CDF DAQ UPGRADE AND CMS DAQ R&D: EVENT BUILDER TESTS USING AN ATM SWITCH

G. Bauer, T. Daniels, K. Kelley, P. Sphicas,
K. Sumorok, S. Tether, J. Tseng, D. Vučinić
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, USA

E. Barsotti, M. Bowden, J. Patrick
Fermi National Accelerator Laboratory
Batavia, Illinois 60510, USA

Presented by J. Tseng
E-mail: jtseng@fnal.gov

ABSTRACT

The present data acquisition system of the CDF experiment has to be upgraded for the higher luminosities expected during the Run II (1999+) data-taking period. The core of the system, consisting of a control network based on reflective memories will remain the same. The network used for data transfers, however, will have to be changed. We have investigated ATM as a possible replacement technology for the current Ultranet switch. We present preliminary results on this new ATM-based event builder system.

1. Event Building at CDF

The Collider Detector at Fermilab [1] (CDF) is a general purpose particle detector which has taken over 100 pb^{-1} of data at the Fermilab Tevatron since 1987 and is scheduled to take data again in 1999, accumulating well over 10 pb^{-1} per week. To take advantage of the high luminosity of the upgraded Tevatron, the three-level trigger heirarchy will be preserved, where the first two levels, implemented in hardware, will reduce the event rate from 7.6 million events/s to about 300 events/s (up to 1000 events/s) which are then assembled and analyzed by the Level 3 trigger. The average event size will be about 150 kB, assembled from about a dozen sources, the fragments ranging from several kilobytes to about 16 kB. The largest fragment size is expected to be 32 kB; to transfer such a fragment at 300 events/s requires that an individual link sustain traffic of 10 MB/s. [2] A promising technology for building such events is ATM, which is also being investigated for use in the CMS experiment. [3] This article reports on preliminary studies conducted at CDF with an ATM-based event builder test system.

2. The Event Builder Test System

Figure 1a shows the conceptual architecture of the CDF DAQ system with its separate command and event networks. The Run Ib (1994–96) event network was a commercial network called Ultranet; it is this network for which an ATM switch

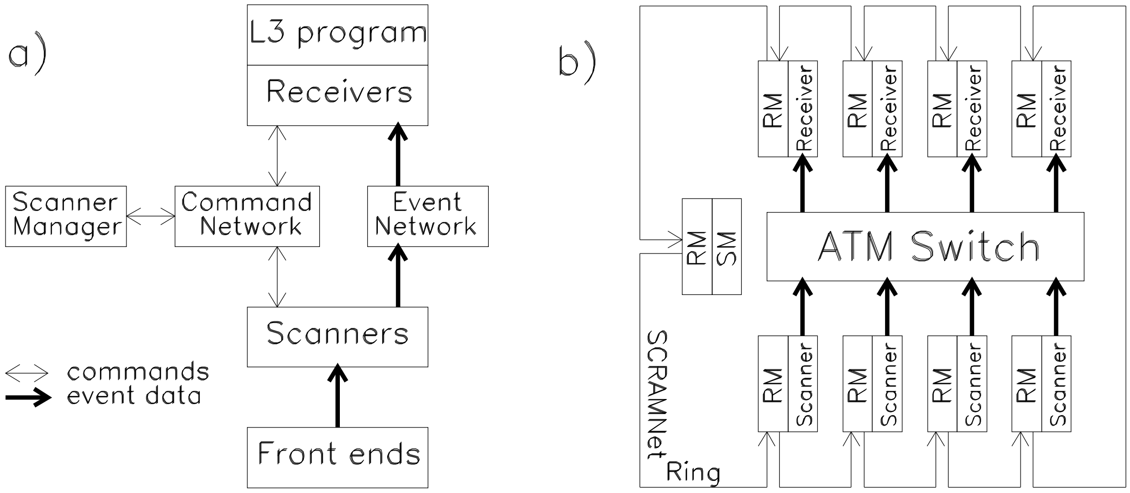

Fig. 1. (a) Conceptual architecture of the CDF data acquisition system. Data flows up from the front ends under the control of the Scanner Manager. (b) The event builder test system. The eight computers connected to the ATM switch are four Motorola MVME1603's and four Radstone Technology RS603's. The Scanner Manager (SM) computer is a Motorola MVME1604. The SCRAMNet ring connects the VME reflective memory (RM) modules.

