UltraScience Net: Network Testbed for Large-Scale Science Applications

Nageswara S. V. Rao William R. Wing Steven M. Carter Qishi Wu

***Abstract*— UltraScienceNet is an experimental wide-area network testbed to enable the development of networking technologies required for the next-generation large-scale scientific applications. It provides on-demand, dedicated, high bandwidth channels for large data transfers, and also high resolution, high-precision channels for fine control operations. In the initial deployment, its data-plane consists of several thousand miles of dual 10 Gbps lambdas. The channels are provisioned on-demand using layer-1 and layer-2 switches in the backbone and multiple service provisioning platforms at the edges in a flexible configuration using a secure control-plane. A centralized scheduler is employed to compute the future channel allocations, and a signaling daemon is used to generate the configuration signals to switches at appropriate times. The control-plane is implemented using an out-of-band virtual private network, which encrypts the switching signals and also provides authenticated user and application access. Transport experiments are conducted on a smaller test connection which provided us useful information about the basic properties and issues of utilizing dedicated channels in applications.**

***Index Terms*— Network testbed, dedicated channels, SONET, 10GigE WAN-PHY, control-plane, data-plane, bandwidth scheduler.**

I. Introduction

The next generation of large-scale scientific applications involve expensive and powerful resources such as supercomputers, experimental facilities, and massive storage systems [4], [13]. Often these resources are created with a mission to support the scientific community that may span across several countries, for example, Earth Simulator [7] or Spallation Neutron Source [17]. In these applications, the scientific progress may depend on an adequate network access to these facilities to move data across wide-area networks and also to steer computations and experiments from remote sites. In fact, in some cases inadequate network connectivity – in terms of both bulk and stable bandwidths – may create resource bottlenecks, thereby falling short of reaching the full potential of these valuable resources.

The high-performance networking requirements for these large-scale applications belong to two broad classes: (a) high bandwidths, typically multiples of 10Gbps, to support bulk data transfers, and (b) stable bandwidths, typically at much lower bandwidths such as 100s of Mbps, to support interactive, steering and control operations. Currently, the Internet technologies are severely limited in meeting these demands. First, such bulk bandwidths are available only in the backbone, typically shared among a number of connections that are unaware of the demands of others. Second, due to the shared nature of packet switched networks, typical Internet connections often exhibit complicated dynamics, thereby lacking the stability needed for steering and control operations [14]. In both cases, the problem of transport becomes particularly difficult due to challenges in adapting TCP: it is extremely hard to sustain 10s of Gbps throughputs over wide-area links or to stabilize its dynamics even at lower bandwidths.

It is generally believed that the above networking demands can be effectively addressed by providing on-demand dedicated channels of the required bandwidths directly to end users or applications. However, networks with such capabilities cannot be readily deployed now using only the existing networking technologies, for most of them have been developed for the Internet. Note that Internet is based on packet-switched paradigm wherein packets from various sources simultaneously share the network, which is in stark contrast with the dedicated channels that share the network across time. Indeed, a number of diverse component technologies are needed to realize such a capability to support infrastructure, provisioning, transport, and application access. While technologies for infrastructure and application interfaces can be significantly leveraged from the existing ones, certain provisioning and transport technologies require significant development [5]. Furthermore, these technologies must be tested and demonstrated to be effective under realistic connections since current simulations are limited for such special networks. Thus there is a need for a testbed that can provide adequate environments for developing these technologies with an objective of providing these capabilities on-demand to the end users or applications.

The UltraScience Net (USN) is commissioned by the U. S. Department of Energy (DOE) to facilitate the development of these constituent technologies specifically targeting the large-scale science applications carried out at national laboratories and collaborating institutions. Its main objective is to provide developmental and testing environments for a wide spectrum of the technologies that can lead to production-level deployments within the span of next few years. In fact, some portions of USN may be left in place or merged into production networks if they prove to be effective. There are a number of testbeds such as UCLP [20], CHEETAH [2], and DRAGON [12] that provide dedicated channels. Compared to them, USN has a much larger backbone bandwidth (20-40 Gbps) and larger footprint (several thousands of miles), and a close proximity to several DOE facilities.

USN provides on-demand dedicated channels: (a) 10Gbps channels for large data transfers, and (b) high-precision channels for fine control operations. User sites can be connected to USN through its edge switches, and can utilize the provisioned

The authors are with the Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, raons@ornl.gov

dedicated channels during the allocated time slots. In terms of layer-1 backbone connectivity, USN's design and deployment is similar to the Internet, but it's control-plane is quite different mainly due to the ability of users and applications to setup and tear down channels on-demand. Its design required several new components including a Virtual Private Network (VPN) infrastructure, a bandwidth and channel scheduler, and a dynamic signaling daemon. In this paper, we briefly describe the design considerations and deployment details of these components.

In the initial deployment, its data-plane consists of dual 10 Gbps lambdas, both OC192 SONET and 10GigE WAN PHY, of several thousand miles in length from Atlanta to Chicago to Seattle to Sunnyvale. The channels are provisioned on-demand using layer-1 and layer-2 switches in the backbone and multiple service provisioning platforms at the edges in a flexible configuration using a secure control-plane. In addition, there are dedicated hosts at the USN edges that can be used for testing middleware, protocols, and other software modules that are not site specific.

The control-plane employs a centralized scheduler to compute the channel allocations and a signaling daemon to generate configuration signals to switches. Due to access to users and applications, the control-plane raised a number of security issues that are not addressed in conventional IP networks. This control plane is implemented using a hardware based VPN that encrypts all signals on the control plane and also provides authenticated and authorized access.

The dedicated channels are quite appealing in addressing the above network demands, but our current operational knowledge of utilizing them is quite limited, particularly for large bandwidth connections over long distances. While USN is being rolled out, we conducted preliminary experiments to understand the properties of dedicated channels using a smaller scale connection from Oak Ridge to Atlanta. Despite the limited nature of this connection, several important performance considerations have been revealed by these experiments. We describe these results here due to their relevance in utilizing the channels that will be provided by USN. Particularly, we describe experimental results on large data transfers and stable control streams over a dedicated 1Gbps channel of several hundred miles length implemented over ORNL-Atlanta production OC192 link. The performance profile generated from traffic measurements on this channel indicates non-zero packet losses and non-trivial jitter levels, both of which must be accounted for by the transport protocols to ensure high throughput and robust performance. We describe a UDP-based transport protocol by leveraging existing methods to achieve close to 100 percent channel utilization for file and data transfers. We also tested an existing protocol for implementing stable control streams over this channel. These results provide valuable insights into both the channel and host aspects of supporting data transfers over dedicated links.

This paper is organized as follows. In Section II, we describe the high-performance networking demands of large-scale science applications and the limitations of current infrastructures and technologies in meeting them. The overall configuration and footprint of USN are described in Section III. The details of USN's data-plane are described in Section IV. The basic modes of utilizing USN's data paths and hosts are described in Section V. The details about the control-plane are described in Section VI. The transport experiment results on Oak Ridge-Atlanta connection are described in Section VII.

