Tree-Saturation Control in the AC^3 Velocity Cluster Interconnect

Werner Vogels
Dept. of Computer Science, Cornell University

David Follett
GigaNet, Incorporated

Jenwei Hsieh
Dell Computer Corporation

David Lifka
Theory Center, Cornell University

David Stern
Server Architecture Lab, Intel Corporation

***Abstract* -- In a multi-user production cluster there is no control over the intra-cluster communication patterns, which can cause unanticipated hot spots to occur in the cluster interconnect. In a multistage interconnect a common side effect of such a hot-spot is the roll-over of the saturation to other areas in the interconnect that were otherwise not in the direct path of the primary congested element. This paper investigates the effects of tree-saturation in the interconnect of the AC^3 Velocity cluster, which is a multistage interconnect constructed out of 40 GigaNet switches. The main congestion control mechanism employed at the GigaNet switches is a direct feedback to the traffic source, allowing for fast control over the source of the congestion, avoiding the spread from the congestion area. The experiments reported are designed to examine the effects of the congestion control in detail.**

A. Introduction.

An important issue in traffic management of multistage interconnects is the handling of tree saturation (caused by hot spot traffic[6]), and the impact that tree saturation can have on unrelated flows. In a production multi-user parallel machine such as the AC3 Velocity cluster, this is particularly important as traffic patterns are not predictable, and hot-spots cannot be avoided through application level structuring.

There are two aspects of the handling of the saturation effects that are of primary importance; first there is the fairness among the flows that travel through a region of the switch fabric that contains a 'hot-spot'; flows that cause the congestion should be reduced equally and fairly to relieve the congested link. Secondly there are the effects on flows that are not traveling over congested link, but that do cross switches that are part of the tree that is saturated. Foremost of those effects is second order head-of-line blocking, which can occur even if the individual switches are constructed to handle head-of-line blocking gracefully**.**

This paper describes the GigaNet multi-stage interconnect of the AC^3 Velocity cluster, which is constructed of 40 switching elements organized into a Fat-Tree. A number of techniques are employed in the GigaNet interconnect that control the saturation effects, and that allow the interconnect to gracefully adapt to occurrence of hot-spots. The feedback and flow-control based techniques provide fairness in the scheduling of the competing streams and predictable behavior of unrelated streams that could potentially be impacted by second order effects.

This paper is organized as follows: in sections 2 and 3 the cluster and the interconnect are described in detail. Section 4 examines the problems that are related to saturation in multistage interconnects, and section 5 describes the setup of the experiments to investigate the saturation effects. In section 6 the results of the experiments are presented with conclusions and related work following in section 7 and 8.

B. The AC^3 Velocity Cluster.

AC^3 Velocity is composed of 64 Dell PowerEdge Servers, each of which has four Intel Pentium III Xeon SMP processors running at 500 Mhz with 2 MB of Level 2 cache per processor. Each Power Edge contains 4 gigabytes RAM and 54 gigabytes of disk space. Microsoft Windows NT 4.0 Server, Enterprise Edition, is the operating system. Each rack holds eight servers. The switch fabric is comprised of 40 Giganet cLAN 8x8 switch elements.

The experimental super computer and cluster facility is based at the Cornell Theory Center: a high-performance computing and interdisciplinary research center located at Cornell University. AC^3 is the center's research and IT service consortium for business, higher-education, and government agencies interested in the effective planning, implementation, and performance of commodity-based systems, software, and tools.

C. The GigaNet Interconnect.

The interconnect of the AC^3 Velocity cluster is a multistage interconnection network constructed out of *GigaNet* cluster area network (*cLan*) switching elements and host interfaces.

The host interface provides a hardware implementation of the Virtual Interface (VI)

Architecture specification [1], delivering the interconnect's raw latency and bandwidth directly into application processes, while maintaining full security [3].

