WAN EMULATION DEVELOPMENT AND TESTING AT FERMILAB.

A.Bobyshev, R.Rechenmacher, P.Demar, FNAL,Batavia,IL 60510, US
M.Ernst, DESY, Hamburg, Germany

Abstract

The Compact Muon Solenoid (CMS) experiment at CERN's Large Hadron Collider (LHC) is scheduled to come on-line in 2007. Fermilab will act as the CMS Tier-1 centre for the US and make experiment data available to more than 400 researchers in the US participating in the CMS experiment. The US CMS Users Facility group, based at Fermilab, has initiated a project to develop a model for optimizing movement of CMS experiment data between CERN and the various tiers of US CMS data centres and to design a WAN emulation facility which will enable controlled testing of unmodified or modified CMS applications and TCP implementations locally under conditions that emulate WAN connectivity. The WAN emulator facility is configurable for latency, jitter, and packet loss. The initial implementation is based on the NISTnet software product. In this paper we will describe the status of this project to date, the results of validation and comparison of performance measurements obtained in emulated and real environment for different applications including multistreams GridFTP. We also will introduce future short term and intermediate term plans, as well as outstanding problems and issues.

SYSTEM ARCHITECTURE

The WAN IP Emulation Laboratory (IPEL) is based on a dedicated switch Catalyst 3750G Cisco Systems., Inc., connected to the core of the Fermlab network. It has multiple connections to the workgroup concentration points to provide capability to move production systems in emulated environment. An IP network of the emulator is split by multiple VLANs that are used to represent networks of workgroup or remote sites. Control and monitoring of the IPEL is supported via a dedicated interface to avoid interference with emulated traffic. After researching of available public domain software we selected the NISTnet [1] package as initial software to build IPEL. The following traffic impairments can be reproduced in our current setup:

- delay or latency to packets
- jitter – a random time variation in the arrival of packets
- drop – a random elimination of one or more packets

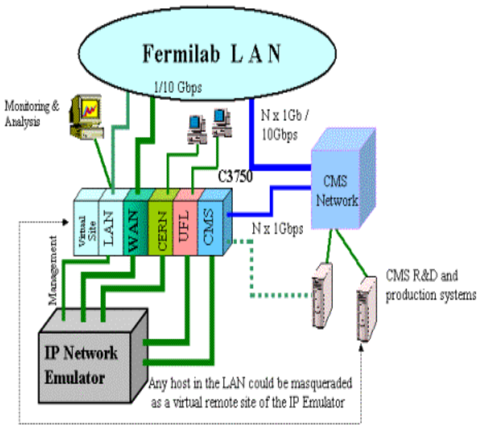

Figure 1: System Architecture of the IPEL.

- traffic shape, or limiting bandwidth
- traffic asymmetry, different network conditions for each direction
- duplicate packets
- jumbo frames

The specification of the Linux boxes that are used to run the NISTNet software and reproduce end user's systems is Pentium IV, 2x2.66Hz CPUs, 2GB RAM, multiple SysKonnect SK9821 V2.0 Gigabit interfaces. All emulated connections are supported for up to 1Gbps rates.

TESTS AND RESULTS

In the first tests we verified how accurately the NISTNet software reproduces basic traffic conditions such as delay, jitter, bandwidth shaping, packet loss and duplications. We used several different applications PING, IPERF[2]. All systems involved in testing were the Linux boxes with the standard Linux kernel 2.4.22 as well as in-house modified version of the TCP/IP stack that allows an increase in achievable throughput. During testing the IP Emulator was configured by an automatic script for certain traffic conditions symmetrically in both directions. Measurements of RTT, jitter, drops and duplications were implemented by using the summary output from the regular ping program that was sending a few hundred packets in each iteration. Throughput was measured by iperf tool. The graphs in figure 2 give an example of emulation with 60ms (one way) delays that is typical for the path between Fermilab and CERN.

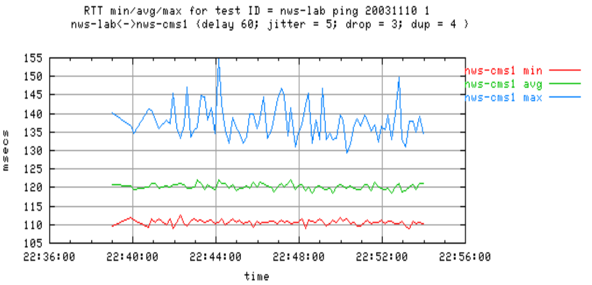

Figure 2: Example of emulated delay and jitter

The graphs in figure 3 reflect to the traffic conditions with 3% drops and 4% duplications.