is being investigated as a replacement. The test system for the upgrade is built around a FORE Systems ASX-1000 non-blocking ATM switch with 13 K cell output buffers. [4] The switch is currently equipped with eight 155 Mbps input/output ports but is expandable to up to 64 such ports. "Non-blocking" refers to the fact that the switch's internal bandwidth accomodates the maximum input bandwidth, even when the switch is expanded. Each port is connected to a PowerPC-based VME single-board computer running VxWorks 5.2; these computers can act as either Scanners, which in the real DAQ system read the detector front ends and send the event fragments through the event network, or Receivers, which combine the fragments into events for the Level 3 system. A $4 \rightarrow 4$ system is shown in Figure 1b. The ATM interfaces are Interphase 4515 PCI-ATM adapters with 128 kB on-board RAM ("packet RAM"). A ninth PowerPC computer is used as a Scanner Manager and is connected to the other computers via the command network, in this case a Systran SCRAMNet ring of VME reflective memories.

An ATM address consists of a virtual path identifier (VPI) and a virtual circuit identifier (VCI). In the test system, the VPI functions as a physical address, uniquely assigned to individual computers, while the VCI addresses one of several event building buffers on a given Receiver. A "virtual connection" connects each Scanner with each event building buffer.

The CDF event building software from Run I has been ported to this system and adapted for test purposes. This setup makes possible realistic performance measurements such as data and event throughput and investigations of cell losses, as well as

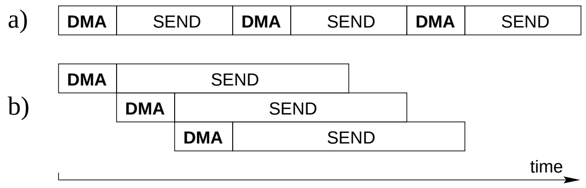

Fig. 2. ATM driver send operations, "slow" (a) and "fast" (b) versions.

refinements as necessary or advantageous.

3. ATM Interface

The ATM interface driver software used in the present system uses the bare AAL5 protocol with a hardware-calculated CRC in order to investigate system performance without the overheads of lossless protocols such as TCP/IP. The driver software's in-house development also allows for flexible configuration and eases optimization.

One optimization concerns how data is sent using the interface card. A send operation consists of two steps: first, the packet to be sent is loaded using DMA from the CPU's local memory into the packet RAM. The board is then instructed to send the packet, whereby the hardware takes over the operation, fragmenting the packet into ATM cells, packaging the cells in SONET frames, and sending the frames over the optical link. If the two steps are performed separately, as shown in Figure 2a, the output link is not used during the DMA step, which then contributes to transmission overhead. The DMA throughput has been measured to be 37.3 MB/s in the absence of other DMA operations, whereas the theoretical maximum payload throughput over the optical link, counting ATM and SONET overheads, is 17 MB/s; the theoretical maximum payload throughput of the "slow" send operation is therefore 11.7 MB/s. The marginal payload throughput is measured to be 11.2 MB/s, with the actual throughput using 32000-byte packets at 10.5 MB/s.

It is possible to hide the DMA overhead by loading one packet while sending another as shown in Figure 2b. This transmission mode requires multiple output buffers in the packet RAM. The output hardware then interleaves cells from buffers being sent to different ATM addresses; sending two buffers to the same address causes the second buffer to wait on the first. With this "fast" transmission mode, which is used in all the subsequent tests, the payload throughput is measured to be 16.2 MB/s with very little overhead.

