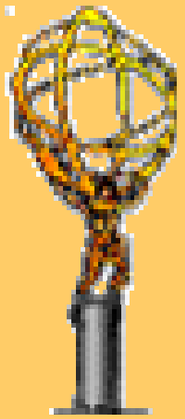


Performance DataCollection

measurement of ATLAS software with QoS

IEEE –NPSS Real Time Conference 2003, 18-23 May 2003

Yoshiji Yasu, Yoji Hasegawa[2], Yasushi Nagasaka[3] Makoto Shimojima[4],
Atsushi Manabe, Masaharu Nomachi[5], Hirofumi Fujii and Yoshiyuki Watase
on behal of the Atlas Trigger/DAQ group
High Energy Accelerator Research Organization(KEK), Tsukuba, Japan (e-mail:Yoshiji.YASU@kek.jp)
[2]Shinshu University, Matsumoto, Japan,
[3]Hiroshima Institute of Technology, Hiroshima, Japan,
[4]Nagasaki Institute of Applied Science, Nagasaki, Japan,
[5]Osaka University, Osaka, Japan

INTRODUCTION

Congestion avoidance of event data flow in ATLAS event builder network is crucial. Traffic management of the data flow is an essential point to avoid the congestion. Therefore, adopting congestion avoidance and flow control techniques for the event builder using switching network technologies are major issues.

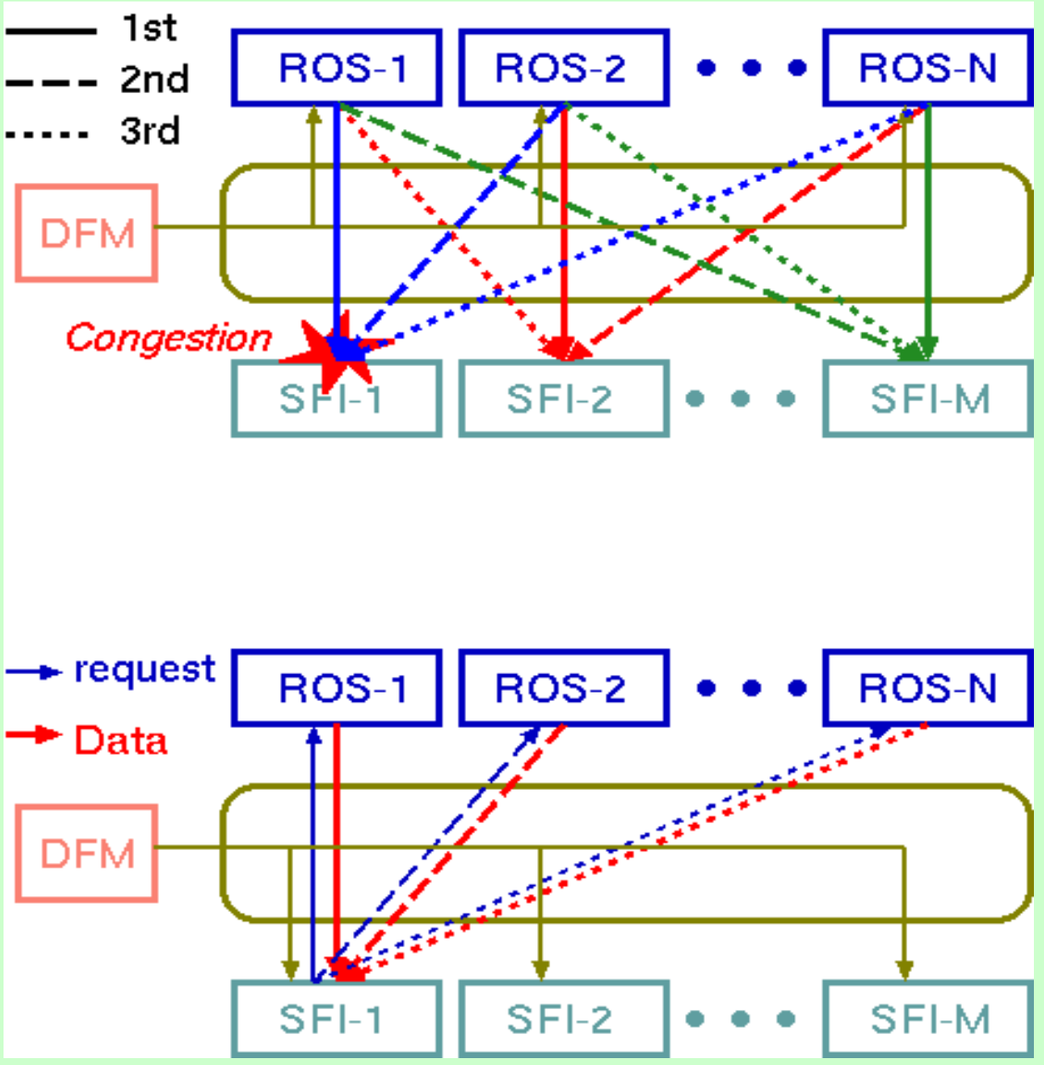
On one hand, Gigabit Ethernet is one of the technologies which enables a high speed transfer for the event builder and a major candidate of ATLAS event builder network. Ethernet provides a best-effort service to all of their applications, with few traffic shaping method in comparison with ATM. IP QoS technique, which is a control scheme at the level of event fragment such as traffic shaping for a packet-oriented network, had been investigated.

On the other hand, there are two different scenarios of data flow for an event builder, which is distinguished by whether the event manager informs the assignment to the sources or the destination. A push scenario is that the sources are responsible for initiating the data transfer, while a pull scenario is that the destination initiates the transfer by requesting the sources to send the data. Both of the push and pull scenarios are implemented into the ATLAS event builder and selectable at execution for investigating the feasibility of the event builder architecture.

