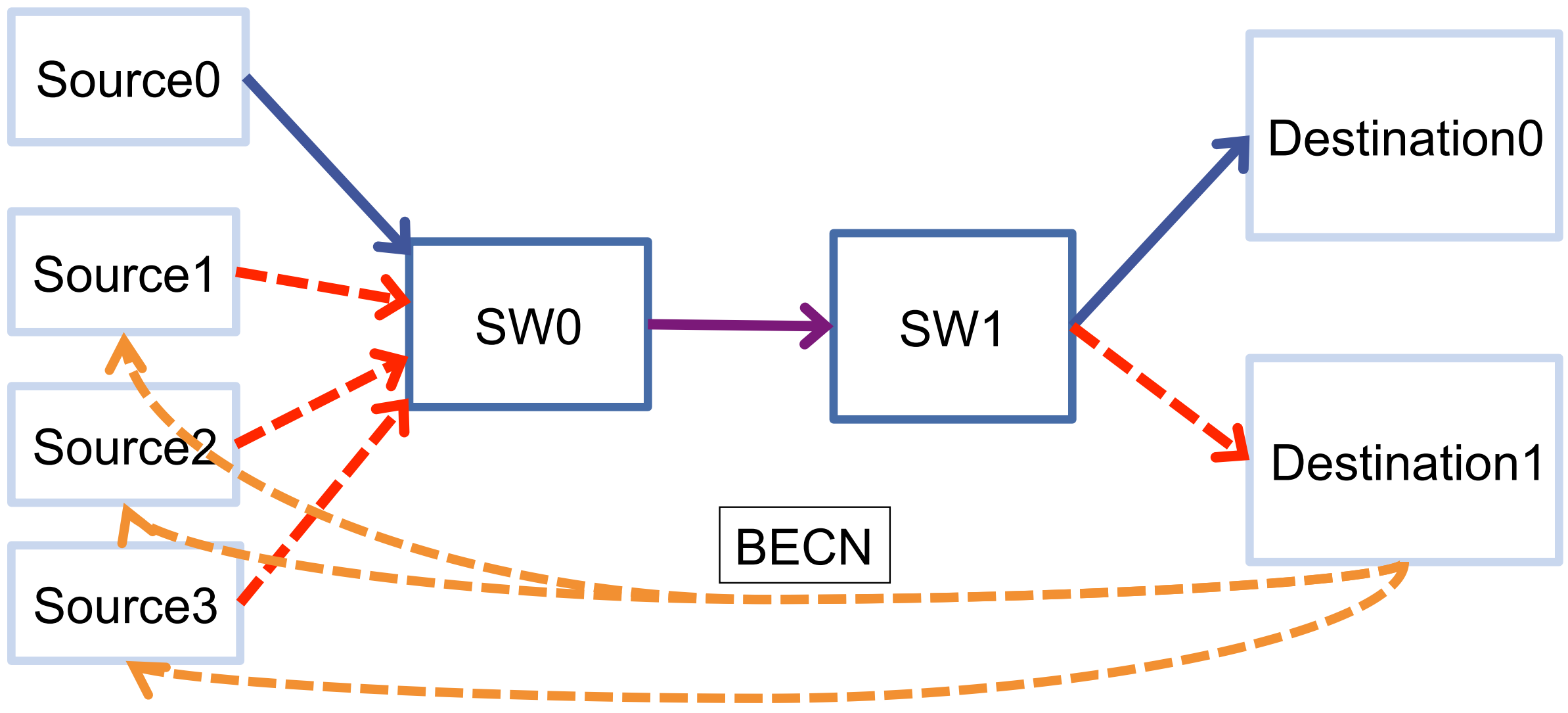


Without a CC mechanism, congestion will spread throughout a fabric and severely degrade the bandwidth of all the flows under the congested switch. This growth is called head-of-line (HOL) blocking.

Experimental Setup

- Six node partition on the Hyperion cluster, each with:
  - Red Hat Enterprise Linux 6.2 with CHAOS patched kernel 2.6.32-220 x86_64
  - 2 * Intel Xeon E5640 2.66 GHz CPU
  - Mellanox ConnectX PCIe 2.0 MT26428 QDR InfiniBand HCA
  - OFED 1.5.4
- 2 * Mellanox InfiniScale-IV switches
- OpenSM 3.3.12 (LLNL branch) subnet manager
- Lustre Networking Self-Test (LNET self-test): performance testing utility

Experimental Overview



Overview of traffic flow with CC that is designed to model the traffic of a running Lustre filesystem. Flows of Source[1-3], in red, contribute to a congested flow between the switches to Destination1. The blue flow from Source1 is known as the victim flow.

