

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.

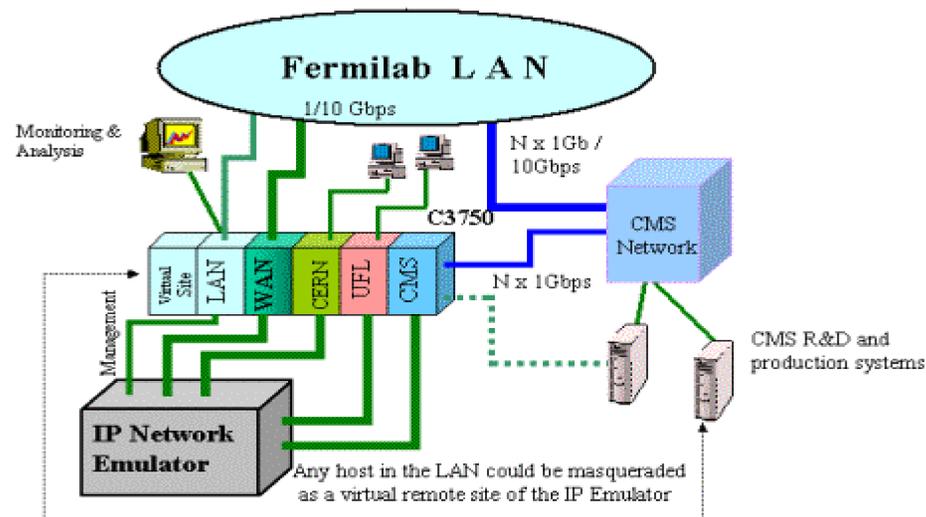
SYSTEM ARCHITECTURE

The WAN IP Emulation Laboratory (IPEL) is based on a dedicated switch Catalyst 3750G Cisco Systems., Inc., connected to the core of the Fermilab 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 the workgroups 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
- 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:

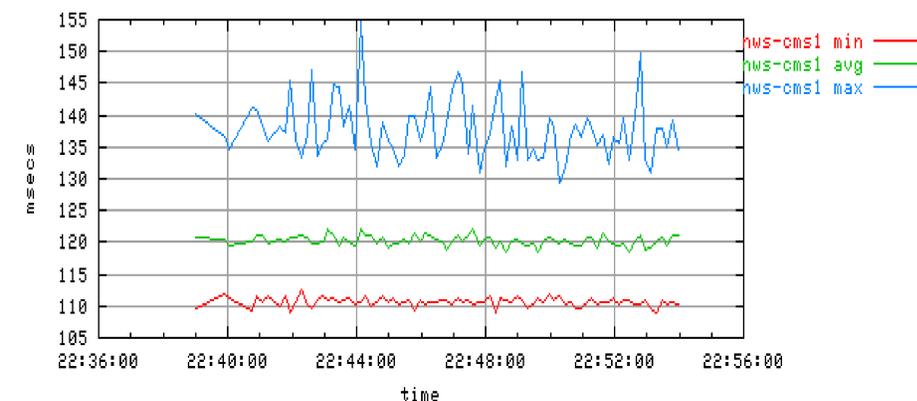
- Pentium IV, 2x2.66Hz CPUs
- 2GB RAM
- multiple SysKonnect SK9821 V2.0 Gigabit interfaces.



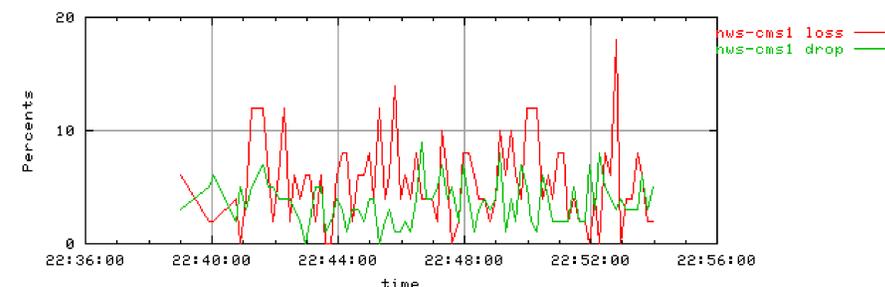
TESTS AND RESULTS

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 a few different applications PING, IPERF. 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. Throughput was measured by the iperf tool. The graphs below give an example of emulation with 60ms (one way) delays typical for the path between Fermilab and CERN. Another graph illustrates emulation of traffic with 3% drops and 4% duplications.