4. Direct Driver Tests

The most basic tests involving the ATM components are those in which N_{send} computers perform uncoordinated rapid-fire packet transmissions to each of N_{rec} receiving computers. The number of packets moving through the switch at a given

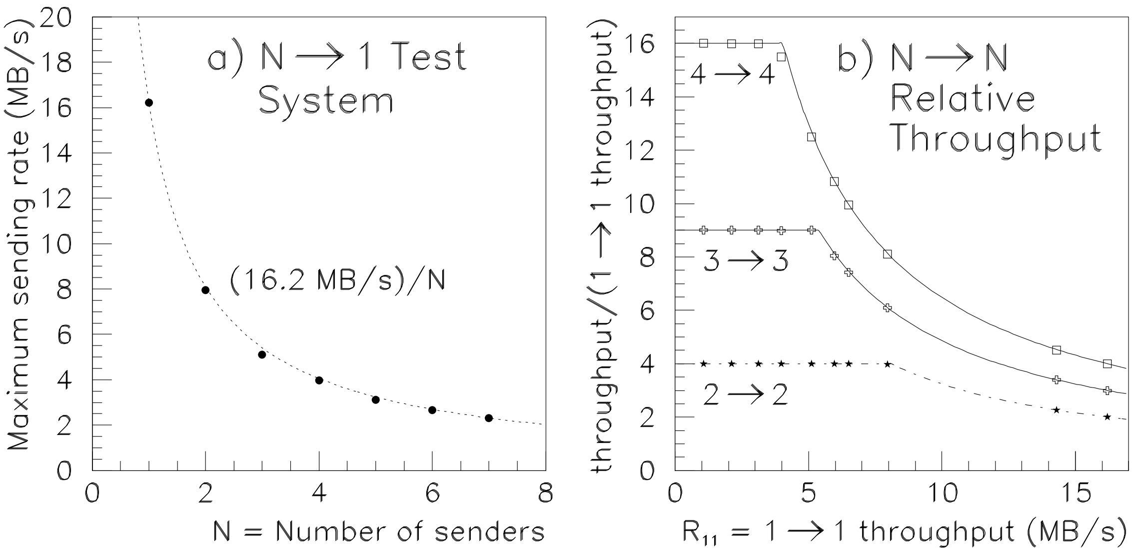

Fig. 3. (a) Maximum sending rate for N_{send} equivalent senders for one to seven senders to one receiver. The curve is the theoretical maximum, above which cells will be lost. The points occasionally lie below the curve because of the coarse-grained rate limit control. (b) Relative throughput (to R_{11}, the $1 \to 1$ data throughput) vs. R_{11}, for $N_{send} = N_{rec}$.

time is therefore $N_{send} \times N_{rec}$. The driver is called directly, and the control network is ignored. These tests therefore reflect the best possible throughput performance.

One obvious issue in this setup with multiple senders and receivers is that if several senders send data to a single receiver faster than it can be received, the ATM switch will simply drop the overflowing cells. However, the interface can be instructed to restrict its own sending rate on any given virtual connection by setting a hardware prescale counter. If the maximum reception rate is v_{max}, one naively expects that the maximum sending rate from equivalent senders will be v_{max}/N_{send}, above which cells will be lost, and indeed this is seen to be the case in Figure 3a.

Since the rate limit is implemented per virtual connection, a sender's unused bandwidth can be used to send to other receivers in an $N_{send} \to N_{rec}$ system. The total throughput of the system should therefore scale as $N_{send} \times N_{rec}$. This scaling behavior is shown for $N_{send} = N_{rec}$ by the plateaus in Figure 3b. The falling relative throughput for $R_{11} > v_{max}/N_{rec}$ is due to having saturated the senders' ATM links. Furthermore, no cells were lost at any set rate limit. It was also confirmed that the same data was received as was sent.