II. High-Performance Networking

Supercomputers such as the new National Leadership Class Facility (NLCF) and others being constructed for large-scale scientific computing will reach speeds approaching 100 teraflops within the next few years. They hold an enormous promise for meeting the demands of highest priority projects including climate modeling, combustion modeling, and fusion simulation. They are also critical to other large-scale science projects and programs, which span fields as diverse as earth science, high energy and nuclear physics, astrophysics, molecular dynamics, nanoscale materials science, and genomics. These applications are expected to generate petabytes of data at the computing facilities, which must be transferred, visualized, analyzed by geographically distributed teams of scientists. The computations themselves may have to be interactively monitored and actively steered by the scientist teams. In the area of experimental science, there are several extremely valuable experimental facilities, such as the Spallation Neutron Source, the Advanced Photon Source, and the Relativistic Heavy Ion Collider. At these facilities, the ability to conduct experiments remotely and then transfer the large measurement data sets for remote distributed analysis is critical to ensuring the productivity of both the facilities and the scientific teams utilizing them. Indeed, high-performance network capabilities add a whole new dimension to the usefulness of these computing and experimental facilities by eliminating the "single location, single time zone" bottlenecks that currently plague these valuable resources.

Both classes of the above applications require next generation network capabilities in terms of multiple 10Gbps channels, which are currently offered as single lambda services, namely OC192 or 10GigE WAN PHY. For sub-lambda speeds of low-bandwidth low-jitter control channels, the requirements of both usable bandwidth and precise control are extremely difficult to meet over the current Internet. This is primarily due to the shared nature of these TCP/IP networks, which leads to unpredictable traffic levels and the complex transport dynamics. By utilizing dedicated channels over switched circuits these difficulties can be almost, if not completely, eliminated.

The existing testbeds for exploring such high bandwidth or fine control channels typically have a very small footprint or bandwidth, and hence do not provide adequate development environments for the required high performance networking tasks. In particular, they do not completely reflect the operational effects of cross-country links operating at full capacity, which are critical to optimizing these protocols and applications. Furthermore, these testbeds are not field hardened for high-performance production deployments and cyber defense. That is, they do not include multiple layers of redundant control to deal with such real world events as the inevitable power failures and controlled recovery from them, or the need to defend against cyber attacks to subvert the control plane. USN

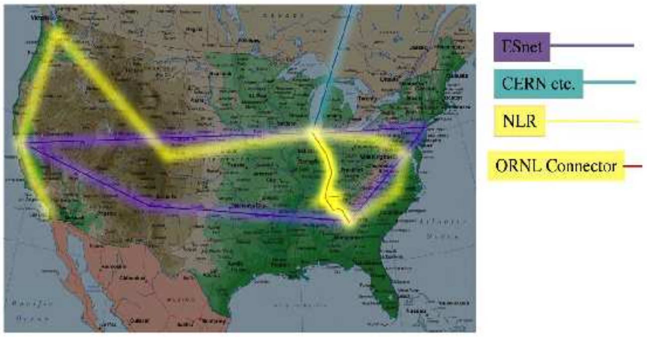


Fig. 1. UltraScience Net backbone consists of dual 10 Gbps lambdas from Atlanta to Chicago to Seattle to Sunnyvale.

is designed to address these networking requirements and the limitations of existing testbeds.

III. UltraScience Net Backbone

The requirements described in the previous section led directly to the design of UltraScience Net, which is an infrastructure testbed to facilitate the development of the capabilities needed for supporting distributed large-scale DOE science applications. It links Atlanta, Chicago, Seattle and Sunnyvale as shown in Figure 1, where each connection is supported by two and four 10 Gbps long-haul links in the first and second phases, respectively. These sites are chosen for their close proximity to various DOE science national laboratories and collaborating universities. Atlanta site provides proximity to Oak Ridge National Laboratory (ORNL); Chicago site provides proximity to Argonne National Laboratory (ANL) and Fermi National Laboratory (FNL); Seattle site provides proximity to Pacific Northwest National Laboratory (PNNL); and Sunnyvale site provides proximity to Stanford Linear Accelerator Facility (SLAC) and Lawrence Berkeley National Laboratory (LBNL). Our expansion plans include extending USN to New York to provide proximity to Brookhaven National Laboratory (BNL). Also, Atlanta and Chicago sites facilitate peering with ESnet [8], Internet2 [18] and connectivity to CERN [1]. However, USN provides the connectivity only between the above four sites, and the individual institutions provide their own connections to these USN edge sites including the required equipment such as linecards.

USN utilizes the ORNL network infrastructure to provide two OC192 SONET connections from Atlanta to Chicago in phase one; this connection is approximately 1000 miles in network length. Also, the lambdas from National Lambda Rail (NLR) are utilized from Chicago to Seattle to Sunnyvale; this connection is about 2000 miles in network length. In phase one, initial deployment consists of 10GigE WAN-PHY connections from NLR, which will be augmented with OC192 SONET connections. The complete network, including the bandwidth supplied by ESnet and the backup capacity provided by NLR is shown in Fig 1. First phase deployment of the data-plane with two 10 Gbps backbone connections is expected in early 2005, and the second phase with 40 Gbps backbone is expected to be completed in 2006.

IV. Data-Plane

The data-plane of USN shown in Figure 2 consists of two dedicated OC192 SONET (10 Gbps) connections between Atlanta and Chicago in phase one. These two lambdas are terminated on OC192 linecards of core switches at both sites. These switches can house additional OC192 and 10/1 GigE linecards that terminate connections from the user sites or peered networks. These switches can dynamically cross connect the linecards to realize SONET-SONET or GigE-SONET connectivity to USN from the user sites or peered networks. For example, the OC192 connection from ORNL will be terminated on a OC192 linecard on Atlanta switch, and can be cross-connected to OC192 connection to Chicago.

The OC192 and 10 GigE WAN-PHY connections between Chicago and Seattle will terminate at the core switches at the respective sites. In Chicago, the core switches are capable of “connecting through” or terminating the connections from Atlanta or Seattle. The terminating connections may be cross-connected to the linecards that carry connections to user sites (FNL or ANL) or ESnet or CERN. While SONET connections can be carried through the core switches, for some connections SONET-10GigE media conversion may be needed in Chicago since connections to Atlanta are solely SONET-based. The Seattle-Sunnyvale connections are both 10GigE and OC192 SONET, and will terminate at the core switches at the respective sites. The core switches in Seattle can realize SONET-SONET and 10GigE-10GigE through connections between Chicago and Sunnyvale. They can also terminate connections from Chicago and Sunnyvale on linecards that connect to PNNL, and they can also implement SONET-10GigE media conversion.

Multi Service Provisioning platforms (MSPP) are located at USN edges as shown in Fig. 2, which provide SONET and Ethernet channels at finer resolutions. In general, USN provides on-demand dedicated channels at multi-, single- and sub-lambda resolutions between its core switches and MSPPs, which are generically referred to as USN *switches*[1]. The schematic in Figure 2 is generic in that core switching and MSPP functions may be supported by a single device or two devices. The SONET channels can be provisioned at OC1 granularity depending on the core switches and the MSPPs that constitute the channel. Similarly, the channels provisioned entirely through GigE connections can be rate limited at the resolutions supported by the switches. For hybrid channels, the resolutions will be appropriately translated and aligned. The channels for user connections can be provisioned exclusively through the core switches or can utilize the MSPPs at the end points of the channels. Typically, user sites that need dedicated channels between them will provide their own connectivity to USN and request suitable dedicated channels. Since the provisioned channels are typically at layer-2, the user sites need to suitably set up layer-3 devices and modules to support IP services such as sockets or ftp.

Currently, there are two primary mechanisms for wide-area connections, namely SONET and 10Gig WAN-PHY. SONET connections have been utilized in most Internet deployments for

[1] Due to the on-going procurement process, the exact devices and their manufacturers could not be included at this time, and will be included in the next version of this paper within next few months.