A cLan switch is designed using a single chip architecture based on GigaNet's proprietary chip for ATM switching. The first generation GigaNet chip present in the switches that make up the AC^3 Velocity Cluster switching fabric switches at 8x1Gb/sec using a non-blocking, shared memory architecture with 16 Gb/sec cross-sectional bandwidth. The switch uses the memory to implement virtual buffer queue architecture, where cells are queued on a per VCI per port basis. The host interface also implements a virtual buffer queue architecture, where cells are queued on a per VCI basis.

cLAN switches are shipped in eight port 1U and 32 port 2U configurations. These building blocks can be interconnected in a modular fashion to create various topologies of varying sizes. In the AC^3 Velocity Cluster 40 eight port switches are deployed in a fat tree topology as shown in Figure 1. Each stage holds 8 switches, which results in that the maximum number of hops between any two nodes in the system is 5.

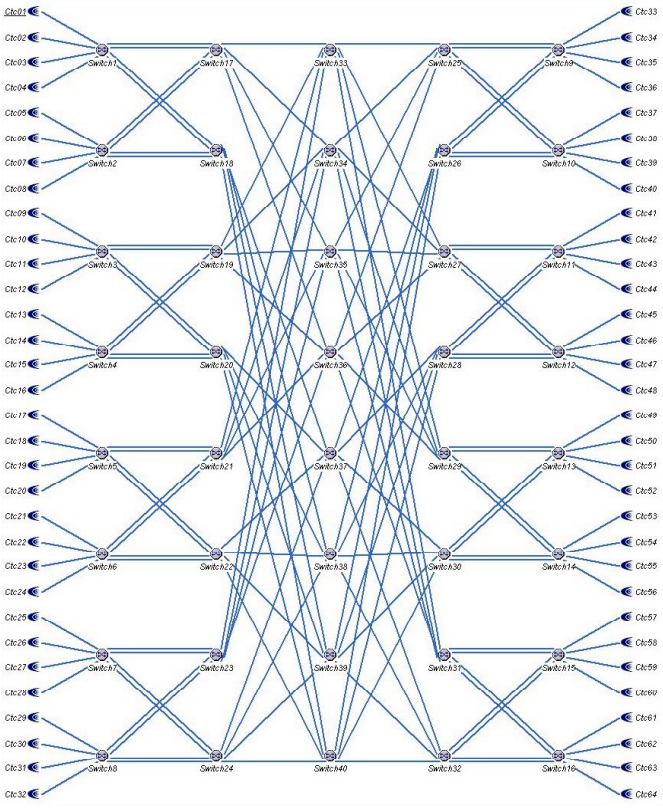

Figure 1. Layout of the switches

The use of ATM for transport and routing of messages, is transparent to the end host. VI endpoints correspond directly to a VCI, using AAL5 encapsulation for message construction, similar to [2]. Switching is performed on a per VCI basis; and no grouping techniques are used at the switch, as flow control policies are implemented on a per VCI basis.

Congestion is evaluated on a per VCI basis, taking into account VCI, link, and general buffer utilization, as well as general system configuration. If flow control is triggered, the switch will start sending Source Quench indications to VCI source, which will respond immediately by shutting down the source until an un-quench indication arrives. The flow control mechanism is implemented in hardware and quenches can be generated at very high frequency. In practice there are always a large numbers of Quench/Unquench indications flowing through the network.

The very high frequency of the flow control indications allows the sources to be bandwidth controlled in a relatively flat manner. It enables the switches that experience potential congestion to schedule the competing streams in a fair manner according to the overall traffic pattern. A second effect of this flow control architecture is that the data sources can be constructed in a simple manner, executing as greedy as possible, relying on the switch flow control indications to perform the traffic shaping.

The clan interconnect is loss-less. A special modification to the clan product allowed flow control to be disabled and replaced with link-level flow control for the purpose of the experiments in this paper.

D. THE PROBLEM.

Interconnect behavior under a variety of realistic workloads has been studied for a long time and this resulted in improved switch and interconnect designs. One of the problems that has been the hardest to solve is that of congestion management in the face of unpredictable traffic patterns.