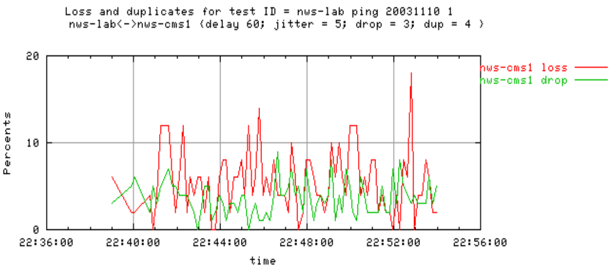

Figure3: Example of emulated drops and duplications

The results of traffic shaping were also very close to what was configured and expected.

Our general conclusion is that NISTnet software does reproduce the basic impairments with sufficient accuracy.

In the next step of validation we ran a few tests concurrently in the real world and in an emulated environment under similar traffic conditions. The traffic characteristics for the real paths were taken from the results of IEPM-BW[4] project as well as from short-term and more precise measurements that we conducted before our tests. However, it should be mentioned that for accurate comparison it is necessary to have identical systems. While doing tests in the real world we had to use the systems that were available to us. Typically, these computers are the working nodes at the remote sites loaded with jobs. In addition, some tests might produce a high volume traffic. That is why we could not run it for a long time to avoid interference with production traffic. The graphs in figure 4 show the results of measurements for the path between Fermilab and the University of Toronto, both in the real world and emulation. As we can see RTT and throughput in emulation and the real world are close. This was not always the case for other sites.

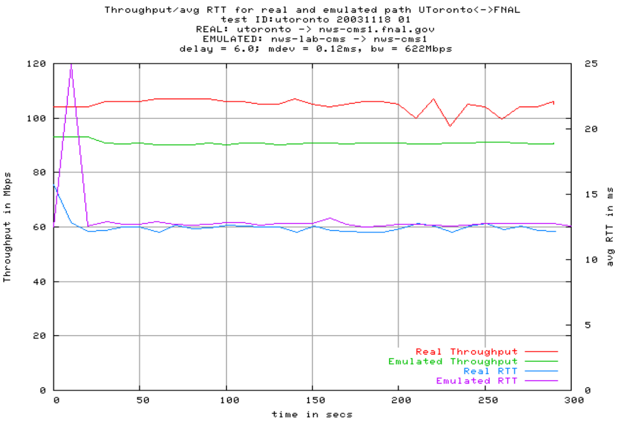

Figure 4: Comparison of an emulation and the real path measurements.

After measuring the network performance characteristics we also researched the behaviour of applications used for data transferring. Our focus was on GridFTP[3] under conditions similar to the path between Fermilab and CERN. The measurements for 1 stream transferring were very close in the emulation and in the real world. With increasing a number of parallel streams difference in performance was also increasing.

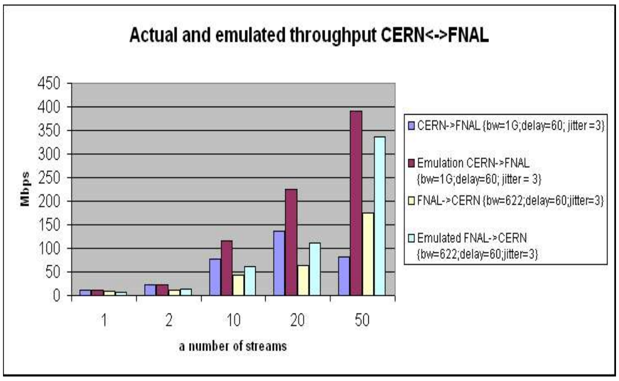

Figure 6: GridFTP in emulation and in the real path.

Testing of the modified Linux kernel

As mentioned above one goal of IPEL is developing and evaluating of modifications to TCP stack in linux to improve performance characteristics in a large data transfer. The main phenomena which causes poor bandwidth utilization in the large bandwidth delay product environment is the standard congestion avoidance algorithm. This is not to say that the algorithm is defective, just that the problem is non-trivial.

The current setup of the IPEL was used to implement modifications to the standard Reno TCP/IP stack, evaluate it for traffic conditions adequate to the path between Fermilab and CERN, and then compare with the performance in the actual path.

Detailed discussion of the algorithm is out of scope of this brief article. The brief explanation follows. The goal of our algorithm would be to have a recovery from packets loss or retransmissions occur on the order of seconds in any RTT environment. For every packet received, reduce or increase the cwnd by the RTT as opposed to cwnd ±= 1 / cwnd (Ref. tcp_input.c). A further algorithm will attempt to determine how many bytes each ACK is ACKing. Figure 6 presents the results of throughput measurements for a 0.0015% packet loss in unmodified and modified Linux kernel evaluated in the emulator. In average we saw a 25-35% throughput increase for the modified kernel. After evaluation in the emulator we deployed these modifications on the production systems at Fermilab. It brought about a 20% increase in achievable throughput to CERN. We ceratinly realize that this not yet a final solution, there are many issues that need to be addressed. Our goal in this stage is to build an emulation tool that will help to investigate behaviour of TCP stacks and applications to improve performance characteristics.