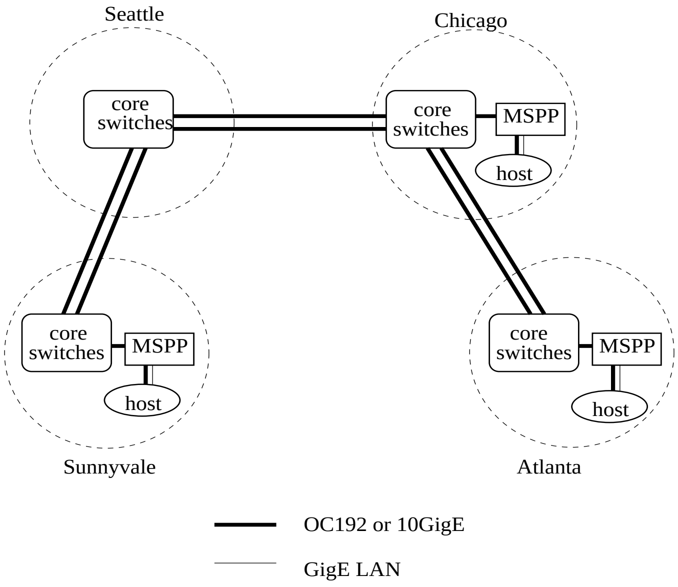


Fig. 2. Data Plane of UltraScienceNet consists of core switches and MSPPs.

the past several years, and are considered fairly mature technology. 10GigE WAN-PHY is a relatively new technology, but is quite promising in part due to lower cost of deployment. On the other hand, its performance over wide area connections is not completely well-understood. By utilizing both technologies for long-haul connections, USN provides an infrastructure where they can be studied in-depth by carefully designed experiments particularly in terms of the performance they deliver to the large-scale science applications. In particular, specific to dedicated control channels of finer resolutions, these two technologies must be analyzed in detail. At the surface, SONET multiplexing seems to provide more stable bandwidth particularly at sub-lambda resolutions due to its time-division multiplexing and strict reshaping. On the other hand, 10GigE connections can be rate limited at the switches to realize sub-lambda rates, but lack of strict time-division multiplexing has a potential for introducing higher levels of jitter. It is an open issue as to whether such jitter levels will negatively impact the application performance, and these issues can be investigated using USN channels.

USN also provides Linux hosts connected to MSPPs as shown in Figure 2 to provide environments to support the development and testing of protocols, middle ware, and applications. Users can be given accounts on USN hosts so that software can be downloaded onto them, and development and testing can be carried out by setting up appropriate channels on USN data-plane. In this mode, user can have access to the dedicated channels of various resolutions at distances ranging from few hundred miles (from user sites to USN) to thousands of miles (on USN data-plane), and the testing can be carried out in a site neutral manner.

V. User and Application Support

UltraScience Net is based on the concept of giving users and applications a direct access to layer-1 light paths with zero packet re-ordering, zero jitter, and zero congestion. In addition, it also provides dedicated level-2 paths with low re-ordering rate, low jitter and no congestion. In this sense, it is an implementation of the research network described in the DOE Roadmap Workshop document [6]. Its underlying networking technologies are guided by the DOE workshop on provisioning and transport areas [5]. Users can provision USN dedicated channels through a bandwidth scheduler as needed by their tasks. The channels might be utilized for tasks as varied as file transfers, computations scheduled on supercomputers, testing new protocols or middleware, or developing techniques for remote visualization. User sites connect their hosts or subnets to USN channels through their own specialized connections to core switches or MSPPs. They may need to support the underlying layer-3 capability if IP services need to be executed transparently. In the simplest case, GigE Network Interface Cards (NICs) of two hosts may be connected to the end-points of a dedicated USN channels as shown in Figure 3(a). Then IP connectivity between the two hosts may be ensured simply by forwarding the destination packets to the NICs and appropriately making the arp entries. By utilizing Ethernet switches that are VLAN-enabled it is possible to utilize multiple USN channels as in Figure 3(b), and also realize multiple subchannels over a single USN channel by VLAN tagging as in 3(c).

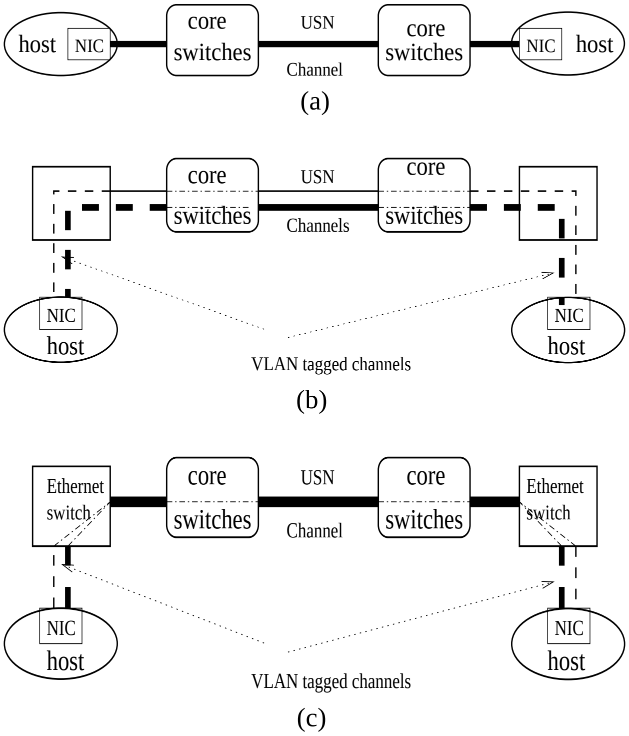


Fig. 3. Connecting hosts to USN channels: (a) NICs directly connected to core switches, (b) VLAN tagging to utilize multiple USN channels, and (c) VLAN tagging to share single USN channels.

On the other hand, when subnets are connected to the end points of a USN channel, the connected routers must be suitably configured to appropriately forward the IP packets as shown in Figure 4. Once such layer-3 configurations are made, various types of protocols, middleware and application modules can make use of the provisioned dedicated circuits. However,

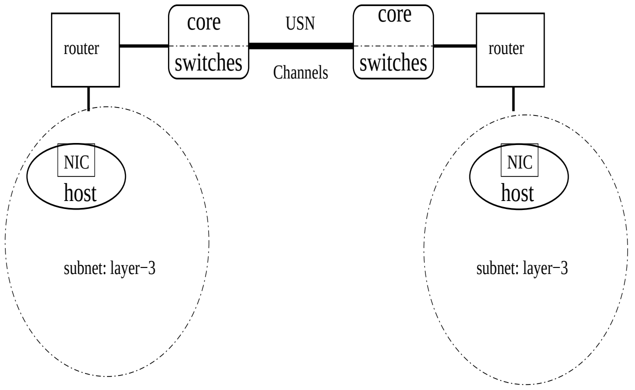


Fig. 4. Connecting subnet to USN channels.

strict USN cyber policies and guidelines are to be followed by the user sites before they are allowed to connect to USN and request channels. Users can utilize hosts located at USN edges as regular users to test protocols, middleware and other application level technology. In this mode, users will be provided accounts on the hosts and to the scheduler that will enable them to setup channels between the hosts.

VI. Control-Plane

A control-plane is needed for facilitating a number of USN functions:

(a) monitoring, configuration and recovery of its core switches, MSPPs, and hosts,
(b) providing user access to USN hosts, and user/application access for requesting channel setup and obtaining state information about hosts and channels,
(c) signaling for on-demand setting-up and tearing down of the dedicated channels, and
(d) facilitating peering with other networks, particularly those that support user/application controlled paths.