Feedback techniques [7] such as multi-lane backpressure [5] have been experimented with and the results are promising. The flow-control techniques in a GigaNet based interconnect are novel in that (1) the feedback is directly to the VCI source and not to the predecessor switch in the path, (2) it does not employ any credit based scheme, and (3) that the flow-control is used to perform traffic shaping in the overall interconnect..

The AC^3 velocity cluster provides an excellent opportunity to examine the effectiveness of these techniques given the number of switches in the fabric.

There are three particular problem areas that are of interest, when examining congestion control:

1. **Tree-saturation**. When a switch becomes congested will there be saturation roll-over and spread the congestion to other switches in the region?
2. **High-order head-of-line blocking**. Even if individual switches are constructed such that they exhibit no head-of-line blocking when ports become congested, placing them in a multi-stage interconnect may trigger higher-order HOL occurrences because of link and buffer dependencies between switches [4].
3. **Fairness among congested streams**. If a number of streams flow through single congestion point, will the traffic shaping be such that all streams are treated fairly.

To examine these three problem areas a number of experiments have been designed that are described in detail in section 6. All experiments were performed with the flow control enabled as well as disabled. In the extended summary we report only on the results of the flow-control enabled tests, while the full paper will include a comparison with the flow control disabled tests.

E. BASELINE PERFORMANCE.

In this section we briefly touch on the baseline performance of the interconnect. Standard latency and throughput tests were conducted between sets of nodes in the cluster. Bottom line latency is 10 usec, maximum throughput close to 114 Mbytes/sec and the maximum message rate is over 250,000 messages/sec.

In figure2 a histogram of latency is shown of 16 and 1024 bytes messages in relation to the number of hops between source and destination. Each additional hop adds 1 usec to the latency. Figure 3 shows the average latency with respect to message size.

Figure 4 shows the bandwidth in relation to message size. For bandwidth measurements of single streams, the number of switches in the stream did not matter.

The maximum message throughput is 266,000 messages/sec, which is limited by the host PCI bus, and which is achieved with messages with 4 bytes payload.

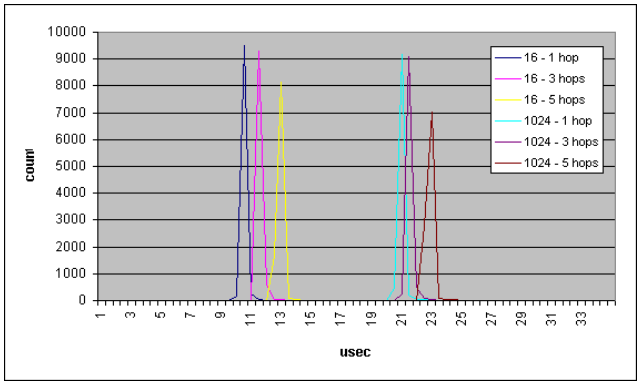

Figure 2. Latency histogram of 16 and 1024 byte message per number of hops.

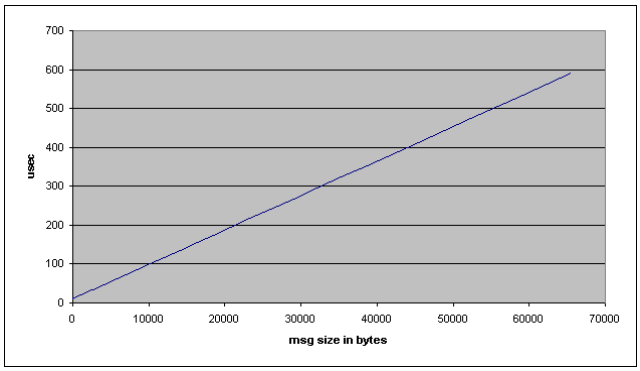

Figure3. Average latency per message size

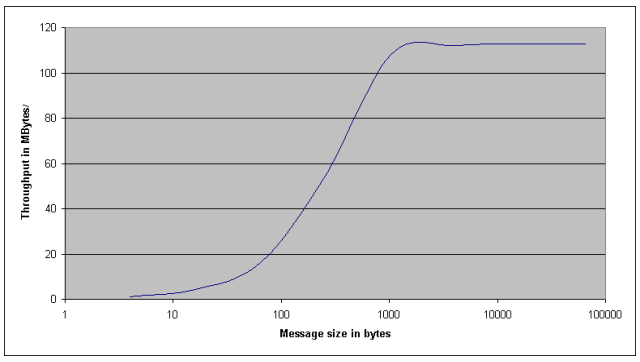

Figure 4. Average throughput per message size

.