ATLAS TDAQ Event Builder

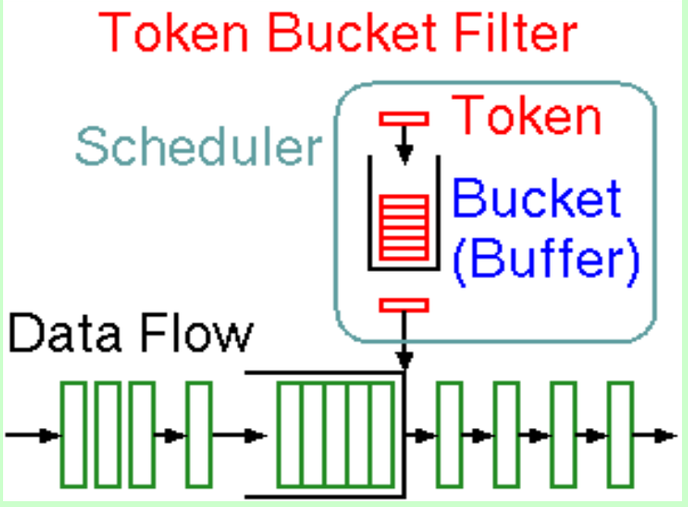
PUSH scenario : DFM assigns an SFI to all ROSs via multicast mechanism. ROSs then respond to the assigned SFI with their respective ROS event fragment. SFI acts as an open receiver and builds the complete event out of the individual fragments received. ROSs will need to control the amount of traffic sent to each SFI individually.

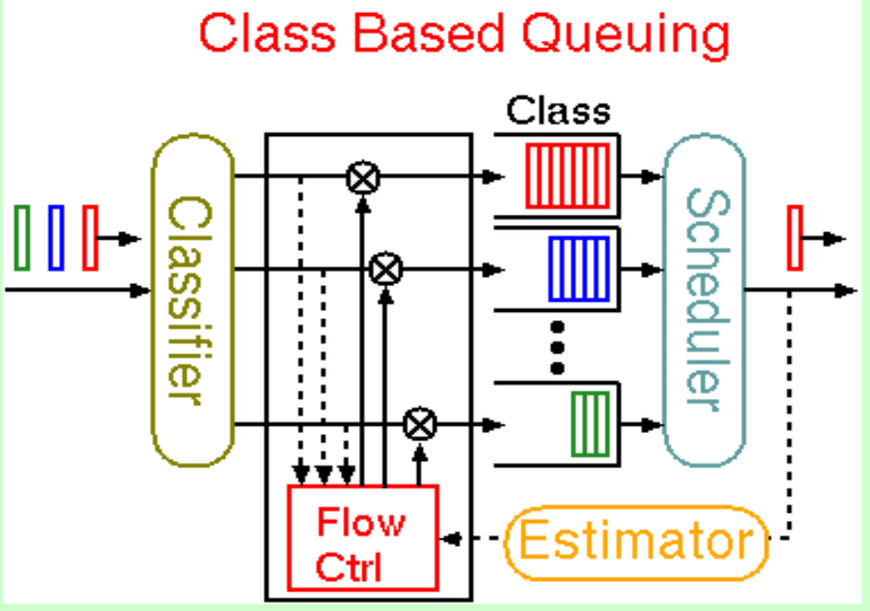
PULL scenario : DFM assigns an event to an SFI. The SFI then requests from each ROS its event fragment via a series of unicast messages. The SFI receives from each ROS individually and builds the complete event. The pull scenario offers the advantages with respect to controlling the flow of traffic although a doubling of the message rate at the level of the SFIs.