In conventional IP networks, a control-plane is employed for the function (a), which is typically implemented out-of-band using proprietary vendor technologies. Such a control-plane provides access only to network operators and typically supports (infrequent) manual configuration of various switches and routers, all of which are typically produced by the same vendor. The functions (b)-(d) above distinguish USN from the current production IP networks to a large extent.

USN accepts user requests for scheduling dedicated channels in future time-slots and grants them based on the bandwidth availability and feasibility constraints. This task involves scheduling the bandwidth on various connections to compose the requested channel, and also deriving the cross-connections at the core switches and MSPPs. Various allocations and cross-connection information is stored on a central server located at ORNL. A signaling daemon on this server constantly monitors the allocations and sends configuration signals to the constituent switches to setup and tear down the channels. The ability of the applications to actively access the control plane of USN has posed unique challenges that are not faced by the Internet and also not directly addressed by existing methods. Recall that the end-points of data-plane are connected to user sites as per USN cyber security guidelines, and the data channels are accessed by only such "physically" connected sites. On the other hand, users that request the channels must be able to access the control-plane to acquire information needed to generate the channel requests. Such users/applications can be located anywhere on the Internet. The ability to affect the channels and switch configurations potentially opens the whole infrastructure to cyber attacks. For example, if sent in the clear such requests can be sniffed and crafted requests can be injected to hijack the control streams. Furthermore once hijacked, the recovery can be prevented through denial-of-service attacks on the control-plane. Thus, users and applications must be appropriately authenticated and authorized before their requests can be granted, and in addition, all traffic between users and control-plane must be encrypted.

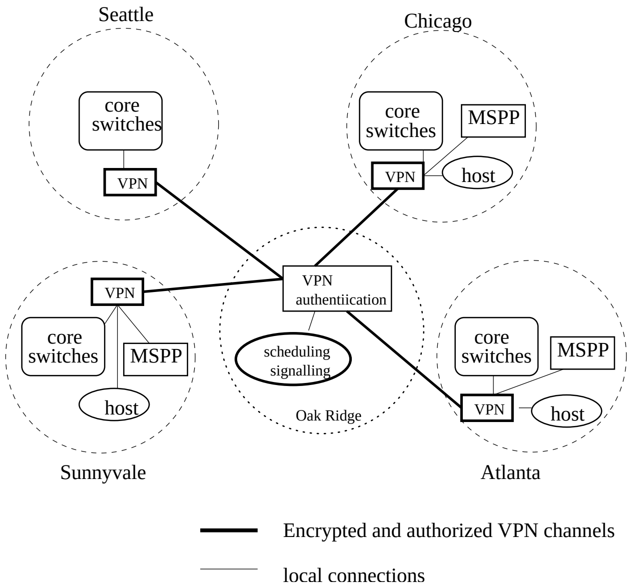


Fig. 5. Control-plane supported on VPN.

In terms of the signaling, the switches accept TL1 or GMPLS (Generalized Multiple Protocol Label Switching) commands to setup and tear down the channels and realize cross connections on demand. USN switches accept only clear text TL1 or GMPLS commands through their management interfaces, which can be easily sniffed and crafted packets can be injected by any one having access to their ports. Most of these devices do not support IPSec or ssh services because traditionally these interfaces are accessible only through a proprietary control-plane with access only to network operators. Thus, the configuration and other commands from the central signaling daemon need to be encrypted so that they cannot be sniffed or altered; furthermore, access to these signaling paths must be protected against the injection of crafted packets. Thus, the control-plane must allow only the authenticated and authorized entities to send and receive the signaling messages.

The control-plane operations are coordinated by a centralized system located at ORNL that: (a) maintains the state of bandwidth allocations on each link, and also the cross-connection configuration information for each core switch and MSPP; (b) accepts and grants requests for current and future channels to applications by suitably composing the segments with required bandwidths; and (c) sends signaling messages to switches as required by the schedule for setting up and tearing down the dedicated channels. The control-plane is supported by a number of hardware VPN units, which securely relay the TL1 signals to the respective switches from the signaling daemon. This scheme facilitates the immediate deployment using interfaces currently available across all USN switches.

A. VPN Implementation

We have designed a control-plane using a VPN shown in Figure 5, which serves the purposes listed in (a)-(d) above. The VPN is implemented in hardware using a main unit (Netscreen NS-50) at ORNL and secondary units (Netscreen NS-5) at each of the remote sites. A VPN tunnel is configured between the main unit and each of the secondary units so that only authenticated and authorized traffic is allowed on each of the tunnels, and the traffic is encrypted using IPSec. Each VPN tunnel carries three types of encrypted traffic flows: (i) user access to hosts, (ii) management access to hosts and switches, and (iii) the signaling messages to switches. The users are provided authenticated access to the VPN through the main unit, and in addition hosts require ssh logins. The management host at ORNL is authenticated and located in the secure domain of NS-50 so that monitoring and related traffic is secured. The signaling server is also located within the secure domain of NS-50 so that the signaling messages are secured via the VPN tunnels. This scheme protects using IPsec all the three types of traffic against sniffing and altering of packets, and also prevents the injection of crafted attack packets by third parties through the access control at NS-50. The channel requests are handled by a secure https server located on the ORNL server, which itself is located within the secure domain of NS-50. Users are authenticated and authorized to access the https server through NS-50.

B. Bandwidth Scheduler

We now briefly describe the bandwidth scheduler that allocates the channels to various requests. Note that MPLS and GMPLS technologies only provide mechanisms to set up channels at the time of request using OSPF-TE and RSVP-TE, respectively. Neither would allow setting up channels in future time-slots. Our scheduler to facilitate future allocations is based on our previous work on the quickest path problems under time-varying bandwidths [10].

The scheduler can be used to check the availability of a channel of specified bandwidth b between two ports located on core switches or MSPPs during a time-slot of duration t in future. It can also list all time-slots during which such channel with bandwidth b is available for duration t. USN is represented as graph $G = (V, E)$ where each node represents a core switch or MSPP, and each edge represents a connection such as OC192 or 10GigE WAN-PHY. Parallel edges are allowed to reflect multiple connections, and each node $v \in V$ is provided the information about which of its edges can be composed to form a channel. For each edge $e \in E$, we store a list R_e of bandwidth reservations as a piecewise constant function of time. We now outline the *all-slots* version of the scheduler that lists all available time-slots for a channel of bandwidth b from port p_s of node s to port p_d of node d. For each $e \in E$, we generate a list L_e of non-disjoint intervals such that bandwidth b is available on e for duration t starting any time within any interval. The algorithm is essentially the well-known all-pairs shortest path algorithm [3] with the modification to utilize the lists L_e's in the computation. Let the nodes be denoted by $1, 2, \ldots n$, and $\mathcal{L}^k[i, j]$ denote the sequence of disjoint intervals listing all starting points of a channel of bandwidth b and duration t from the appropriate ports of nodes i to j only through nodes $1, 2, \ldots k$. Thus $\mathcal{L}^n[s, d]$ lists all slots during which the required channel of bandwidth b and duration t is available. The outline of the algorithm is as follows; for simplicity we skip the initialization and the details corresponding to cross-connection information at the nodes.

```
algorithm ALL-SLOTS;
1. for $k = 1, 2, \ldots, n$ do
2.     for $i = 1, 2, \ldots, n$ do
3.         for $j = 1, 2, \ldots, n$ do
4.             $\mathcal{L}^k[i, j] \leftarrow \mathcal{L}^{k-1}[i, j] \bigoplus \{\mathcal{L}^{k-1}[i, k] \bigotimes \mathcal{L}^{k-1}[k, j]\}$;
5. return($\mathcal{L}^n[s, d]$);
```

In the above algorithm the operation $\bigoplus$ corresponds to merging the intervals of the corresponding lists, and the operation $\bigotimes$ corresponds to computing the intervals obtained by composing the channels from i to k and k to j to form single channels from i to j. The complexity of this algorithm is polynomial in n. This algorithm is based on a special structure within the well-known closed semi-ring framework for shortest path problems [3]. In particular, the closed semi-ring of ALL-SLOTS is defined on infinite sequences of disjoint intervals, where $\bigoplus$ and $\bigotimes$ correspond to the summary and extension operations, respectively.