F. TREE-SATURATION EXPERIMENTS.

To investigate the effects of tree saturation we conducted four dedicated tests:

Front-to-back. A test where 32 connections are made between random nodes that all cross the maximum number of stages of the interconnect, triggering hot-spots in the communication. For practical

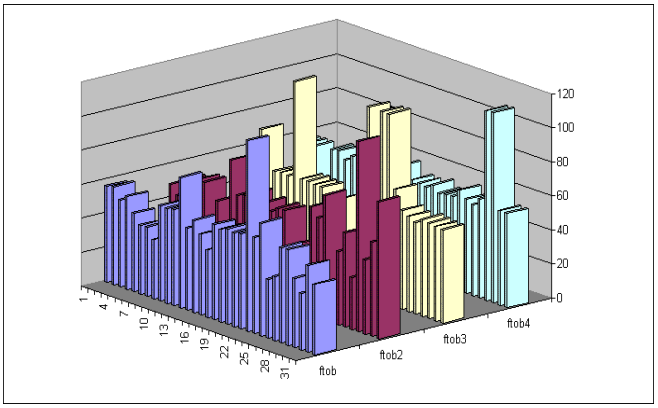

Figure 5. Histogram of the throughput in Kbytes/sec of the individual streams in the 4 different *front-to-back* tests.

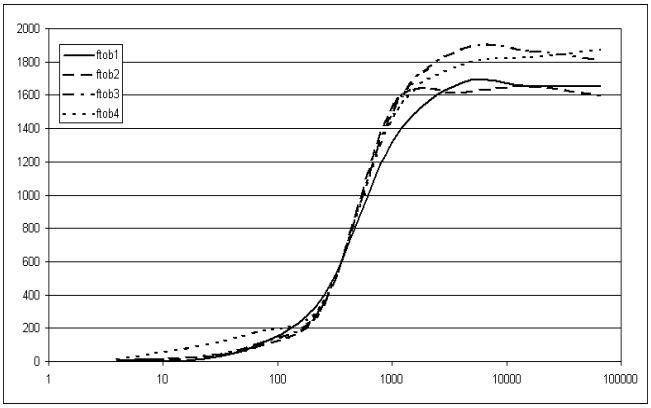

Figure 6. Total fabric throughput in Kbytes/sec per messages size in the *front-to-back* tests

execution of this experiment, all sources are chosen from the nodes 01-32 (the front) while destinations come from nodes 33-64 (the back). Four different connection layouts were tested with a variety of messages sizes. Each test was run for 30 seconds and the results were analyzed for variations in inter-arrival rates, in bandwidth over time, in comparative bandwidth among the streams and overall throughput of the interconnect in relation to message size

Given the randomness in the connection setup, some hotspots occur within traffic, while there are also some connections that share no links at all. Figure 5 shows a histogram of the individual stream throughput measured in the four tests.

Figure 6 shows the overall throughput through the interconnect in relation to the message size.

Slow Host Congestion Fabric. This test is used to examine if congestion will spread through the interconnect when a host network interface controller (NIC) becomes congested. In the test up to seven streams will come into node01, each entering a different port on switch01 and exiting on the port connect to node01. The congested NIC will cause switches 01, 17 and 34 to congest, where switch 34 is a 3rd layer switch. Six large streams will also flow through switch 34, each share an input port with the streams directed to node01. In this test the congestion into node01 is varied and its impact on the overall throughput of switch 34 is measured.