5. Traffic Shaping

In the direct driver tests, there is only one virtual connection between each sender and receiver. However, as noted previously, the event builder system allocates a different VCI to each event building buffer so that one event may be built while

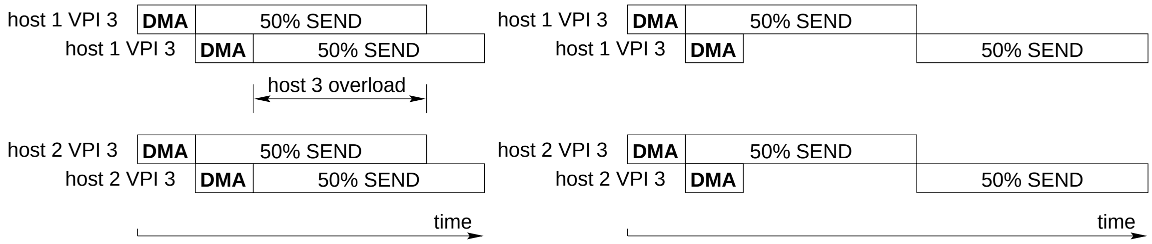

Fig. 4. Left: overloading a receiver ("VPI 3") with multiple transmit buffers on different senders in spite of rate division. Right: the solution implemented in the driver.

another is processed by the Level 3 program. Hence a situation such as the one in Figure 4 (left) may arise where two transmit buffers on the same computer send to different event building buffers on the same receiving computer, in this case with VPI 3: the receiver is overloaded in spite of the rate limitation. One possible solution is to set the rate limit at 25% rather than 50%; however, this would leave some unused bandwidth as well as require more sophisticated control by the Scanner Manager to detect such clashes. It is simpler to restrict senders from transmitting multiple buffers to the same VPI, much as the hardware restricts sending to the same ATM address. Now the second buffer waits for the first to be sent, as shown in Figure 4 (right). The DMA overhead remains hidden. With this restriction in place, the rate limit can be set to v_{max}/N_{send} without consideration of the multiple sending and receiving buffers.

To build an event using these divided sending rates, the Scanner Manager broadcasts one SEND_EVENT message to all the Scanners; each Scanner then sends at its allocated rate, after which it sends its acknowledgement back to the Scanner Manager. Multiple events are built concurrently as in the direct driver tests. This "rate division" algorithm is in contrast with the "barrel shifter" algorithm, which in all forms requires each Scanner to be informed one at a time via the command network when it is to send its data at the full rate. At CDF, this algorithm has been implemented with the Scanner Manager sending the individual SEND_EVENT commands, interleaving events being sent to different Receivers. Thus, the "barrel shifter" method incurs substantial control overhead from generating and passing these messages.

The "rate division" and "barrel shifter" algorithms can be compared by running the event builder system without actually passing any data through the event network. In this case, an "event" is simply a complete round of control messages. These tests therefore measure the best possible (non-empty) event throughputs for the two algorithms. The results for a $4 \rightarrow 4$ system are shown in Figure 5a. The "barrel shifter" plateaus around 450 events/s; the target for Run II is 300 events/s, but 1000 events/s is desired. The "rate division" method, on the other hand, reaches 1000 events/s, albeit without sending any actual data. However, direct measurements show that the CPU is quickly saturated in the "rate division" test; a computer upgrade will likely further increase the event throughput.