The scheduler has also presented interesting problems from a strategic point of view. Clearly, it would be a mistake to design a rigid scheduler that observed strict wall-clock bounds on when circuits were set up or torn down. For example, users may not know precisely when a job on a supercomputer will exit and make data available for transport. In addition, even known-size data sets may have unpredictable load times since the data is typically spread across multiple disks in a parallel file system and load times will vary from run to run. Furthermore, in steered computations, it is not always possible to know the run times in advance since the parameters may be specified on the fly. To accommodate these scenarios, one approach being considered for the scheduler is to allow the user to specify, not an absolute start and stop time for channels, but instead a window within which the transfer must be completed. Also, spare capacity on the links will be used to accommodate jobs that run beyond their allocated time slots.

VII. Experiments With Dedicated Channels

To optimally utilize dedicated channels provisioned by USN it is important to understand the channel properties and their interactions with the hosts, including NIC, kernel and application aspects. Our current experience of network protocols is mostly limited to the Internet environments. For dedicated channels in particular, the application-level experimental results are limited to testbeds with limited capacities and/or distances. As a preparatory phase for USN we set up a testbed with a dedicated 1Gbps channel between two hosts located at ORNL via an IP channel that loops back over ORNL-Atlanta OC192 link. Our objective is to perform experiments to understand the properties of dedicated channels as well as hosts for supporting data transfers at the rate of 1Gbps and stable control streams at significantly smaller bandwidths. Due to the scarcity of experimental results over realistic dedicated channels, our results provide a stepping stone for developing the technologies for USN channels (a more detailed account of these results can be found in [15]).

While the dedicated channels obviate the need for congestion control, there are a number of important issues that critically affect the network performance observed at the application level:

(a) **Capacity and Throughputs:** In general, the application level throughputs are smaller than the channel capacities due to channel and host losses, which in turn are a function of sending rates at the source. Consequently, it is suboptimal to a priori fix the source sending rate right at the channel capacity; instead, it must be maintained at a level to ensure the highest goodput at the destination.

(b) **Host Issues:** In addition to the link properties, a number of host components play a critical role in deciding the achieved throughputs or jitter levels, and their effects become particularly important at 1-10Gbps data rates. Because IP packets from the source application are copied into kernel buffers and then onto NIC, various buffer sizes together with the speeds and policies for clearing them can have an impact on the source rates and dynamics. The differences in the rates of NIC and provisioned channels can result in losses since most Ethernet cards do not support explicit rate controls. Consequently, the packets may experience losses or jitter, both of which could appear random to the sender or receiver.

(c) **Jitter and Stabilization:** When control operations are to be performed over network connections, it is very important that the packets flows be stable. Variations in delays, namely *jitter*, can destabilize transport flows and cause the loss of control. The lost packets have to be re-sent thereby increasing their net delays and contributing to jitter. While the losses over dedicated links are much less pronounced than over Internet, they still need to be explicitly accounted for in designing protocols for stable streams.

Effects of the above factors on applications and protocols can be assessed by conducting experiments over dedicated channels, which is a main focus of this section. We tested a number of existing protocols for high-throughput data transfers, including SABUL, tsunami, and UDT. In particular, a file transfer protocol was proposed in [15] with adjustable rate control parameters, which were manually tuned to achieve 990Mbps file transfer rates. We describe in this section results based on these protocols and also the protocol of [16] for implementing stable control flows.

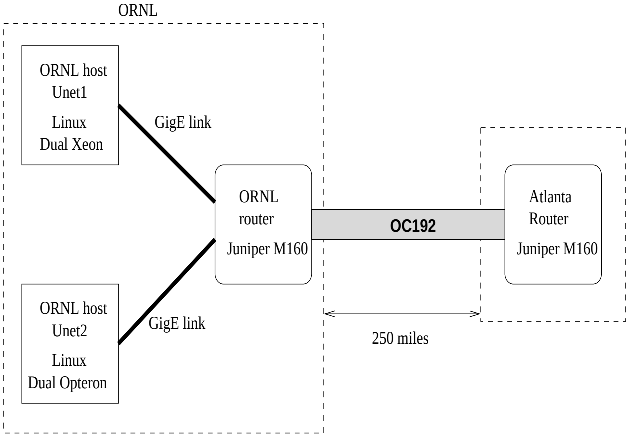


Fig. 6. ORNL-Atlanta-ORNL dedicated 1 Gbps IP connection.

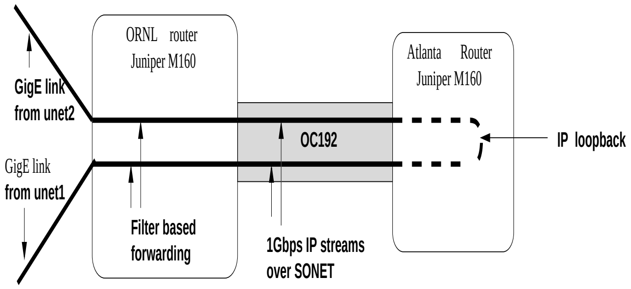


Fig. 7. Router configuration for implementing dedicated channel

A. Channel Provisioning

Our testbed consists of two hosts, called unet1 and unet2, both located at ORNL. Each of them is equipped with a dedicated NIC which is connected to a GigE slot on a linecard of Juniper M160 router located at ORNL as shown in Figure 6. There is an OC192 link from this ORNL router to another Juniper M160 router located in Atlanta, which is approximately 250 miles away. Only 1 Gbps of ORNL production traffic is currently carried on this OC192 link, and thus there is a spare bandwidth of 9 Gbps on this link. We utilize 2 Gbps of this spare bandwidth to implement a loop-back connection from ORNL to Atlanta back to ORNL. The traffic at each of the hosts is limited to 1Gbps due to the Ethernet connection. And the traffic flows from the hosts will flow unimpeded between the routers at ORNL and in Atlanta over the OC192 link. This arrangement effectively realizes a dedicated 1Gbps IP connection between unet1 and unet2 of approximately 500 miles in length.

Both unet1 and unet2 NICs have IP addresses belonging to a local subnet, and thus by default the IP packets between them are forwarded within the GigE linecard of the router itself. We changed this default routing so that the IP packets from each of these ports are statically forwarded to the output port of the OC192 linecard by utilizing the Filter Based Forwarding (FBF) capability of ORNL router. This is achieved by applying a firewall filter to each GigE port to incorporate a routing-instance that specifies the static route for all arriving packets to depart via the OC192 linecard.