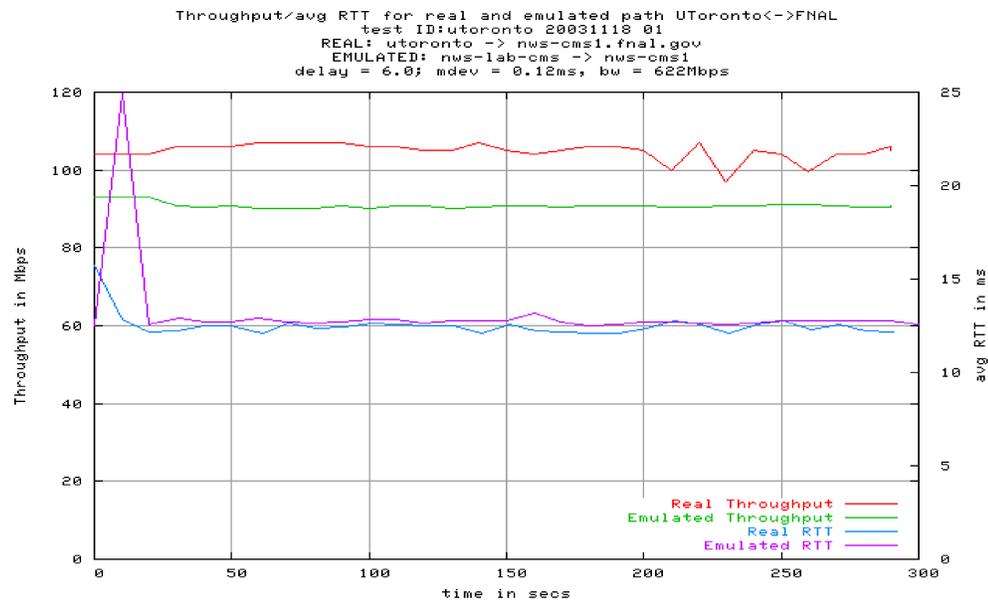
```
RTT min/avg/max for test ID = nws-lab ping 20031110 1
nws-lab<->nws-cms1 (delay 60; jitter = 5; drop = 3; dup = 4 )
```



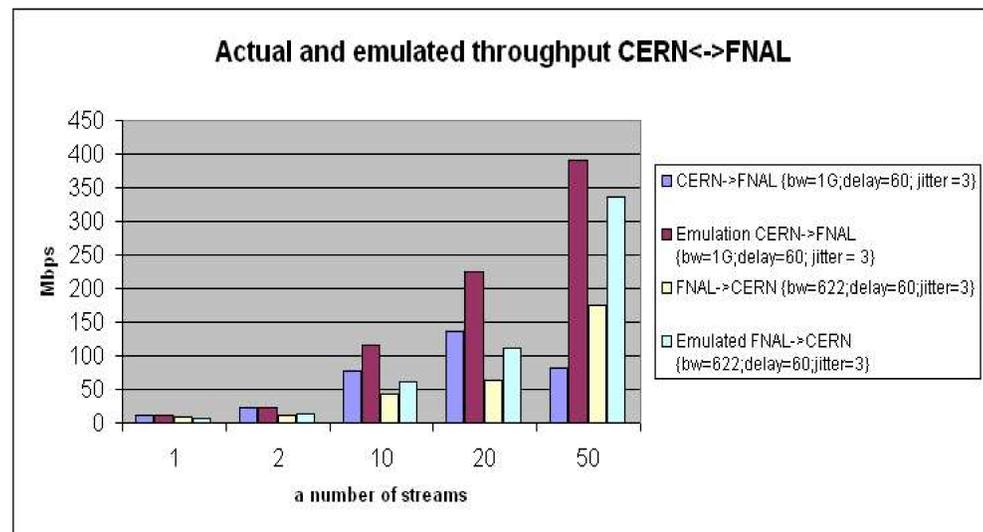
```
Loss and duplicates for test ID = nws-lab ping 20031110 1
nws-lab<->nws-cms1 (delay 60; jitter = 5; drop = 3; dup = 4 )
```



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 the IEPM-BW project as well as from the short-term but 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. Also some tests might produce a high volume of traffic. That is why we could not run it for a long time to avoid interference with production traffic. The graphs below show the results of measurements for the path between Fermilab and the University of Toronto, both in the real world and emulation.



