ULTRALIGHT: A MANAGED NETWORK INFRASTRUCTURE FOR HEP

Shawn McKee, University of Michigan, Ann Arbor, MI, 48109, USA
Harvey Newman, Frank Van Lingen, California Institute of Technology, Pasadena, CA, 91125, USA
Laird Kramer, Florida International University, Miami, FL, 33199, USA
Dimitri Bourilkov, Richard Cavanaugh, University of Florida, Gainesville, FL, 32611, USA

Abstract

We describe the networking details of NSF-funded UltraLight project and report on its status. The project's goal is to meet the data-intensive computing challenges of the next generation of particle physics experiments with a comprehensive, network-focused agenda. The UltraLight network is a hybrid packet- and circuit-switched network infrastructure employing both "ultrascale" protocols and the dynamic creation of optical paths for efficient fair sharing on long range networks in the 10 Gbps range. Instead of treating the network traditionally, as a static, unchanging and unmanaged set of inter-computer links, we instead are enabling it as a dynamic, configurable, and closely monitored resource, managed end-to-end, to construct a next-generation global system able to meet the data processing, distribution, access and analysis needs of the high-energy physics (HEP) community.

THE ULTRALIGHT NETWORK

A primary goal of the UltraLight Project [1] is to augment existing grid computing infrastructures, currently focused on CPU and storage, to include the network as an integral Grid component that offers reliable, and if possible guaranteed, levels of service. Developing and prototyping services to support this vision have been our focus, as we deployed and evolved the UltraLight network throughout 2005. The UltraLight network is shown in Figure 1.

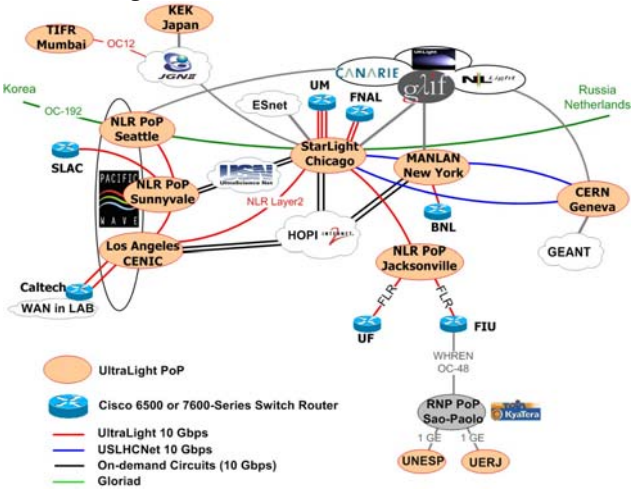

Figure 1 The UltraLight network

UltraLight relies upon NLR, Abilene, ESnet, HOPI, UltraScienceNet (USNet), US-LHCNet and various regional networks (CENIC, FLR, MiLR) to create our UltraLight backbone network.

Basic Network Services

UltraLight intends to provide on demand bi-directional data paths between UltraLight nodes. These paths will be either *dedicated* Layer 2 (L2) channels (with guaranteed bandwidth, delay, etc.) or L2 paths shared with other traffic. In both cases, the only constraint should be the Ethernet framing and the end-to-end connections will appear to be point-to-point. UltraLight will attempt to be as transparent as possible to end-users. In particular, users should be able to run the protocols of their choice over Ethernet[*].

The use of *dedicated* L2 channels is an expensive solution, and often leads to poor utilization of network resources. L2 channels sharing the bandwidth available may be a more cost-effective solution and will have to be used where dedicated L2 channels cannot be provisioned. QoS mechanisms are being studied and deployed to improve the level of services (see the QoS/MPLS section following). The technology used will be based on tagged VLANs and/or MPLS but it should be transparent to end users. Some initial progress has been made (see MPLS/QoS Services and Planning) in conjunction with ESnet and the OSCARS and TeraPaths projects on developing QoS/MPLS capabilities which may eventually provide bandwidth management for the UltraLight infrastructure.