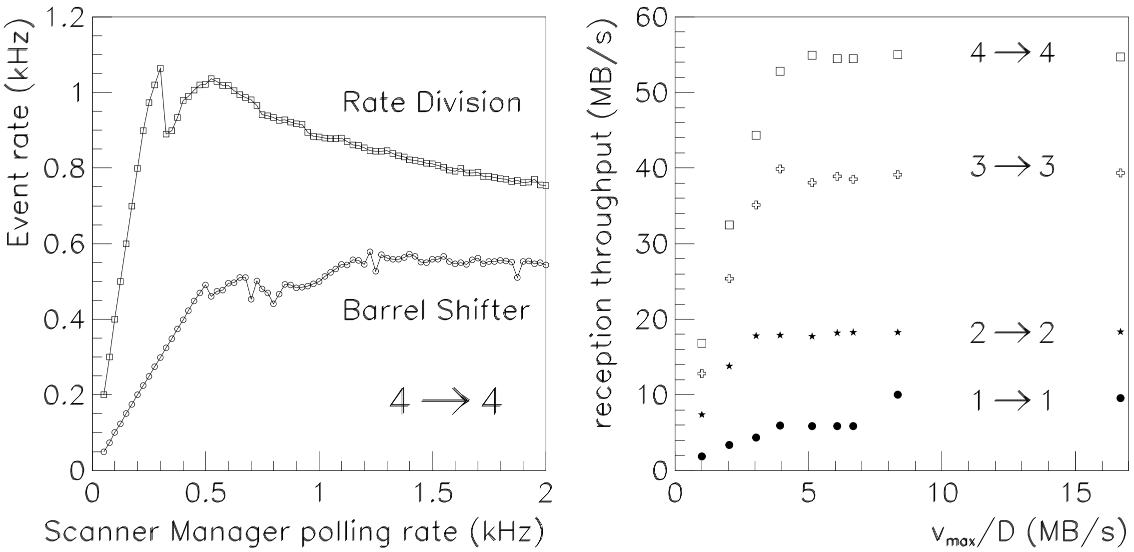

Fig. 5. (a) "Event" (messages only) throughput as a function of the rate at which the Scanner Manager polls for messages, for "barrel shifter" and "rate division" traffic shaping algorithms. The decreasing performance of the "rate division" method above 500 Hz is a result of CPU saturation. (b) A first look at the event builder test system data throughput, with each Scanner sending fixed-size 32000-byte fragments at the set rate limit v_{max}/D.

6. Event Builder System Test

The concern in using rate division for traffic shaping is that cells might be lost. It is clear from the direct driver tests, however, that cell loss can be made very rare or even nonexistent. A first look at sending fixed-size $N_{send} \times 32000$ byte events through the full event builder test system, with all the control software, is shown in Figure 5b. No cell loss was observed. It is encouraging to note that in the $4 \rightarrow 4$ system, each link is carrying more than 13 MB$/s$; the Run II target is 10 MB$/s$. The event throughput is 450 events$/s$ with event fragments twice the size of the average largest fragment size for Run II.

7. Conclusion

This article has reported on results from an event builder test system utilizing an ATM switch and in-house driver and control software. Tests using the ATM driver directly show the expected behavior regarding rate limitations and scaling, all without cell loss. Tests have also begun with the full test system utilizing all the control software. Again, event loss is not observed. Future work will more fully characterize the performance of the system, not only in terms of packet size and rate limits, but also with variable event sizes, in order to better simulate a real event builder system. A software simulation of the switch and event builder system is also planned. Upgrades are already in progress to take advantage of faster computers and optical links as well as to expand to an $8 \rightarrow 8$ test system. The current tests suggest scalability to larger systems such as that projected at CMS, and in the nearer future, with only relatively modest upgrades foreseeable well within the next year, that an ATM-based event builder can meet or exceed CDF Run II performance targets.

8. References

1. F. Abe, *et al.*(CDF Collaboration), *Nucl. Instrum. Meth.* **A271**, 387 (1988); F. Abe, *et al.*(CDF Collaboration), *Phys. Rev.* **D50**, 2966 (1994).
2. The CDF II Collaboration, *The CDF II Detector: Technical Design Report*, FERMILAB-PUB-96/390-E, October, 1996.
3. The CMS Collaboration, *Technical Proposal*, CERN/LHCC 94-38 (LHCC/P1), December 15, 1994.
4. FORE Systems, ForeRunner *ATM Switch Architecture*, April, 1996.