Without any traffic directed at node01 each of the stream achieves maximum throughput (112 Mbytes/sec). The streams used to congest the NIC consist of single cell messages (4 bytes payload) and the streams are added stepwise. The first stream reaches the maximum message throughput of 266,000 messages per second, resulting in up to 1.05 Mbytes/sec. This throttles down the background stream that shares the input port with the congestion stream to 107 Mbytes/sec while the other streams remain unaffected.

Adding more streams towards the NIC causes the competing streams to drop towards equal share of the maximum message throughput at the NIC, e.g. with 3 streams each reaches a throughput of 88,000 per second (.35 Mbytes/sec). Each of the background streams that now share an input port with a congested stream, throttle back slightly, but not less then 110 Mbytes/sec. In the test also one congested stream did not share an input port with a background stream, and this stream did not receive any preferential treatment of the streams that did share input ports.

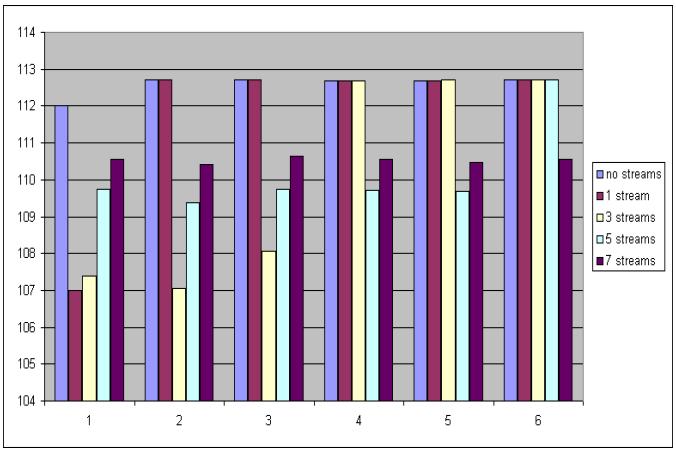

Figure 7. Impact of adding congestion streams to the 6 background stream in *Slow Host Congestion Fabric* test

Switch Port Contention. This test exposes whether contention for a single port on a 3rd level switch will affect other traffic flowing through the same switch. In this test there are seven streams entering switch 40 on ports 2-8, while exiting at port 1. Seven other streams are entering the switch at ports 2-8, but exiting the switch through the same set of ports. The contention of port 1 is varied and the effect on the overall throughput is measured.

Starting without the streams that will congest port 1, the seven background streams all achieve continuously the maximum throughput off 112 Mbytes/sec each. When the congestion streams are introduced their rates are varied from 5%-100% of maximum throughput. At 20% each of the competing streams has reached is maximum throughput of 15 Mbytes/sec, resulting in an output throughput of 91 Mbytes/sec. The background streams have been throttled back to 98 Mbytes/sec (see figure 8).

Increasing the message rates, on the congested streams has no effect, their individual throughput remains at 15 Mbytes/sec. Each of the seven input links continues to run at maximum throughput with a background stream and a congestion stream coming in on each link. The overall throughput in the switch remains at 780 Mbytes/sec independent of how the input streams are varied.

The balance among the streams is close to ideal: the seven background stream throttle back to identical throughput, while the congestion streams each use up 1/7th of the output on port 1.

Multi-stage Congestion. In this test the effects of congestion in a switch on other switches in the fabric is measured, and the fairness among flows through the congested points is examined. For this test there is a set of six sources that each send to both nodes10 and 18, causing contention to occur in switch 36 at port 2 and 3. A second set of sources send to node 18, congesting switch 21 and 5. The traffic into nodes 10 and 18 is varied and the effect on the overall throughput is measured as well at the balance between the individual streams (see figure 9).

The first test is to only send data from the second set of sources, which enter through switch 21 and 5. Jointly they reach a maximum throughput of the 115Mbytes/sec, which is limited by the single link going into node 18. Each stream receives an equal share of the bandwidth (17 Mbytes/sec).