UltraLight will dedicate a few L2 channels to connect each site and offers IPv4/IPv6 services. UltraLight has its own address space and autonomous system. We currently have the following network address spaces and services:

- DNS domain ultralight.org; DNS at **192.84.86.88**
- Autonomous System number **32361**
- IPv4 addresses
  - 192.84.86.0/24
  - 198.32.43.0/24
  - 198.32.44.0/24
- IPv6 addresses 2001:468:0e9c::/48
- A network operations center (NOC) for problems is reachable via email at *noc@ultralight.org*

[*] Dedicated L2 channels are provisioned by interconnecting waves or channels of time-division multiplexing (TDM) systems. A dedicated L2 channel is functionally equivalent to a circuit switched path. The network resource is reserved end-to-end and cannot be used by other traffic if under-utilized. The capacity of the channel cannot be temporarily extended; packets will be dropped if the traffic exceeds the channel bandwidth.

This allows us to interconnect the UltraLight testbed to conventional IP networks and facilitate access to the testbed from sites not connected to UltraLight. UltraLight peers with other backbones at Chicago, Los Angeles, Seattle and in New York.
Rancid[†] systems have been setup at Michigan and Caltech to track equipment configurations. An example is at http://linat08.grid.umich.edu/cgi-bin/cvsweb.cgi showing the type of configuration information tracked.

Data transport protocols

The protocols used to control the information flow across the network are one of the important areas UltraLight plans to explore. The most widely used protocol, especially for reliable data transport, is TCP. TCP, its variants, limitations and extensions will be examined by UltraLight in conjunction with the FAST team [2].

TCP and its variants:

TCP is the most common solution for reliable data transfer over IP networks. Since TCP was introduced in 1981, networks topology and capacity have evolved dramatically. Although TCP has proven its remarkable capabilities to adapt to vastly different networks, recent theories have shown that TCP becomes inefficient when the bandwidth and the latency increase. TCP's additive increase policy (AIMD: Additive Increase, Multiplicative Decrease) for moderating the window size, based on the often-incorrect presumption that packet losses indicate network congestion, limits its ability to use the available bandwidth efficiently.
The Ultralight testbed is the ideal place to evaluate and test new TCP stacks at 10 Gbps speed. Efficiency, the requirements and effect on end-hosts, the ability to coexist stably with other TCP implementations and the ability to share the bandwidth fairly will be evaluated. HSTCP, TCP Westwood+, HTCP, and FAST TCP are some of the new implementations we have tested. So far FAST has proven to the most promising, and adaptable to a variety of working environments.
FAST TCP is an implementation of TCP with a new congestion control algorithm that is optimized for high speed long distance transfers. While the congestion control algorithm in the current TCP implementation uses packet loss as a measure of congestion, FAST TCP uses round-trip delay (time from sending a packet to receiving its acknowledgment). This allows FAST TCP to stabilize at a steady throughput without having to perpetually push the queue to overflow as loss-based schemes inevitably do. Moreover, delay has the right scaling with link capacity that enhances stability as networks scale up in capacity and size [3].
In November 2005 at SC2005, the UltraLight team sustained average data rates above the 100 Gbps level for several hours for the first time. The extraordinary data transport rates were made possible in part through the use of the FAST TCP protocol, and a new FAST release.

Other data transport protocols:

Another approach to overcome TCP's limitations is to use UDP-based data transport protocols. The best known protocol is UDT proposed by B. Grossman. Collaboration with the SABUL/UDT team is under discussion. Some servers dedicated to UDT tests have already been installed at CERN. Other servers may be installed at Los-Angeles and directly attached to the UltraLight backbone.

Network Monitoring

Network monitoring is essential for the UltraLight project. We need to understand our network infrastructure and its performance both historically and in real-time to enable the network as a managed robust component of our infrastructure. There are two ongoing efforts we are utilizing to help provide us with the monitoring information required: IEPM and MonALISA.

IEPM

As part of the UltraLight project we are installing the Internet End-to-end Performance Monitoring (IEPM see http://www-iepm.slac.stanford.edu/bw/) toolkit at major UltraLight sites. This provides a realistic expectation for network performance on the production networks between UltraLight sites, plus a powerful trouble shooting and planning tool.
Active network measurement probes can be sent at regular intervals from the toolkit monitor site to a list of hosts at remote (monitored) sites at regular intervals and the resulting data logged in local archive files.
The probes are deliberately lightweight with minimal network impact (20 probes of 1500 bytes each per host pair per direction, per measurement - a measurement is made at roughly three minute intervals) so currently no special distributed scheduling is needed.