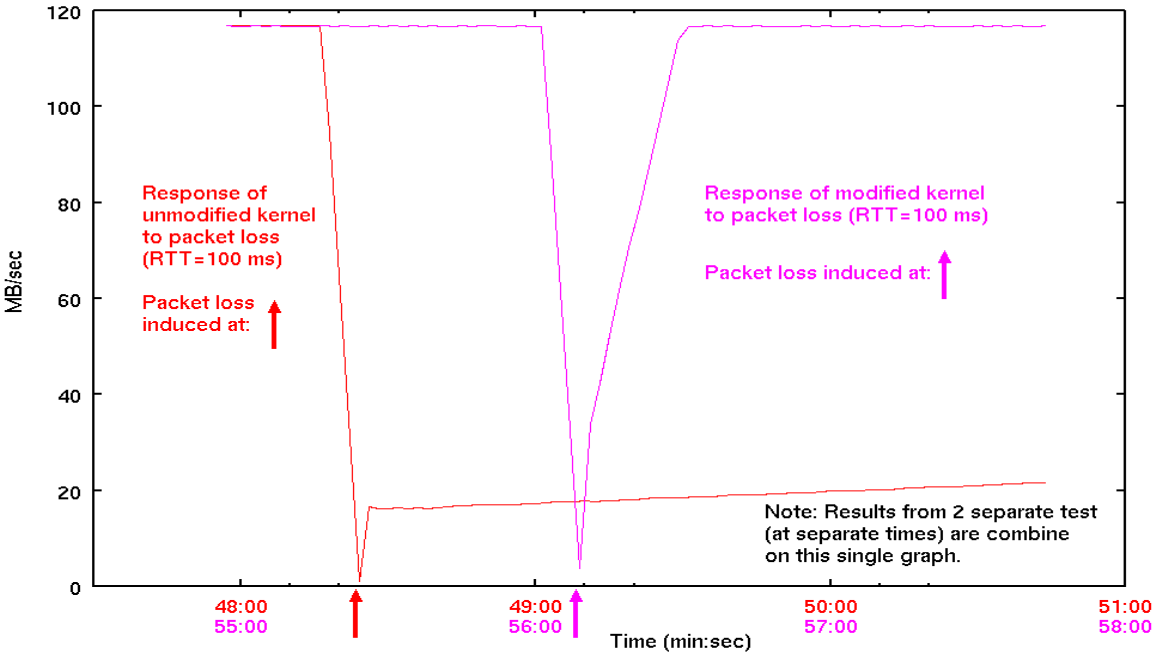

Figure 6: Throughput in modified and unmodified Linux kernel achived in the emulator.

PLANS AND WORK IN PROGRESS

A generator of background traffic is required to make results of emulation more close to the real world. There are a number of open source traffic generators available, however we are looking at having such a noise generator as an integrated feature of the IPEL. The preliminary tests with the packet generator version 1.8 included in the recent Linux distributions were promising.

The network at Fermilab is migrating toward a 10Gb rates in the core and on external links to multiple remote sites through interconnections at the StarLight[5]. It dictates the necessity to emulate bandwidth above 1Gbps and eventually up to 10Gbps.

Our initial experiments with 10Gbps host connections and emulation of bandwidth above 1Gbps show that progress in this direction can be made. However, this work is still in progress and more problems could be discovered later.

CONCLUSION

We found our experiments in an IP emulation as very useful. Although the results of measurements in emulation and in the real world are not always adequate an emulation provides reproducible traffic conditions that are especially important while working with applications that require transferring of a huge amount of data. It is not achievable in the production networks without a risk to disrupt production service. The final goal of this project is to make emulation service available for the end users in order to help them in evaluation and debugging of actual applications. Many user communities at Fermilab, such as CMS, CDF, D0 have expressed their interest in such kind of services.

REFERENCES

[1] The NISTNet http://www-x.antd.nist.gov/nistnet/.
[2] IPERF Version 1.7.0 http://dast.nlanr.net/Projects/Iperf/
[3] The GridFTP protocol and software http://www.globus.org/datagrid/gridftp.html
[4] IEPM-BW http://www-iepm.slac.stanford.edu/bw/
[5] StarLight Project http://www.startap.net/starlight/