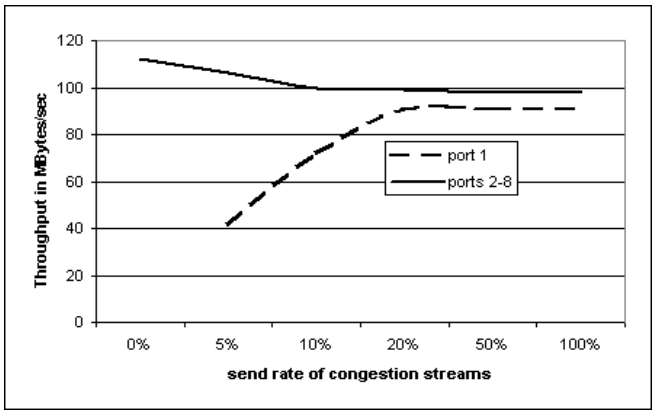

Figure 8. Throughput on each output port in the *switch port contention* test.

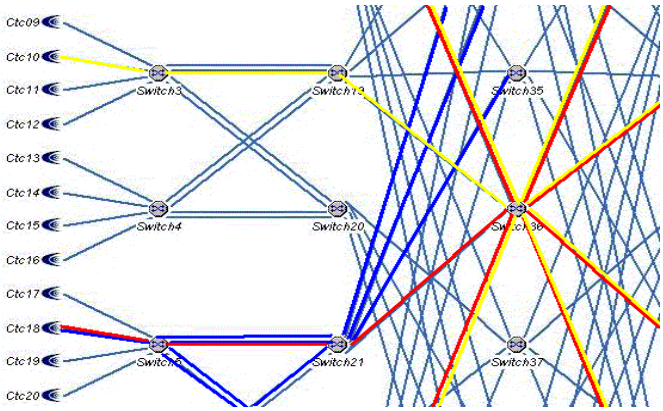

Figure 9. Layout of the multi stage congestion test. The blue stream are the background traffic, the red the congestion traffic, and yellow the side-effect probe

	Node 18 -I	Node 18 - II	Node 18 total	Node 10
Test 1	115	9	115	0
Test 2	115	0	115	114
Test 3 – 5%	52	31	84	0
Test 3 – 20%	49	42	91	0
Test 3 – 100%	61	53	114	0
Test 4 – 5%	61	53	114	30
Test 4 – 20%	62	54	116	64
Test 4 – 100%	60	54	114	114

Table 1. The throughput in Mbytes/sec measured at the destination nodes. Node 18is dived into the set coming from switch 5 & 12, and from switch 36

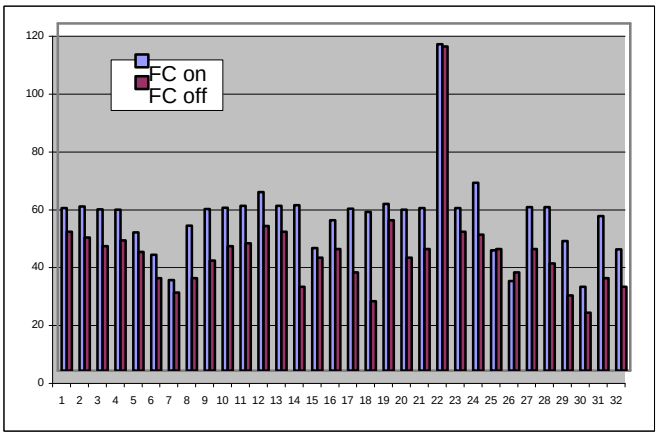

Figure 10. The throughput in Kbytes/sec achieved in a *front-to-back* test with stream based flow control enabled (light bars) and disabled (dark bars).