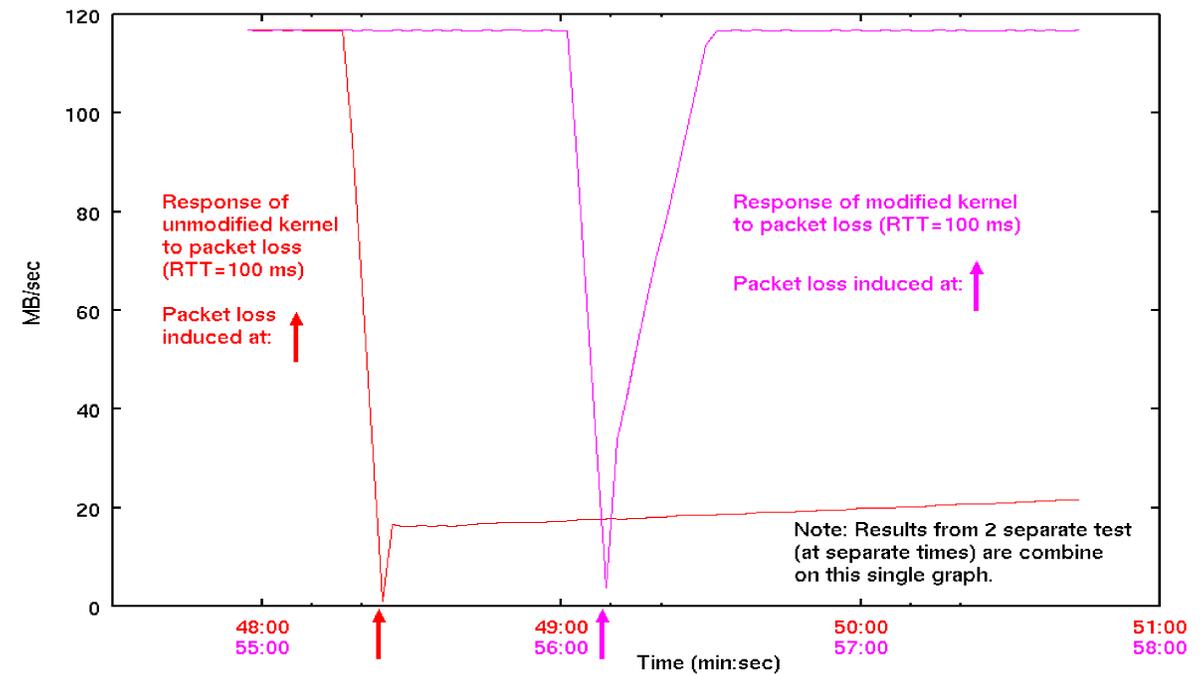
After measuring the network performance characteristics we also researched the behaviour of applications used for data transferring. Our focus was on GridFTP under conditions similar to the path between Fermilab and CERN. The measurements for a 1 stream transfer sessions were very close in the emulation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



TESTING OF THE MODIFIED LINUX KERNEL

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.

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 $cwnd \pm 1 / cwnd$ (Ref. tcp_input.c). A further algorithm will attempt to determine how many bytes each ACK is ACKing. Figure below presents the results 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 in the production systems at Fermilab. It brought about a 20% increase in achievable throughput to CERN. We certainly 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.



PLANS AND WORK IN PROGRESS

A generator of a 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 it as an integrated feature of the IPEL. The first tests with the packet generator version 1.8 included in the recent Linux distributions were promising.

The network at Fermilab is migrating toward a 10Gbps bandwidth range provided in the core and on external links to the multiple remote sites through interconnections at the StarLight. It dictates the necessity to emulate a bandwidth above 1Gbps, 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 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 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 a 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.