Quality of Service (QoS) in Linux Kernel

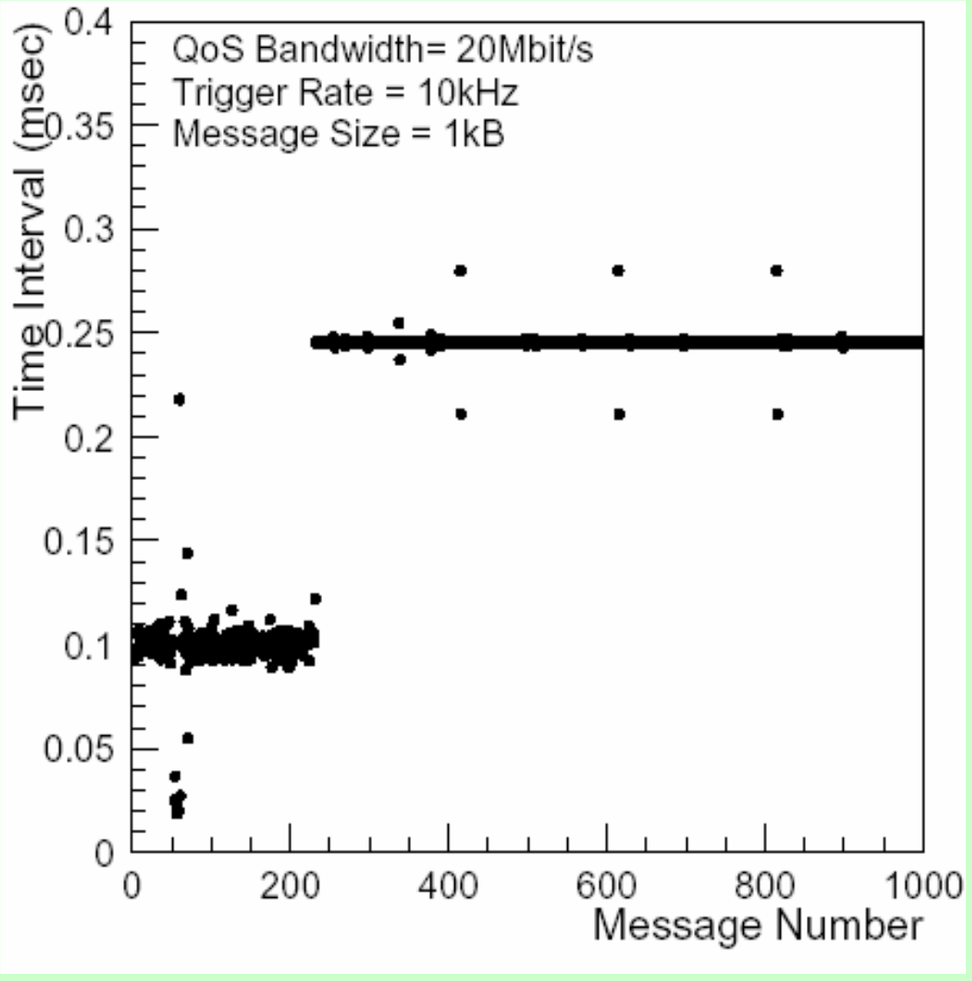
QoS manages the flow of data at the IP level by employing packet classification, packet scheduling and traffic shaping techniques. Packet classification is used to classify incoming packets in groups, such as Class Base Queuing (CBQ). The packet scheduler arranges the scheduling for outgoing packets according to the queuing method and the buffer management selected. Token Bucket Filter (TBF) is an example of one method. The outgoing packet are sent at a rate determined by the size of the token buffer and the rate in which tokens are supplied. The traffic shaping is a technology to make the burst flat. QoS is implemented in the standard Linux kernel at the IP level. QoS can be used to shape the traffic entering a switching network. This removes the necessity of implementing traffic shaping at the level of the application.





Message scheduling in Linux kernel

- Horizontal axis : message number sent from a host to another host.
- Vertical axis : time interval between two messages being sent sequentially.
- HZ parameter (Linux scheduling cycle in Hz) : 4096
- QoS assigned the bandwidth 20Mbit/sec on a gigabit Ethernet.
- Messages whose size is fixed to 1 kB are generated in 10 kHz.
- Once the TBF buffer for outgoing messages is fully filled, messages are sent out in the constant time interval of 0.25 msec., namely, messages are scheduled at 4 kHz.

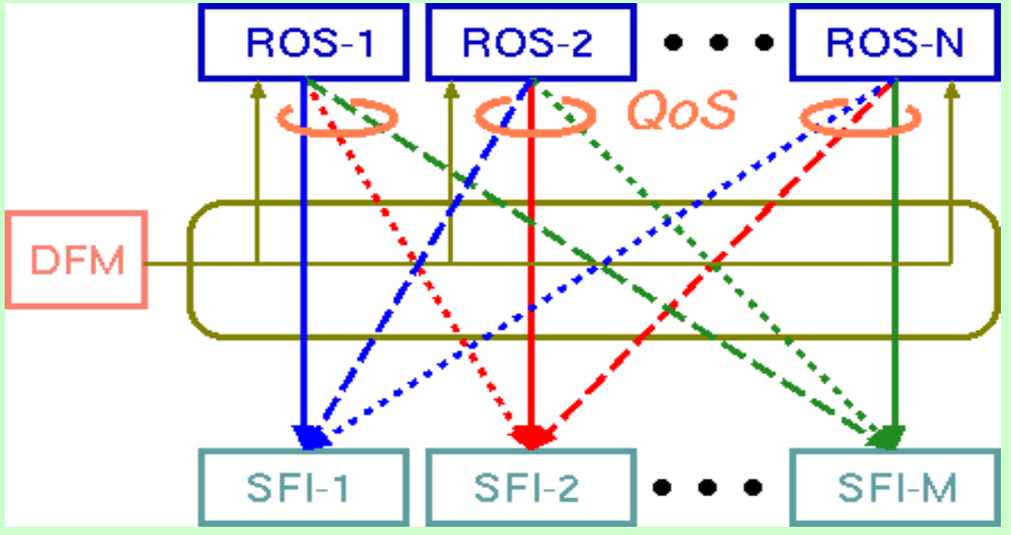


QoS makes the burst flat.