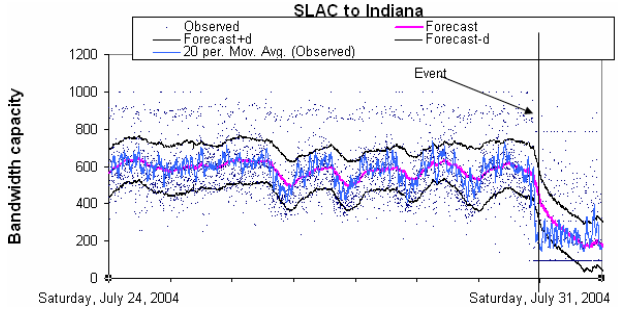

Figure 2 Example IEPM fault detection graph

The probes provide measurements of capacity, cross-traffic and available bandwidth, together with Round Trip Time (RTT) and traceroutes (at 10 minute intervals).
The data is analyzed and graphical reports (time series, histograms, tables) produced at the end of each measurement cycle (see Figure 2).
A web site is created for each monitor site with a top level page [4] providing a table of all the hosts monitored with

[†] A widely used network router and device monitoring system, see for example http://www.shrubbery.net/rancid/

drill down to the time series, histograms, tables, traceroute analysis and topology visualization, raw data, host and probe configuration database querying and updating, and not to forget documentation.

MonALISA

The MonALISA framework will allow us to collect a complete set of network measurements and to correlate these measurements from different sites to present a global picture. Currently the system allows gathering monitoring information from:

- SNMP agents to describe traffic
- PIPES system to measure available bandwidth, one way delay
- ABping a bandwidth/RTT measurement tool
- IEPM measurements via a Web Service interface
- ABILENE traffic via a Web Service interface
- NetFlow to analyze flows
- Ganglia using the multicast protocol or gmetad

We developed a real time network topology monitoring agent in the MonALISA system. It provides complete picture of the connectivity graphs and delay on each segment for routers, networks and AS and is shown in Figure 3.

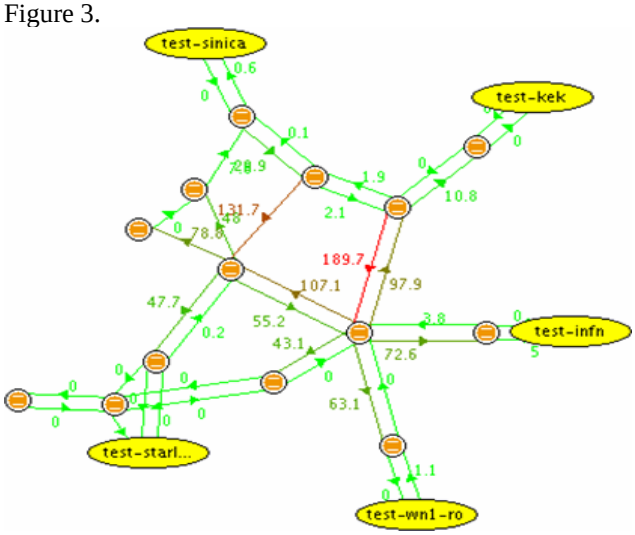

Figure 3 Real-time topology monitoring in MonALISA

This information is used to spot asymmetric routing, segments with problems and it also allows identifying a possibly better alternative path between any end points,

We developed a set of dedicated modules to monitor and control optical switches. These modules are using the TL1 protocol to interact with optical switches. Two versions were developed: for GlimmerGlass and Calient systems. The monitoring modules allow monitoring the optical power per link (for the Glimmerglass switch), the connectivity map and the status of each connections. The MonALISA control agents provide the functionality to modify the cross connections map and they interact with global agents used to generate on demand an end to end optical path or tree.