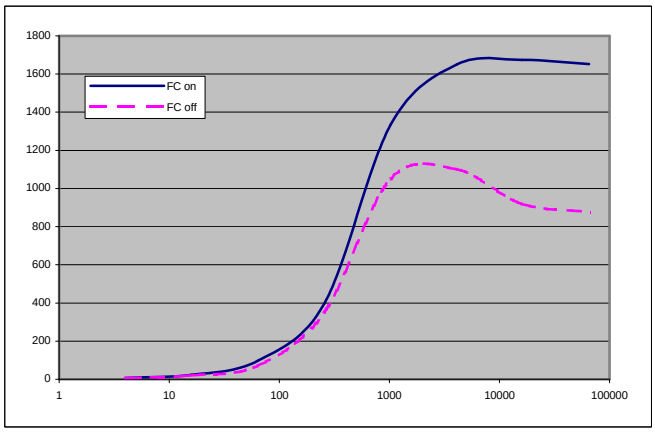

Figure 11. The overall fabric throughput in Kbytes/sec for the first *front-to-back* test with stream based flow-control enabled (solid line) and disabled (dashed line)

Secondly the 6 streams flowing through switch 36 to node 10 are added, and the results show that all streams run at maximum throughput.

In the third part of this test the streams to node 10 are stopped and the additional streams for node 18 coming through switch 36 are introduced, in stepwise manner. The total throughput arriving at node 18 drops to 91 Mbytes/sec when the new streams come in at low rates, while higher rates push the throughput up to 114 Mbytes/sec. The throughput is equally divided over the 13 incoming streams.

The fourth part of this test investigates the impact of this newly congested stream on the traffic that flow through switch 36 to node 10. The congested streams and the streams targeted towards node 10 originate at the same source nodes. The traffic pattern for the streams to node 10 does not change when they have to share the same links with the congested stream, each runs at 19 Mbytes/sec, delivering 114 Mbytes/sec at node 10.

G. Experiments without flow control.

To examine the effectiveness of the per stream flow control mechanism in GigaNet the tests have been repeated with the source-quench flow control switched off. This does not remove flow control completely as GigaNet also employs a link-level flow control.

The results in almost all the tests are identical; as soon as the host interface or a switch port becomes congested this congestion spreads to the other switches in fabric, reducing the overall utilized bandwidth by 50% or more, compared to the bandwidth seen in the case where flow-control was enabled. There was no imbalance noted in the reduction of the throughput over the different streams, suggesting that the scheduling remained fair even under severe congestion. This section presents the results for two tests, *front-to*-back and *switch port contention* which are representative for the observations of all tests.

Front-to-back. In figure 10 the results of the tests with the first front-to-back configuration are presented for the individual tests, with and without per stream flow control. One observation from the first test was that stream originating at node 22, did not share any links with other streams and as such was able to traverse the whole interconnect without any loss in bandwidth (112 Mbytes/sec). Because this stream does not share any links with other streams it is never subject to congestion and link-level flow control is almost as effective as the per stream flow control. All other flows however are experiencing a reduction in throughput as flow-control is exercised on a per link instead of a per flow basis. The non-discriminatory aspect of link-level flow control effect streams at places where they may not be the cause of congestion. The reduction in throughput for this particular test with a 2048 bytes message size is on the average 27%.

In figure 11 the overall throughput of the interconnection fabric is presented in relation to message size. As soon as congestion occurs at switches, which is already noticeable at 512 bytes, the overall throughput is reduced. With all streams running at maximum message size the overall fabric throughput is reduced to 58%.

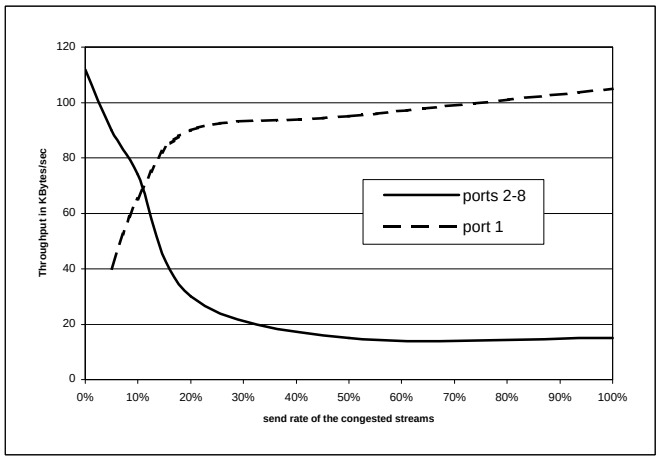

Figure 12. Throughput seen at each of the output ports of the switch, in the *switch port contention* test with stream based flow control off.