Implementation of QoS to the Event Builder

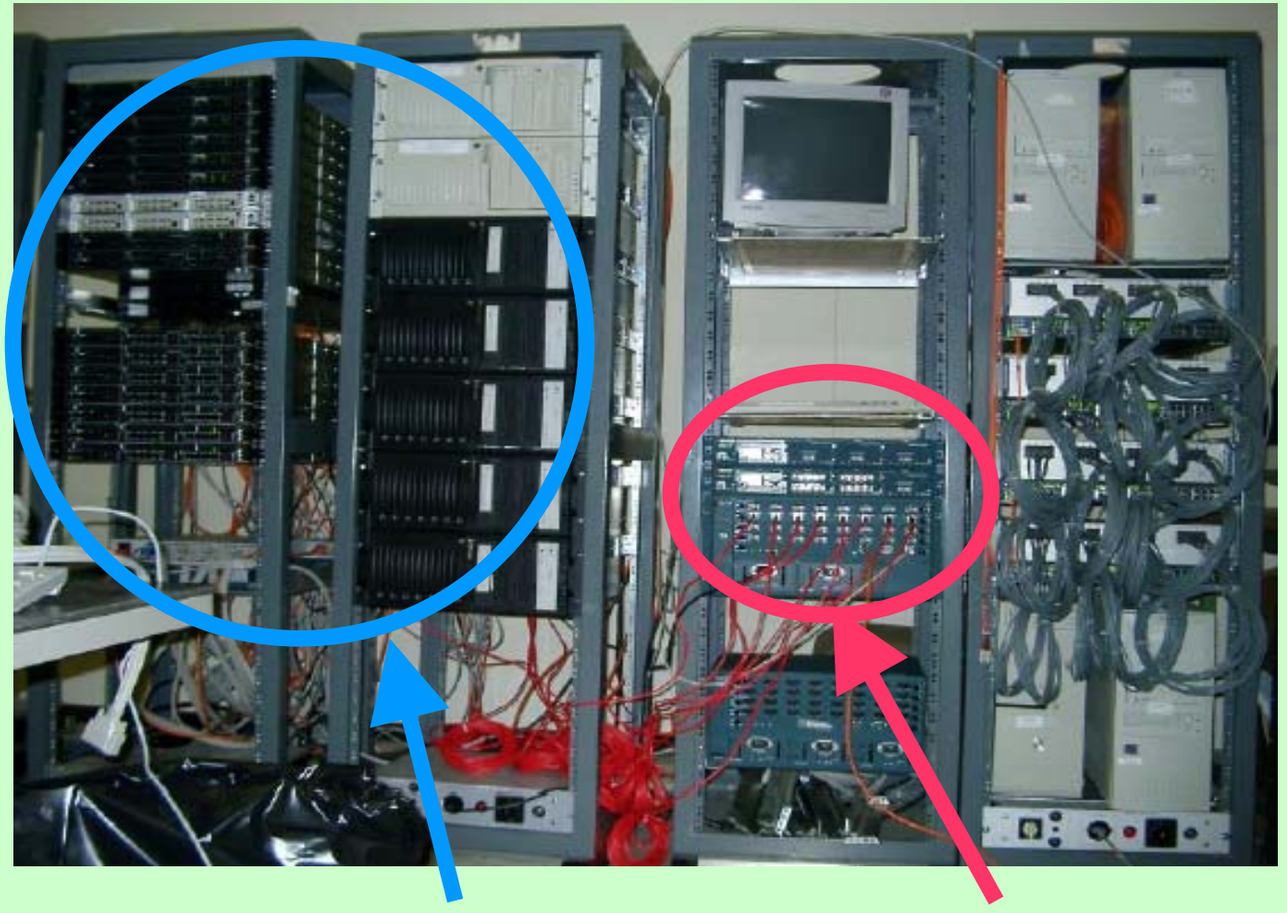
QoS as implemented by the kernel is performed on in the message output queues, i.e at the level of the ROSs, in coming packets continue to be accepted on a best effort basis. It is also important to realize that packets are scheduled at best at the rate of the Linux kernel scheduler, which is a configurable parameter.

Event building is to be performed at a rate of ~3 kHz, therefore the data should be scheduled to at least the same rate