End-node monitoring

We have developed and additional monitoring tool to gather information about end-host systems since many "network" problems are really problems with underpowered or misconfigured hosts. A simple Perl script was developed to gather host related details in three areas: system information, TCP configuration and network device information. The ApMon API was used to publish the found information into a MonALISA repository. This utility was tested in the run-up to SC|05 and proved very useful.

We plan to convert this script into a service, perhaps launched by LISA [5] during 2006. This type of monitoring is planned for deployment on OSG in the next major release.

Kernel Development

Kernels and the associated device drivers are very important to the achievable performance of hardware and software. In addition the FAST protocol implementation for Linux requires a modified kernel to work. For both of these reasons we have undertaken developing a "standard" UltraLight kernel as part of our project.

Significant progress has been made as of the end of 2005. We are now maintaining a web page with a few flavors of Linux RPMS for the kernel available from the UltraLight web site [1]. We currently support both an i686 (32 bit) and two x86_64 (64 bit, Opteron and EM64T flavors) kernel rpm sets. We are working on providing support for IA64 with the FAST TCP team.

As part of the kernel development process for UltraLight, we learned to deal with many pitfalls in the configuration and versions of linux kernels, particularly how they impact the performance of the system on the network. An example is that we found intrinsic problems in both the 2.6.13 and 2.6.14 kernels as we worked to develop and deploy an UltraLight kernel for SC|05. These problems may not have been easily noticed by the linux tcp kernel developers as they preferentially impacted wide-area network connections. This resulted in us reverting to a 2.6.12 version for the bandwidth challenge. These problems have been fixed in 2.6.15, partly due to feedback from our group and we plan to implement a new set of RPMS once 2.6.15 and FAST are ready.

MPLS/QoS Services and Planning

UltraLight is in the process of realizing its plans to explore the full range of end-to-end connections across the network, from best-effort, packet-switched through dedicated end-to-end light-paths. This is because the scientific applications supported by UltraLight have a wide variety of transfers that must be supported, ranging from the highly predictable (movement of large-scale simulated data between a few national centers) to the highly dynamic (analysis tasks initiated by rapidly changing teams of scientists at dozens of institutions). Current network engineering knowledge is insufficient to predict what combination of "best-effort" packet

switching, QoS-enabled packet switching, MPLS and dedicated circuits will be most effective in supporting these applications. We intend to engineer the most reliable and cost-effective combination of networking technologies, test them in our integrated environment, and begin deploying the resulting mix to meet the networking needs of the LHC community by first-collisions in 2007.
For UltraLight we are working to enable a combination of QoS on the LAN and MPLS "pipes" across the network to support such intermediate flows. In addition we are working closely with DoE funded efforts (like those of the TeraPaths project [6], and Lambda Station [7]) to find common extensible solutions to managing such virtual pipes across UltraLight. Shown in Figure 4 are the results of some tests showing QoS protected flows competing with unprotected flows.

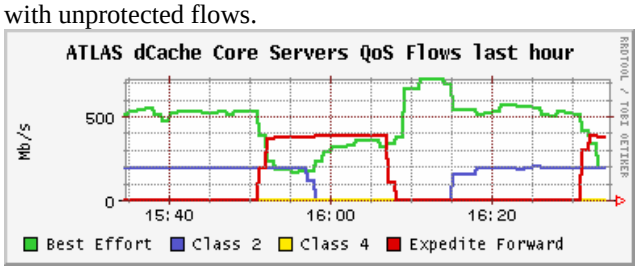

Figure 4 BNL to U. of Michigan: two bbcp disk-to-disk transfers (at 200Mb/s and 400Mb/s) against competing iperf traffic through ES-net MPLS tunnel.