The IP packets arriving at Atlanta router are handled by the

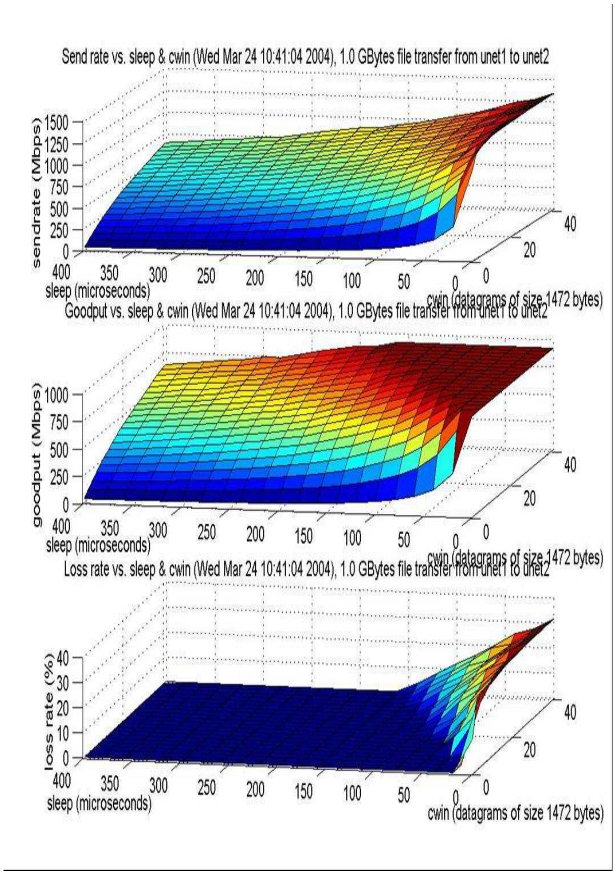


Fig. 8. Measurements for ORNL-Atlanta-ORNL dedicated 1Gbps IP channel. Each point in horizontal plane represents sending rate given by window size and idle time pair. Top plot corresponds to sending rate, middle plot corresponds to the goodput at the destination and the bottom plot corresponds to the loss rate.

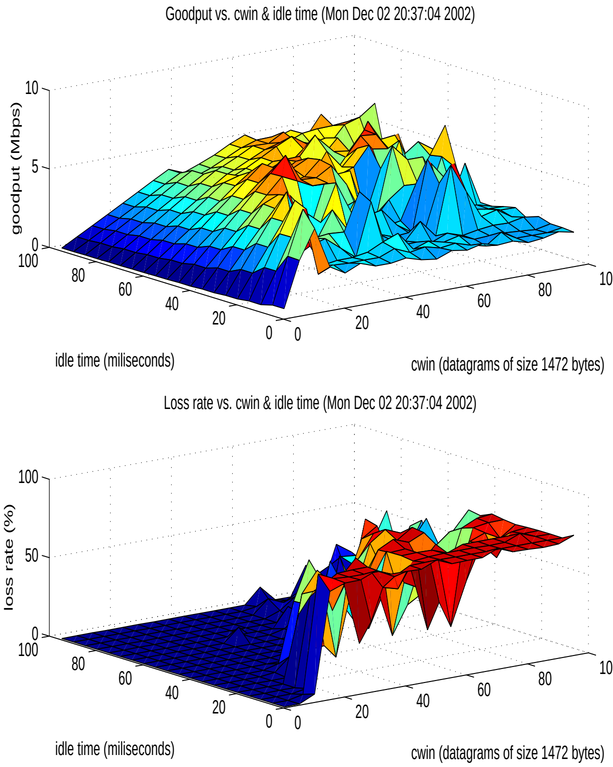


Fig. 9. Measurements for ORNL-LSU Internet Connection. Each point in horizontal plane represents a sending rate given by window size and idle time pair. Top plot corresponds to the goodput at the destination and the bottom plot corresponds to the loss rate.

default routes, namely, the packets from a ORNL host destined to other ORNL hosts are simply routed back at the OC192 linecard. Under this configuration, packets between unet1 and unet2 are routed along the loop-back path implemented over OC192 link as shown in Figure 7. While this configuration provides a dedicated 1Gbps channel between unet1 and unet2, it is not a lightpath or an MPLS tunnel in a strict sense. The underlying mechanism of this channel provisioning makes it more similar to an MPLS tunnel than a lightpath. On the other hand, when viewed from an end-host viewpoint, this configuration is quite similar to how typical PC hosts might be connected to utilize a dedicated USN SONET channel (lightpath), namely through an Ethernet interface as described in Section V.

1) Channel Characteristics: Using a UDP stream with varying sending rates we measured the effective throughput, called the *goodput*, at the destination, and also the loss rate. The sending rate is controlled by transmitting a number of datagrams, denoted by the *window size* $W_c(t)$, in a single burst and then waiting for a time period called the *sleep time* $T_s(t)$. Thus the sending rate is specified by a point in the horizontal plane, given by $(W_c(t), T_s(t))$, and its corresponding sending rate is shown in the top plot of Figure 8. The goodput measurements at the destination corresponding to various window size and sleep (idle) time pairs are shown in the middle plot, which is commonly known as the *throughput profile*. When the sending rate is small, the destination goodput increases with the sending rate, and reaches a plateau within the vicinity of 1Gbps as shown in the right hand side of the throughput profile. In the bottom plot, the loss rate is shown as a function of the window size and idle time. The loss rates are near zero when the sending rate is low, but they becomes significant when the sending rate reaches the vicinity of 1Gbps, where they monotonically increase with the sending rate. We also observed that the loss rates from multiple runs of an experiment with the same sending rate vary within a certain range even though the average trend was monotonic as shown in Figure 8.

Based on the measurements, one can draw two important observations:

(a) For throughputs around the vicinity of 1 Gbps, suitable sending rate must be computed to achieve goodput plateau with minimal loss rate. From a transport perspective, the lost packets have to be identified and re-sent, and this is a process which consumes CPU resources, particularly so at high throughput rates. It is important to minimize this overhead activity to optimize the throughput, and this in turn involves utilizing a sending rate at a minimal loss rate. On the other hand, extremely low loss rates can only be achieved when the goodputs are significantly below 1 Gbps.

(b) At all high sending rates, the losses are non-zero and random. Hence flow stabilization at these fixed target bandwidths requires explicit step size adaptation to achieve overall flow stability [16]. This stability is not particularly

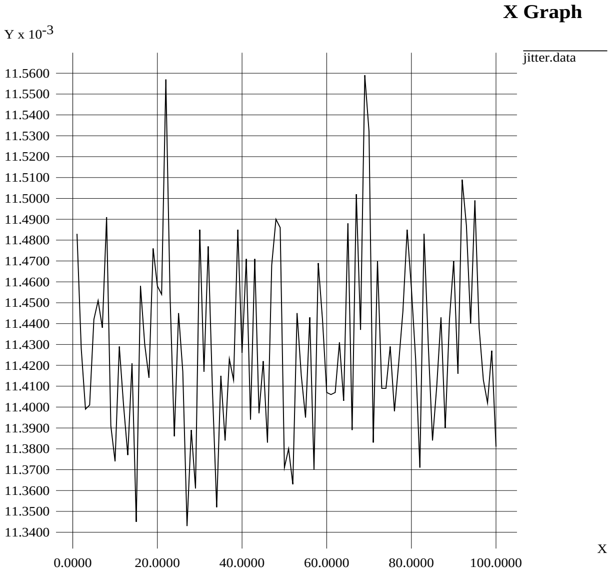


Fig. 10. Jitter levels over ORNL-Atlanta-ORNL dedicated channel

vital to data transfers but is extremely important in control streams.

It is instructive to compare this throughput profile with that observed for Internet connections [21]. The measurements collected over the Internet are shown in Figure 9 between ORNL and Louisiana State University (LSU). This connection runs over the OC192 link from ORNL to Atlanta, on Internet from Atlanta to Houston, and on a local network from Houston to LSU. There are two important features:

(i) There is an overall trend of increase followed by decrease in the goodput as sending rate is increased. This overall behavior is quite stable although the transition points vary over time. It is to be noted that goodput for the dedicated channel reached a plateau and remained constant afterwords. For Internet connections, the goodput actually decreased when the sending rate is increased beyond a certain level.