for the traffic shaping to be effective. In the studies performed the Linux kernel scheduling frequency was set to ~4 kHz.



SETUP @ CERN

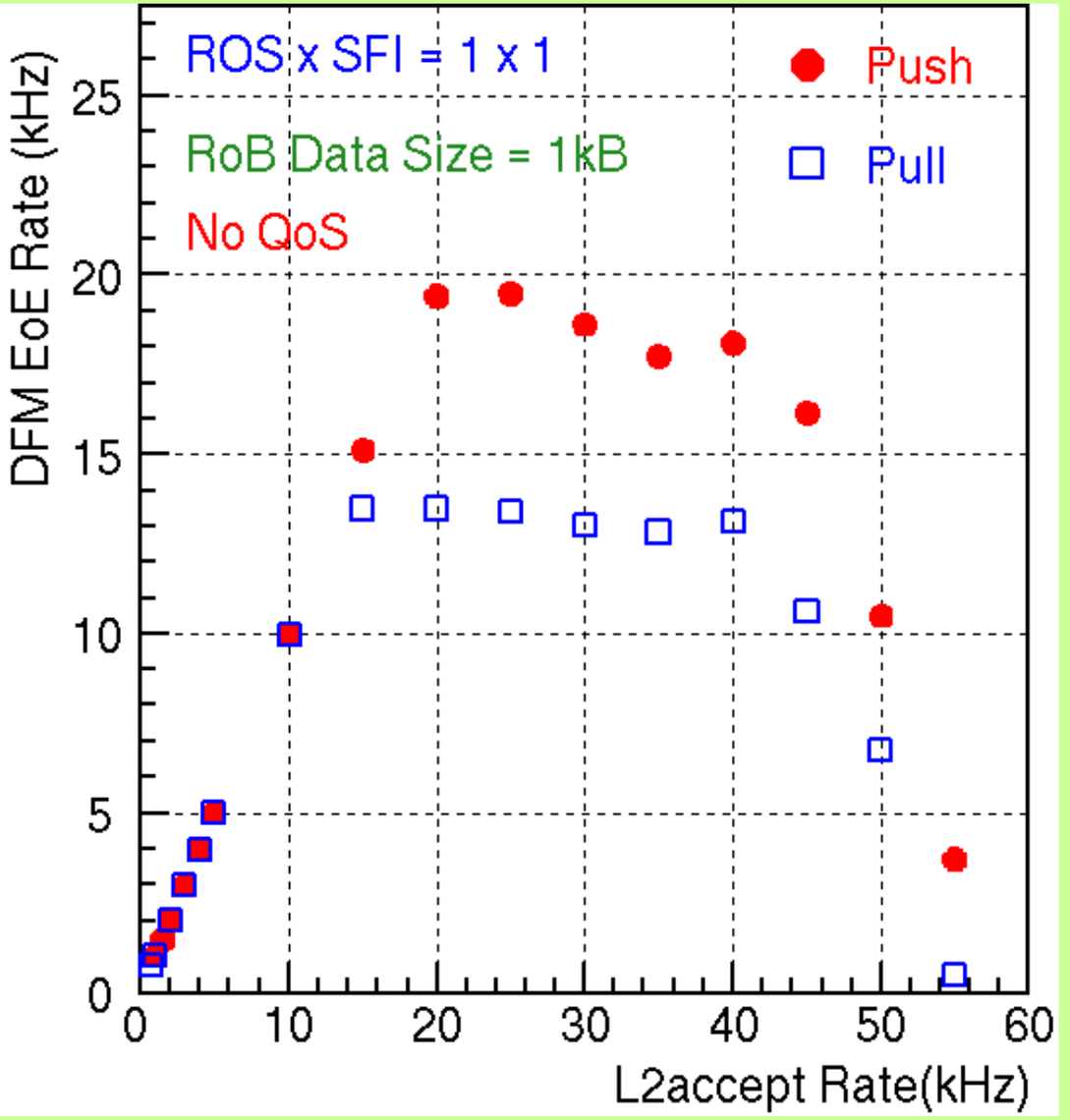
- DFM, ROS, SFI : Dual Xeon(2.2/2.4GHz) PCs with 1GB RAM, GbE NIC (intel Pro 1000, driver e1000)
- GbE Switch : BATM
- OS : RedHat 7.3.1 Linux kernel 2.4.18-27 (QoS included) with :
  - Scheduling time HZ
    = 4096 (Hz)
  - Kernel buffer size
    = 8 MB
- EB software :
  DC-00-02-02
  - UDP/IP is used
- Online software:
  Online-00-17-02