Optical Path Management Status

We developed a multi-agent system for secure light path provisioning based on dynamic discovery of the topology in distributed networks. Autonomous software agents in this system act on behalf of services, managing access to resources, and collaborate with other services to generate the end-to-end optical path on demand. We used the MonALISA framework [4] for developing the Optical Control Plane system, and MonALISA services to host the distributed set of collaborating agents. The ensemble of services and agents provide near real-time access to complete monitoring information, analyze and process this information, and are able to feed the results to higher level services that provide decision support (and/or automated decisions) for workflow planning, or problem diagnosis and mitigation.
This prototype system (shown in Figure 5) is currently used with two types of optical switches, Calient [8] and Glimmerglass [9], and is able to create dynamically an end to end light path in less than one second independent of the number of switches involved and their location. It monitors and supervises all the created connections and is able to automatically generate an alterative path in case of connectivity errors. The alternative path is set up rapidly enough to avoid a TCP timeout, and thus to allow the transfer to continue uninterrupted.
We are working to further develop this distributed agent system and to provide integrated network services capable to efficiently use and coordinate shared, hybrid networks and to improve the performance and throughput for data intensive grid applications. This includes services able to dynamically configure routers and to aggregate local traffic on dynamically created optical connections. We are also developing agents able to interoperate with standard protocols (MPLS [10], GMLPS [11]) and other network services (Dragon [12] and UCLP [13]).

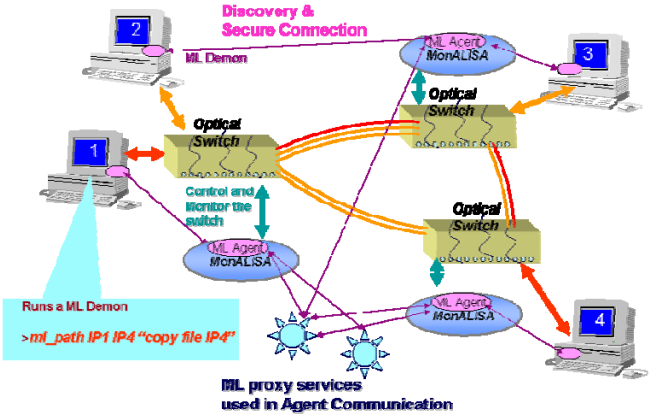

Figure 5 Diagram of the prototype optical switch management system.

High Speed Data Transfers Using "bbcp"

We have investigated the performance of several network file transfer tools, including gridftp, rootd, scp, bbftp and bbcp. The most straightforward to install, easy to configure, and performant tool was bbcp, which is capable of copying files at rates approaching line speed.
bbcp is a utility developed by Andy Hanushevsky at SLAC for the BaBar experiment. Full details may be found at the bbcp Web page [14]. The utility works as a peer-to-peer application, as opposed to the more usual client-server model used by other file transfer tools. This makes installation very simple: it is sufficient to place the bbcp executable in the path on each machine in the WAN that is participating in file transfers.
Initial tests with bbcp showed that data transfers between source memory and target memory could match (and in some cases exceed) rates obtained using the "iperf" network performance tool. For LAN transfers, using multiple streams offered the best aggregate rates, whereas in the WAN it was best to use one or two streams.
Some further information on bbcp, including tips on how best to set up transfers for maximum throughput, can be found in [15]. A presentation that describes our tests with bbcp in more detail can be downloaded from [16].

Disk-to-disk: Breaking the 1 GB/s barrier

One of the goals of UltraLight is to enable ~1 GB/s disk-to-disk data transfers across the UltraLight network. This is a critical capability for data intensive science and an area we think we can make significant contributions in.

Technology limitations

There is a huge gap in the current state of development between memory-to-memory and disk-to-disk transfers. This is essentially due to the end-hosts' resources (CPU power, bus bandwidth, I/O and memory bandwidth on the motherboard) being shared by both transmission and read/write tasks. The performance achievable separately from the end-host memory to disks, and that from memory to memory across the network, do not automatically translate into same level of performance for real disk-to- disk transfer across the network.

One of the most important performance limitations for disk to disk transfer currently comes from the PCI-X bus throughput. The theoretical peak bandwidth of the 64-bit/133MHz PCI-X bus is 1064MBytes/s (64[bits] ×133[MHz]). This bandwidth is larger than the sustained bandwidth achievable on a PCI-X bus because of the signaling which must take place to transmit data on the bus. Thus it is not possible to transmit TCP data at 1 GB/s across a network adapter inserted in a PCI-X slot.