(ii) The plot is quite non-smooth mostly because of the randomness involved in packet delays and losses. The variation in the goodput is particularly high at high sending rates.

To estimate the jitter levels, we sent packets of fixed sizes (10K) between the hosts and measured the application level delays. The variations are shown in Figure 10. The average delay is approximately 11 millisec with jitter level of about 2%. While this jitter level is extremely low compared to Internet connections where the jitter levels could be as much as 30%, control streams for highly sensitive end devices could require an explicit handling of the jitter.

2) Host Configurations: The storage devices and file systems on unet1 and unet2 are carefully configured to achieve the file access speed of 1Gbps. Specifically, we implemented RAID 0 disk system on both hosts using dual SCSI hard drives and implemented xfs file system that achieved disk I/O rates in excess of 1 Gbps.

protocol	throughput
Tsunami	919 Mbps
UDT	890 Mbps
FOBS	708 Mbps

TABLE I

THROUGHPUTS ACHIEVED BY VARIOUS UDP-BASED PROTOCOLS FOR FILE TRANSFERS.

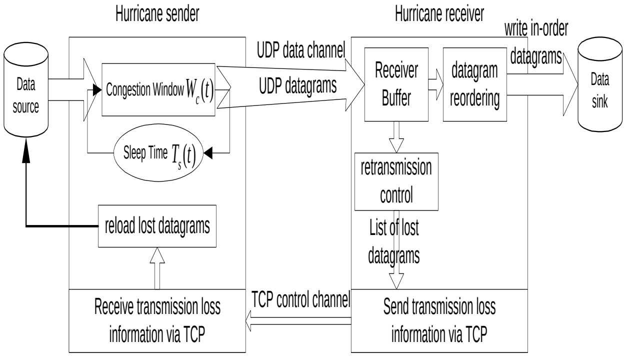


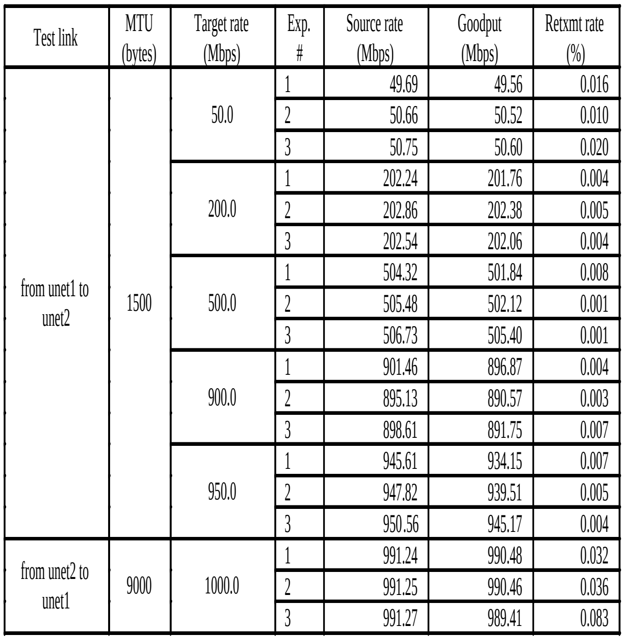
Fig. 11. Hurricane transport control structure.

Note that the measurements in the previous section are collected at the application level, and hence they are subject to processor scheduling between the application processes and also between application and kernel processes. The measurements could be significantly affected if other applications are concurrently running on the hosts since the processor is shared between them. The plots in Figure 8 were collected when no other user programs are executed at the hosts, and in that sense represent the best case performance experienced by the applications. Our motivation is to utilize unet1 and unet2 as dedicated hosts for data transfers. If hosts were to be used as user workstations as well, the throughput profile must be generated under the normal host conditions. In general, additional applications running on the host will result in higher application-level losses and lower goodputs. Also, jitter levels shown in Figure 10 are observed when no other user level processes are running.

B. Transport Protocols

We consider protocols for data transfers, both memory and file transfers, and stable control streams. The default TCP throughputs were below 100 Mbps and could be improved by a factor of 2-3 with parameter tuning. Since dedicated channels do not have competing traffic, UDP-based protocols are more suited for these channels, although a careful parameter tuning was necessary to achieve goodput rates in the vicinity of 1Gbps. All UDP protocols we tested for file transfers required some manual parameter tuning to achieve throughputs close to 900 Mbps; this process required some understandings of the protocols and their implementations as well. The details were different among the protocols and it required significant efforts to gain even a partial understanding of the relationship between the parameters and the achieved throughput.

For implementing stable flows, TCP is inherently ill-suited



Test link	MTU (bytes)	Target rate (Mbps)	Exp. #	Source rate (Mbps)	Goodput (Mbps)	Retxmt rate (%)
from unet1 to unet2	1500	50.0	1	49.69	49.56	0.016
			2	50.66	50.52	0.010
			3	50.75	50.60	0.020
		200.0	1	202.24	201.76	0.004
			2	202.86	202.38	0.005
			3	202.54	202.06	0.004
		500.0	1	504.32	501.84	0.008
			2	505.48	502.12	0.001
			3	506.73	505.40	0.001
		900.0	1	901.46	896.87	0.004
			2	895.13	890.57	0.003
			3	898.61	891.75	0.007
		950.0	1	945.61	934.15	0.007
			2	947.82	939.51	0.005
			3	950.56	945.17	0.004
from unet2 to unet1	9000	1000.0	1	991.24	990.48	0.032
			2	991.25	990.46	0.036
			3	991.27	989.41	0.083

Fig. 12. Hurricane transport test results on unet1 and unet2.

because by default it attempts to infer and occupy the available bandwidth, which is the entire channel capacity in case of a dedicated channel. The sending rate of TCP can be clipped to a desired level by suitably restricting the flow window sizes. If there are no losses, then TCP would indeed maintain the same sending rate. But as indicated by the throughput profile, the non-zero loss rates at various sending rates result in TCP underflow, since it interprets the loss as a congestion indication. Also, the randomness of the losses makes the TCP flow stabilization a difficult task. We tested the recently developed flow stabilization method [16] based on stochastic approximation which provided quite robust results as will be described in this section.

1) High Throughput Data and File Transfer: Recently researchers have been seeking solutions to develop UDP-based high-performance transport protocols that overcome TCP's throughput limitation. Such research efforts include SABUL, Tsunami, RBUDP, UDT and others (see [9] for an overview). We tested several of these protocols for file transfers and their peak throughput results are shown in Table I. The best performance we achieved for file transfers is slightly above 900Mbps. It was clear from the throughput profile that goodput rates of 990 Mbps are possible if the source rate is suitably maintained. A protocol, called *Hurricane* [15] is developed exclusively for high-speed file transfer on dedicated links. The design goal of Hurricane is to maximize link utilization without any expectation of sharing the channel. The architecture of Hurricane transport is illustrated in Figure 11. The source rate $r_S(t)$ of a Hurricane sender is controlled by two parameters, congestion window size $W_c(t)$ and sleep time $T_s(t)$:

$$r_S(t) = \frac{W_c(t)}{T_s(t) + T_c(t)} = \frac{W_c(t)}{T_s(t) + \frac{W_c(t)}{BW}} = \frac{1.0}{\frac{T_s(t)}{W_c(t)} + \frac{1.0}{BW}} \quad (1)$$

where $T_c(t) = \frac{W_c(t)}{BW}$ is the time spent on continuously sending a full congestion window of UDP datagrams, which is determined by the congestion window size and link capacity *BW*, i.e. the maximum speed at which the NIC can generate the bit signal and put it on wire. According to Eq (1), we may control the source rate $r_S(t)$ by adjusting either the congestion window $W_c(t)$ or sleep time $T_s(t)$ individually, or both simultaneously.