1 ROS x 1 SFI system w/o QoS

- Size (Event Fragment Size from ROS to SFI) : 1kB
- L2accept Rate : Variable
- Maximum Event Builder Rate (EoE Rate) :
  - Push: 20 kHz
    Bottleneck : CPU power at SFI
  - Pull: 14 kHz
    Bottleneck : CPU power at SFI
- Over 40 kHz of Trigger Rate, Event Build fails.

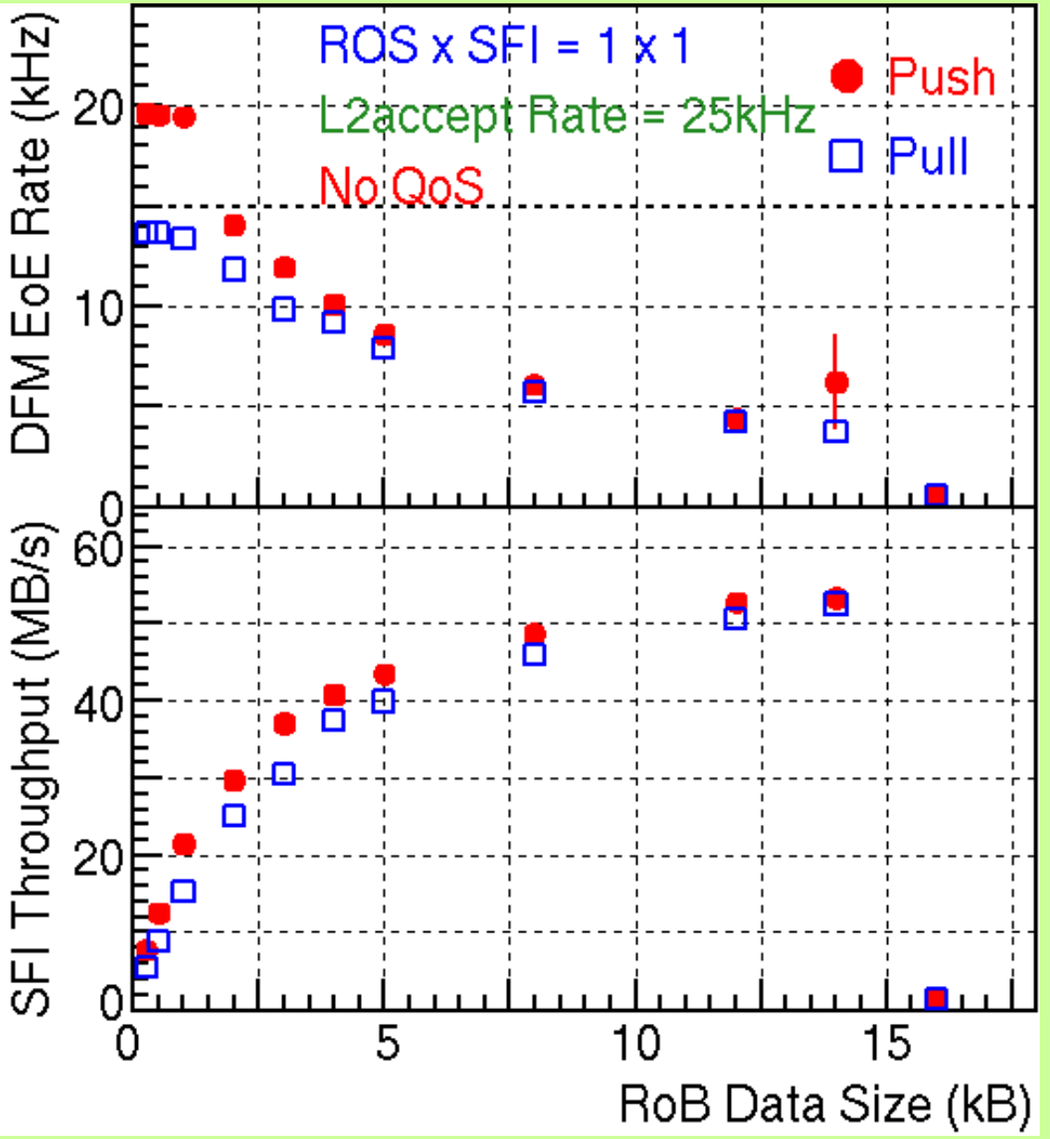
Event Builder Rate vs. L2accept Rate



1 ROS x 1 SFI system w/o QoS (cont.)

Event Build Rate &Throughput vs. RoB Data Size

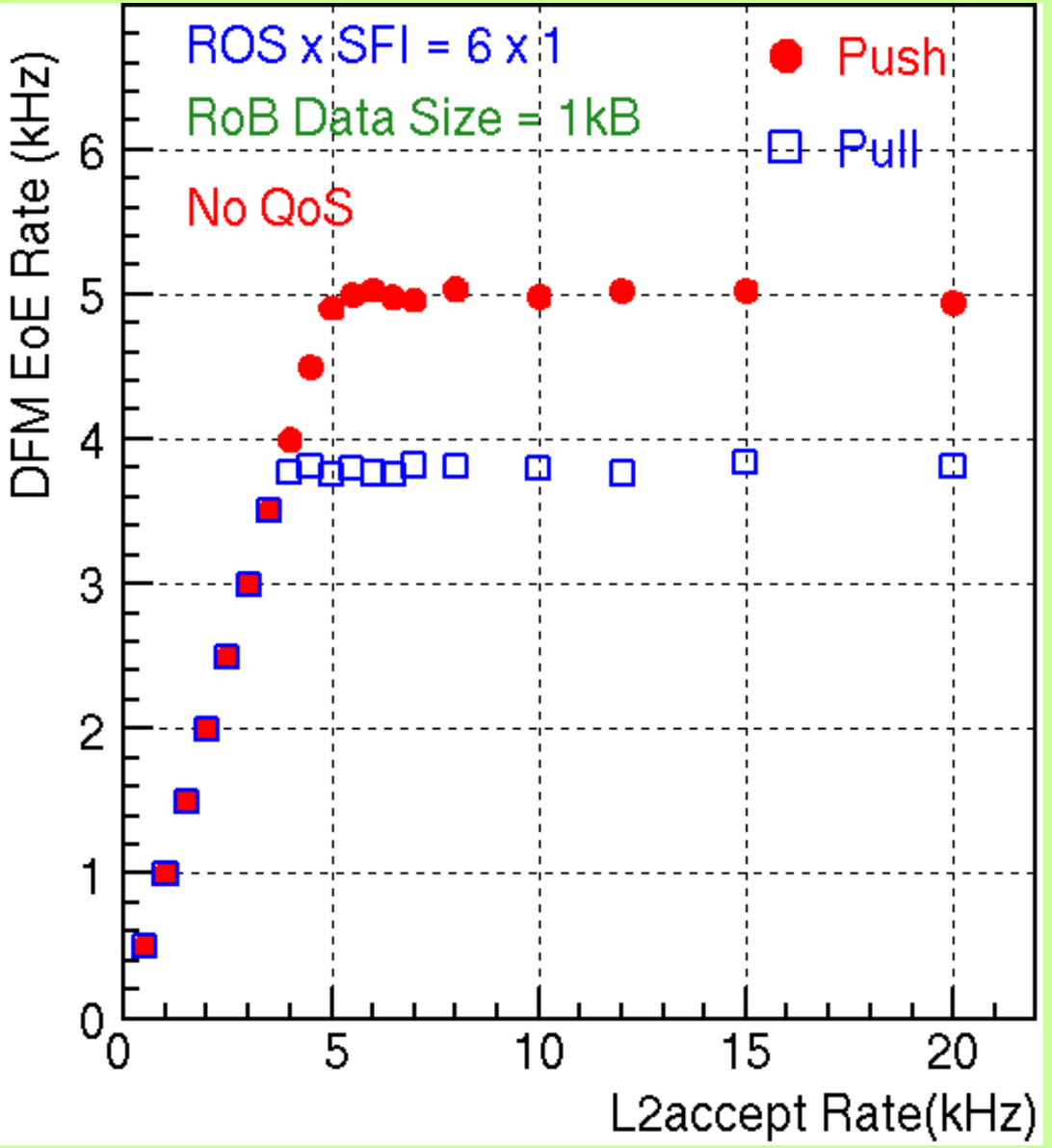
- L2accept Rate : 25 kHz
- RoB Data Size (Event Fragment Size from ROS to SFI) : Variable
- Maximum Throughput :
  - Push, Pull : 52 MB/s @RoB Data Size = 14 kB
- At RoB Data Size is over 15 kB, Event Build fails.