Motherboards are now becoming available which utilize PCI-X 2.0 or PCI-Express (PCI-e). The PCI-X 2.0 [17] standard is only available (as of the beginning of 2006) on proprietary servers from IBM (x366, for example). PCI-X 2.0 doubles (or quadruples) the PCI clock to 266 MHz (or 512 MHz) thereby providing adequate bandwidth to match a 10 GE network adapter. PCI-e is a multilane serial standard and slots are speed rated according to how many lanes they support. An individual lane is a serial data connection running at 2.5 Gbits/sec. However the signaling utilizes a 10/8 bit code (10 bits are sent for each 8 bits of data) to allow error-detection and correction. Thus each lane can transmit 250 Mbytes/sec before signalling and protocol overhead. PCI-e is capable of approaching 95% of this bandwidth for transmitted data in the case of large, unidirectional data transfers. Both types of buses are being used in our UltraLight work.

Possible solutions

The best disk-to-disk performance we could achieve so far for a single TCP stream between CERN and Caltech, over an 11,000 km path, is 300 MBytes/s[‡] from disk to disk and 700 Mbytes/s from disk to memory. On the CERN side, we used HP 4-way 1.5 GHz Itanium2 systems and 3ware controllers. On the Caltech side, we used 2.4 GHz Opteron systems and Supermicro controllers. We will continue to work to improve these numbers and move towards tests in a production setting in the near future. Detailed reports on disk to disk performance are at http://ultralight.caltech.edu/d2d/.

Our next version of hardware will utilize 3Ware (9550SX) RAID controllers[18], dual dual-core Opteron motherboards and new PCI-e 10GE NICs from Myricom[19]. This 10GE NIC was demonstrated at SC|05 and was able to reach line speed with only 45% CPU usage on a 2.4 GHz Opteron system. New firmware which includes TSO and other enhancements is anticipated to drop the CPU usage to ~25% while reaching wire speed.

[‡] Using the Microsoft Window 2003 server operating system, we could transfer 1 TByte of data at 536 MBytes/s between CERN and Caltech.

CONCLUSION

The UltraLight project has had a productive year during 2005 deploying and extending both the physical networking infrastructure and the required services to effectively manage and utilize that infrastructure. We plan to more broadly deploy UltraLight technologies in time for LHC turn on in 2007 to ensure a dynamic, robust and manageable network for HEP.

Acknowledgements

This work is partly supported by National Science Foundation grants: PHY-0427110 (UltraLight), EIA-0303620 (Wan in Lab) and the Department of Energy grants: DE-FG02-05-ER41359 (LHCNet), DE-FG02-04ER25613 (Lambda Station). Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Department of Energy.

REFERENCES

[1] See http://www.ultralight.org
[2] http://netlab.caltech.edu/FAST/
[3] J. Wang, D. X. Wei and S. H. Low. *Modeling and stability of FAST TCP*. Proceedings of the IEEE Infocom, Miami, FL, March 2005
[4] MonALISA at http://monalisa.caltech.edu
[5] *LISA: Localhost Information Service Agent End to End Monitoring Tool*, available from MonALISA[4]
[6] http://www.atlasgrid.bnl.gov/terapaths/
[7] http://www.lambdastation.org
[8] Calient Networks http://www.calient.net
[9] Glimmerglass http://www.glimmerglass.com/
[10] Rosen, E., Viswanathan, A. and R. Callon, "Multiprotocol label switching Architecture", RFC 3031, January 2001
[11] E. Mannie, "Generalized Multi-Protocol Label Switching Architecture", draft-ietf-ccamp-gmpls-architecture-07.txt, November 2003
[12] The Dragon Project web page: http://dragon.east.isi.edu/
[13] UCLP (User Controlled LightPath Provisioning) web page: http://phi.badlab.crc.ca/uclp/
[14] http://www.slac.stanford.edu/~abh/bbcp/
[15] http://pcbunn.cacr.caltech.edu/bbcp/using_bbcp.htm
[16] http://pcbunn.cacr.caltech.edu/bbcp/UltraLight-Data-Transfers-v2.ppt
[17] http://www.pcisig.com/specifications/pcix_20
[18] See the 3Ware web site at http://www.3ware.com
[19]http://www.myricom.com/Myri10G/10gbe_solutions.html