Our setup is designed to model Lustre setups:

1. Data transfer occurs in 1 MB sizes
2. Clients do not talk to each other
3. Storage nodes do not talk to each other

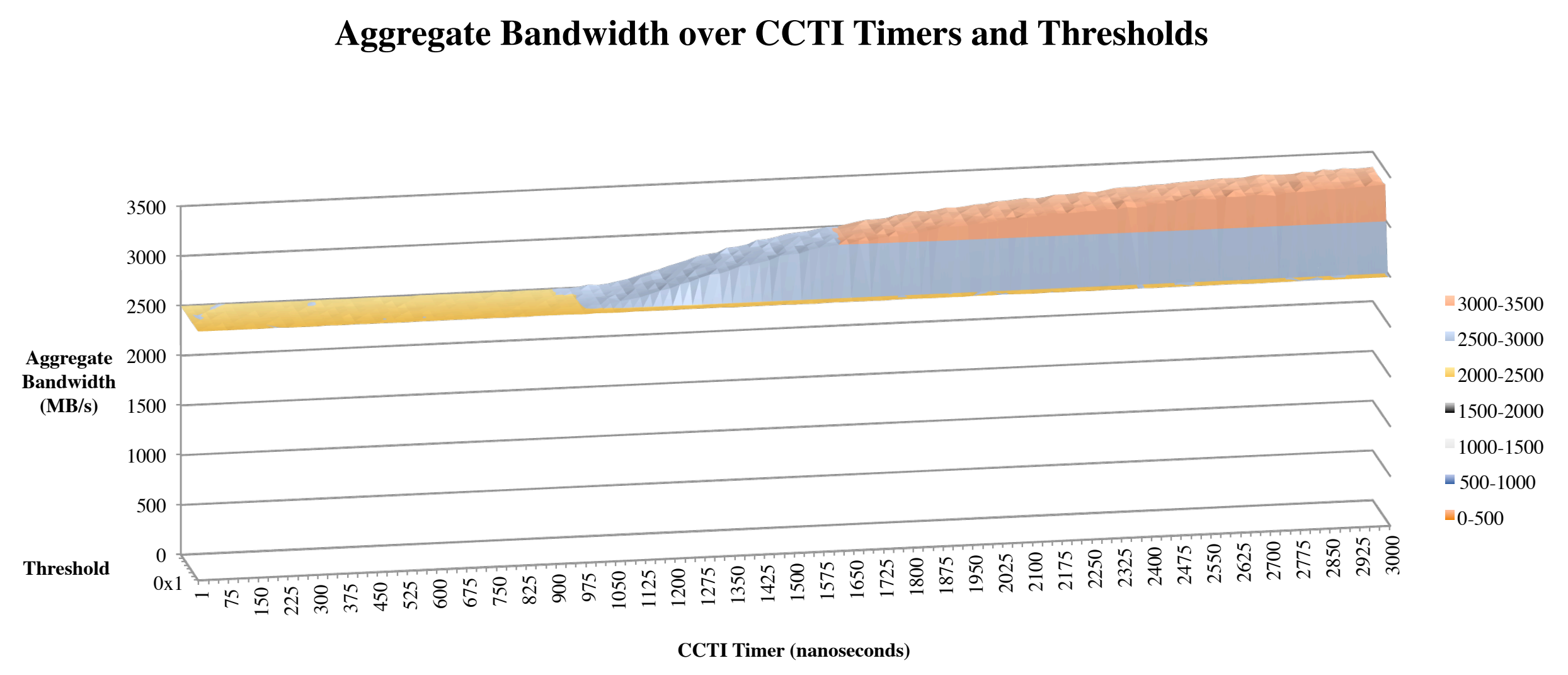
We have limited the theoretical bandwidth of one of the storage nodes to half bandwidth to cause head-of-line blocking. Our experiment will test and optimize congestion control settings for our model to maximize aggregate throughput. Three nodes will be sending data to the limited, congested node, Destination1, and one (victim) will send data to Destionation0. Our model will represent a single point of congestion, which can be scaled to many points on a larger fabric.

Optimization Process

Prior research indicates that two main variables are to be optimized:

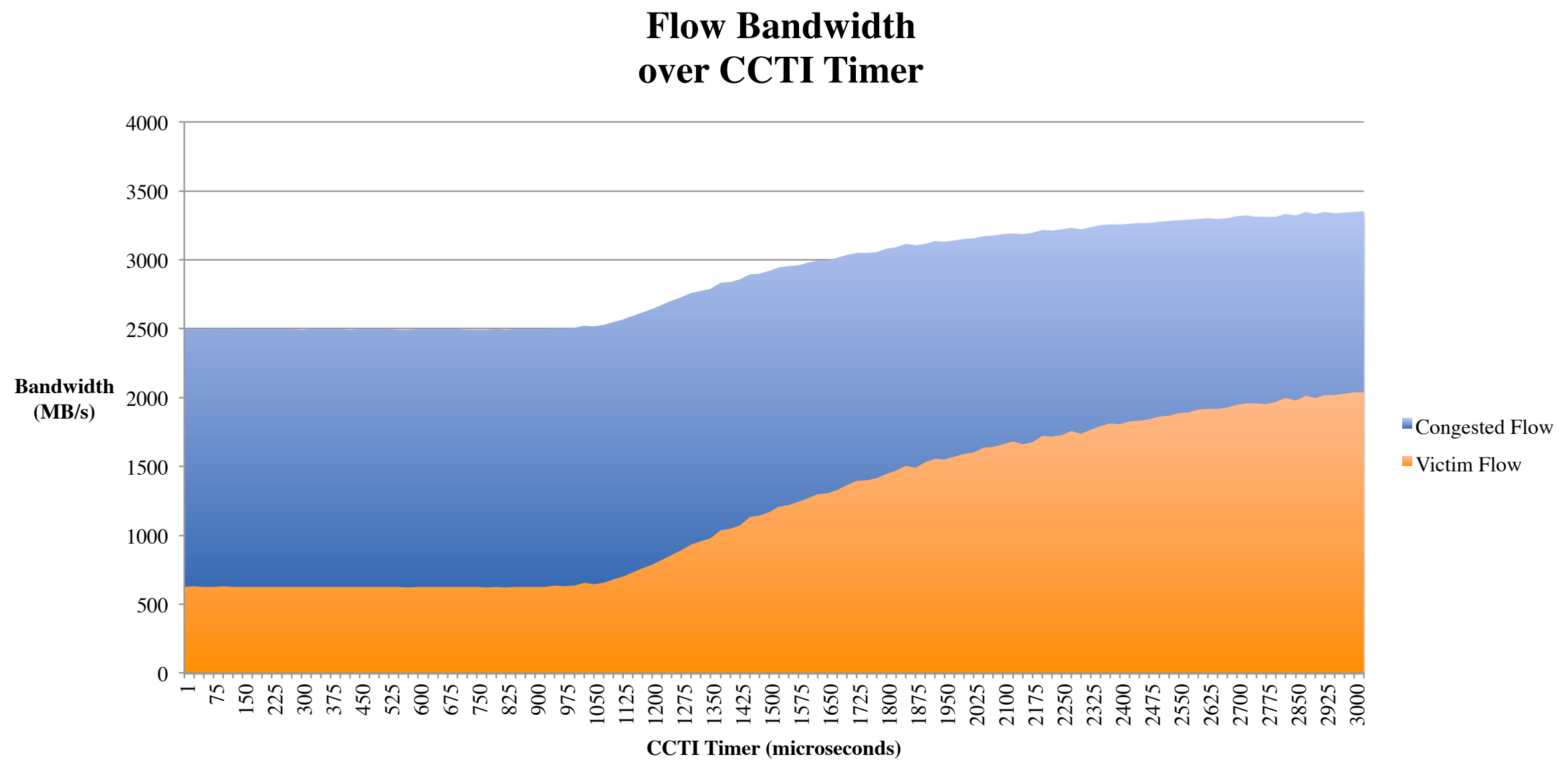
- *Switch threshold*: represents a uniformly decreasing threshold value before a virtual lane is marked as congested
- *Congestion Control Table Increase (CCTI) Timer*: value of time used to periodically decrement the injection rate delay in the Congestion Control Table

Experimental Results



We chose an optimization that provided a balance between maximum aggregate throughput in situations with and without victim flows present.

	Congested Flow (MB/s)	Victim Flow (MB/s)	Aggregate (MB/s)	Congested Flow with no Victim (MB/s)
CC Off	625.92	1870.76	2496.68	1870.16
CC On	1350.20	1668.92	3019.12	1628.67
Δ	+215.15%	-10.79%	+20.93%	-12.91%



Note the losses in throughput of the congested flow as more aggressive CCTI timer values are used.

Conclusion

Ideal variables for Lustre performance study:

- Switch threshold: 0x9 (moderate aggressiveness)
- CCTI Timer: 1650 nanoseconds

Our study has shown that the recently implemented CC mechanism within InfiniBand can be enabled and tuned to mitigate head-of-line blocking and to reduce bandwidth degradation. When fabrics have congested flows with no victim flows, we can minimize the cost of enabling CC on the aggregate bandwidth. This mechanism should carefully enabled by Lustre users to gain very impressive performance increases, as seen in the previous charts.

Ongoing research includes the study of using the CC mechanism to increase overall bandwidth of fabrics used for message passing (MPI) on the large-scale. We are also researching more methods to increase fabric performance, such as dynamic congestion control policies that can be tuned as traffic patterns change, assigning individualized CC settings rather than completely uniform, and more.

Contact

Perry Huang
+1 (925) 423-2912
huang32@llnl.gov
Al Chu
+1 (925) 422-5311
chu11@llnl.gov
Lawrence Livermore National Laboratory
7000 East Avenue
Livermore, CA 94550

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. 638892