Switch port contention. In this test 7 streams enter and exit the switch through port 2-8. Without any competing traffic each of the port outputs the maximum bandwidth. When to each of the input ports an additional stream is added, targeted for port 1, the effect of link level flow control is visible as soon as these streams start. At 15% send rate the flow control kicks in because of potential congestion at switch port 1. Because the flow-control operates at link-level instead of individual stream level, it causes all streams coming in over port 2-8 to be equally reduced. The congestion at port 1 now determines the overall throughput of the switch: each stream destined for port 1 runs at $1/7^{th}$ of the throughput of port 1 (105 Mb/sec), but the other streams are now reduced to run at equal throughput, given the interleaving of cells of both streams at each link, combined with the link-level flow control. This reduces the throughput per outgoing link to 15% of the result with stream based flow control enabled.

H. SUMMARY.

This paper detailed the multi-stage interconnect of the AC3 Velocity cluster. A set of experiments was performed to investigate the effectiveness of the flow control techniques employed by the GigaNet switches and host adapters. The results show that the traffic shaping in face of congestion performs very well: hot-spot regions do not expand beyond the original switch, no higher order head-of-line blocking could be detected and the resulting balancing between streams competing for bandwidth is fair.

These are very important properties in a production switch were there is no advance control over the communication pattern.

The experiments at the AC3 Velocity Cluster continue, with a focus on the impact of non-uniform traffic patterns, impact of the flow control on message latency, the impact of thousands of competing streams and the impact of burstiness in the traffic sources.

I. ACKNOWLEDGEMENTS

This work was performed by the AC3x study group with participation from Cornell University (Theory Center and Computer Science), GigaNet Inc., Dell Corporation and the Server Architecture Lab of Intel Corporation. The Network Analyses were performance on the Velocity Cluster, a high-performance computing resource of the Cornell Theory Center.

The study group is very grateful for the help of George Coulouris and Resa Alvord during the execution of the experiments. Special thanks goes to Shawn Clayton of GigaNet Inc., for the assistance in designing the experiments.

Werner Vogels is supported by the National Science Foundation under Grant No. EIA 97-03470, by DARPA/ONR under contract N0014-96-1-10014 and by grants from Microsoft Corporation.

REFERENCES

[1] Dunning, D. and Regnier, G., *The Virtual Interface Architecture*, Symposium on Hot Performance Interconnect Architectures, Stanford University, 1997.

[2] Eicken, T von, Basu, A., Buch, V., and Vogels, W., *U-Net: A User-Level network Interface for Parallel and Distributed Computing*", In Proceedings of the 15th Annual Symposium on Operating System Principles, Copper Mountain, CO, Dec 1995

[3] Eicken, T von, and Vogels, W., *Evolution of the Virtual Interface Architecture*, IEEE Computer, Nov 1998.

[4] Jurczyk, M., *Performance and implementation aspects of higher order head-of-line blocking switch boxes*, Proceedings of the 1997 International Conference on Parallel Processing, 1997., 1997 , Page(s): 49 –53.

[5] Katevenis, M., Serpanos, D., Spyridakis, E., *Switching Fabrics with Internal Backpressure using the Atlas I Single Chip ATM Switch*, In Proceedings of the GLOBECOM'97 Conference, Phoenix, Az, Nov. 1997

[6] Pfister, G.F. and V.A. Norton, *Hot Spot Contention and Combining Multistage Interconnection Networks*, IEEE Transactions on Computers, C-34(10). 1985.

[7] Scott, S.L.; Sohi, G.S., *The use of feedback in multiprocessors and its application to tree saturation control*, IEEE Transactions on Parallel and Distributed Systems, Volume: 1 4, Oct. 1990 , Page(s): 385 –398.