6 ROSs x 1 SFI system w/o QoS

- RoB Data Size : 1kB
- L2accept Rate : Variable
- Maximum Event Builder Rate (EoE Rate) :
  - Push: 5 kHz Bottleneck : CPU power at SFI
  - Pull: 3.8 kHz Bottleneck : CPU power at SFI

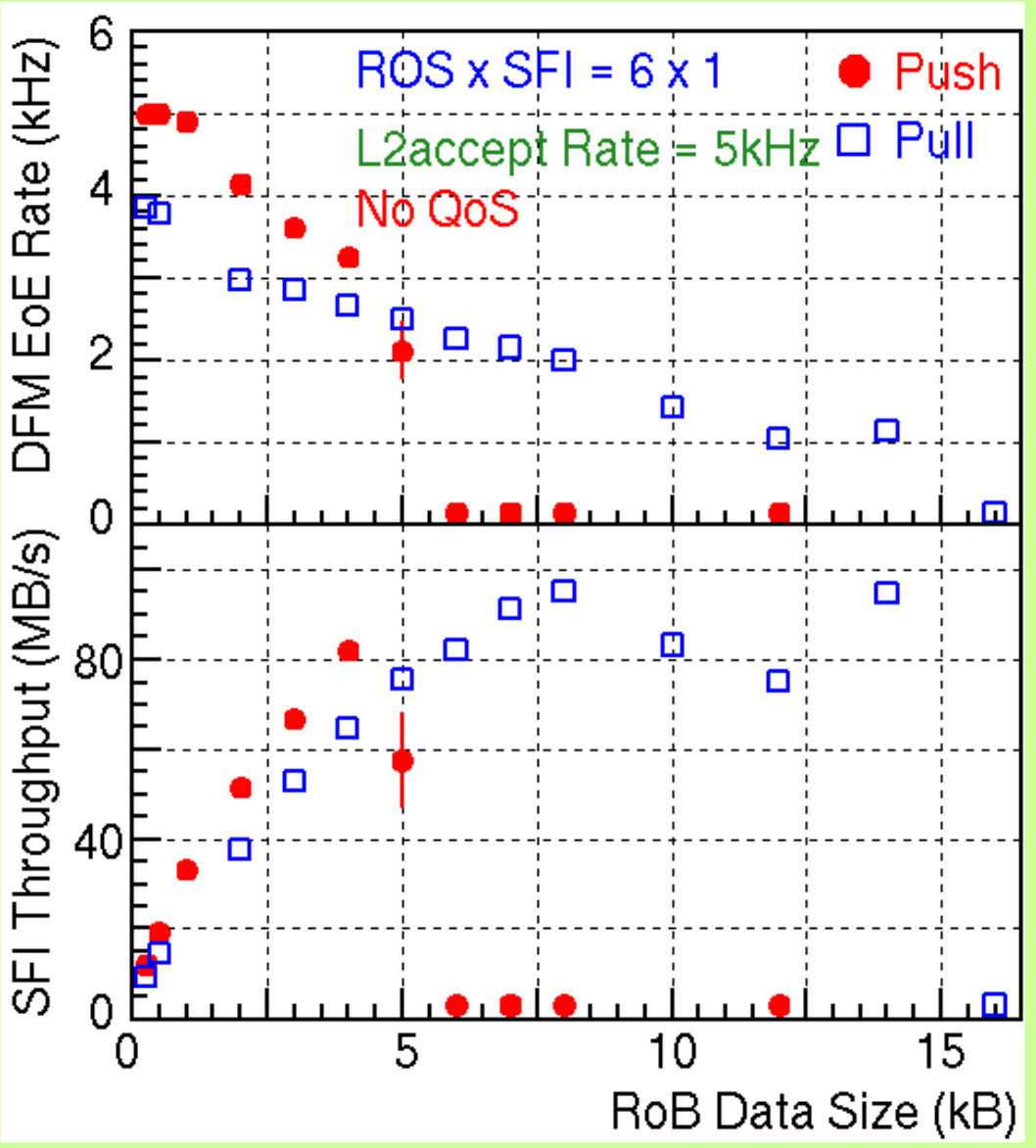
Event Build Rate vs. L2accept Rate



6 ROSs x 1 SFI system w/o QoS (cont.)

Event Build Rate &Throughput vs. RoB Data Size

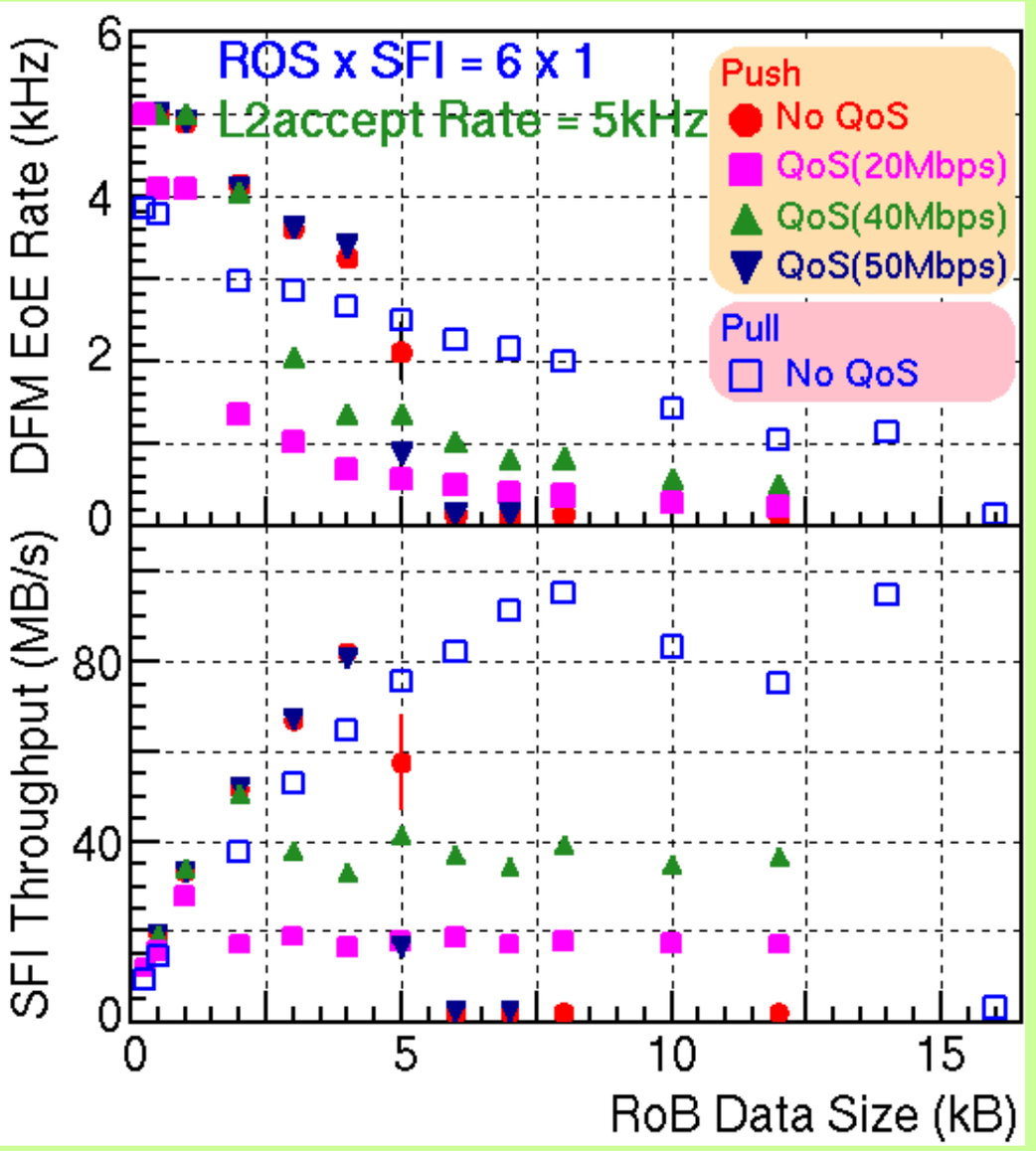
- In push scenario, event building fails when RoB Data Size is over 5 kB - Congestion occurred on SFI.
- Maximum throughput on SFI :
  - Push : 82MB/s - bottleneck : Congestion and CPU power at SFI
  - Pull : 95MB/s - bottleneck : CPU power at SFI



6 ROSs x 1 SFI system w/ QoS

Event Build Rate &Throughput vs. RoB Data Size

- QoS is applied to push scenario
- Assigned rate : 20, 40, 50 Mbps
- When the assigned rate is 20, 40Mbps, there is no congestion at RoB Data Size is larger than 5 kB. Event Build is possible.
- Maximum throughput at SFI :
  - 20Mbps : 18MB/s
  - 40Mbps : ~40MB/s
- However, when the assigned rate is 50Mbps, congestion occurred.
- There is a possibility to avoid the congestion if QoS is applied even if the push scenario is adopted.



CONCLUSIONS

The performance of ATLAS event builder software was measured for case of the push scenario with QoS and the effect of QoS to the software were evaluated. In the push scenario packet loss at the SFI occurs in the condition with larger message size at higher trigger rate and no QoS applied. As a result event building could not be done. However a suitable bandwidth applied to ROSs by QoS makes the event builder working even in the condition which events could not be built without QoS. Now the conclusions are as follows;

1) The pull scenario is better than the push scenario even if the degradation of the performance in the pull scenario was a little bit observed, although large scale ATLAS event builder is so complicated and the results from a small scale test-bed can not be extrapolated easily.

2) IP QoS technique is a good traffic shaping one from viewpoint of the method without modification of the event builder software, but the shaping technique is not always effective on the event builder network. Thus, the QoS technique should be investigated toward the future ATLAS event builder.

REFERENCES

- ATLAS TDAQ: A Network-based Architecture, HP.Beck, R.W.Dobinson, K.Korcyl, M.LeVine, DC-59, February 2000.
- Atlas High-Level Triggers, DAQ and DCS; Technical Proposal., CERN/LHCC/2000-17, March 2000.
- Yoji Hasegawa, Yasushi Nagasaka, Yoshiji Yasu, DAQ/EF-1 Event Builder system and Linux/Gigabit Ethernet, ATL-DAQ-2000-008, March 2000
- Y.Yasu, Y.Nagasaka, A.Manabe, M.Nomachi, H.Fujii, Y.Watase, Y.Igarashi, E.Inoue, H.Kodama, Evaluation of Gigabit Ethernet with Quality of Service for event builder, the 11th IEEE Trans. on Nucl. Sci., Santa Fe, New Mexico, June 14-18, 1999
- Y.Yasu, Y.Nagasaka, Y.Hasegawa, A.Manabe, M.Nomachi, H.Fujii, Y.Watase, Quality of Service on Gigabit Ethernet for Event Builder, the 3rd International Data Acquisition Workshop on Networked Data Acquisition Systems, Lyon, France, October 20, 2000
- Y.Yasu, A.Manabe, Y.Nagasaka, Y.Hasegawa, M.Shimojima, M.Nomachi, H.Fujii and Y.Watase on behalf of the Atlas Trigger/DAQ group, Quality of Service on Linux for the Atlas TDAQ Event Building Network, International Conference on Computing in High Energy and Nuclear Physics CHEP2001, Beijing, September 3-7, 2001
- Message Flow: High-Level Description, HP.Beck, C.Haeberli, DC-12, June 2002
- ATLAS DataCollection home page, http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/DataFlow/DataCollection/DataCollection.html
- ATLAS Online software home page, http://atlas-onlsw.web.cern.ch/Atlas-onlsw/
- S.Floyd and V.Jacobson, Link-sharing and Resource Management Models for Packet Networks, IEEE/ACM Transactions on Networking, Vol.3 No.4, August 1995
- Internet Protocol - Quality of Service, http://qos.ittc.ukans.edu/howto