A Hurricane receiver accepts incoming UDP datagrams, which are either written immediately to the local storage device if they arrived in order, or placed temporarily in a buffer for reordering otherwise. Whenever a control event is triggered, a sequential scanning is performed on the receiving buffer to check for a list of missing datagrams. The datagram ID numbers on this list are grouped together and sent over a separate TCP channel to the Hurricane sender. The sender then reloads the missing datagrams into the sender congestion window for retransmission upon the receipt of such control strings. To account for the limitations posed by the host side factors, a retransmission event is triggered based on the number of missing datagrams within a strategically determined time window of multiple RTT (round trip time) estimates, and we write only in-order datagrams on the fly onto the local storage device to sustain a near-peak receiving rate.

We conducted Hurricane transport experiments on 1Gbps dedicated link between unet1 and unet2 with various levels of target rates using a 2G bytes test file. Each experiment on one target rate is repeated for 3 times. The performance measurements for file transfers are listed in Figure 12. The high throughput and bandwidth utilization are achieved in both cases with reasonably low loss rates. Also, we obtain quite stable throughput when targeting at low rates. The transport control parameters in these experiments were manually tuned for the best performance. We observed that the impact of parameter tuning on throughput and loss rate at source rates far below the peak bandwidth is not as sensitive as those approaching the peak bandwidth.

2) Stable Control Streams: The architecture of the stabilization protocol is similar to the one shown in Figure 11 except that the control channel for datagram acknowledgment is also built over UDP. The rate control is based on the Robbin-Monro Stochastic approximation method [11]. At time step $n+1$, the new sleep time is computed as follows to update the sending rate to a new value (this method is described in detail in [21], [16]):

$$T_{s,n+1} = \frac{1.0}{\frac{1.0}{T_{s,n}} - \frac{a/W_c}{n^\alpha} * (g_n - g^*)}$$

where g^* is the target rate and g_n is the goodput measurement at time step n at the sender side. Coefficients a and α are carefully chosen so that the source rate eventually converges to ensure the required target rate. The step size denoted by a/n^α must eventually become zero such that $a/n^\alpha \rightarrow \infty$ as $n \rightarrow \infty$. But the rate of change must be controlled to be neither too fast such that $\sum_{n=1}^{\infty} a/n^\alpha = \infty$, nor too slow such that $\sum_{n=1}^{\infty} a^2/n^{2\alpha} < \infty$. Under these Robbins-Monroe conditions on step sizes, it can be analytically shown that this protocol achieves the goodput stabilization at g^* under random losses and profiles similar to ones observed for this link (see [16], [21] for details).

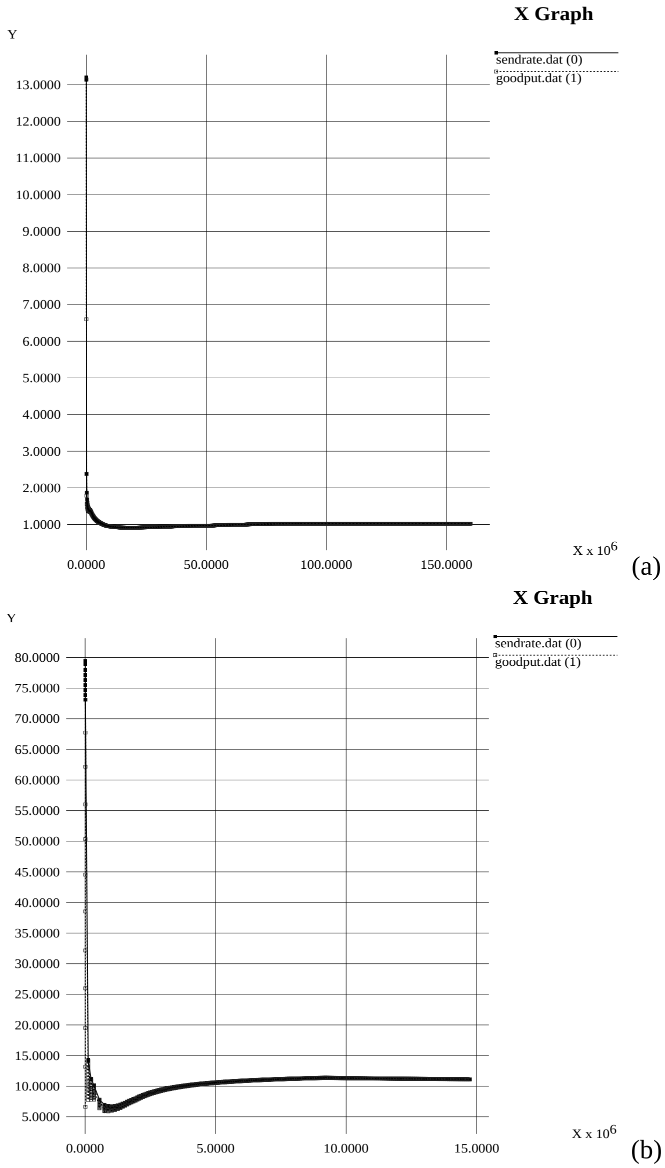


Fig. 13. Stabilization over the dedicated link: target goodput is 1.0 and 10.0 Mbps in (a) and (b) respectively; $a = 0.8$, $\alpha = 0.8$, adjustment is made on sleep time.

We tested this method for flow stabilization on the same dedicated channel between unet1 and unet2. There is no competing traffic on this dedicated channel during the time of experiments. A set of control parameters $a = 0.8$ and $\alpha = 0.8$ are selected and the rate adjustments are applied on sleep time only. Instead of using the default MTU of 1500 bytes in the Internet, we use a MTU of 9000 bytes on this dedicated link. We conducted two stabilization experiments targeted at 1.0 and 10.0 Mbps, respectively. The initial sleep time is set to be 100 ms for each experiment and the window size is fixed at 2 and 6 datagrams, respectively. The performance measurements of source rate and goodput are plotted in Fig. 13 where the time axis is in units of microseconds and the rate axis is in units of Mbps. In both cases, the goodput stabilized at the target rate within seconds and remained constant subsequently.

VIII. Conclusions

The high-performance networks for large-scale applications require high bandwidths to support bulk data transfers and stable bandwidths to support interactive, steering and control operations. Current IP technologies are severely limited in meeting these demands since they are geared to the packet-switched and shared networks. It is generally believed that the above networking demands can be effectively addressed by providing on-demand dedicated channels of the required bandwidths directly to end users or applications. The UltraScience Net's goal is to support the development of these technologies specifically targeting the large-scale science applications carried out at national laboratories and the collaborating institutions. USN provides on-demand dedicated channels: (a) 10Gbps channels for large data transfers, and (b) high-precision channels for fine control operations. Its design required several new components including a VPN infrastructure, a bandwidth and channel scheduler, and a dynamic signaling daemon. USN's initial deployment consisting of OC192 SONET and 10GigE WAN PHY connections from Atlanta to Chicago to Seattle to Sunnyvale is expected to be operational in early 2005. Its control plane is implemented using a hardware-based VPN that encrypts all the signals on the control plane and also provides authenticated and authorized access. Our future plan include enhancing the data-plane with four 10Gbps wide-area connections, and enhancing the control-plane to inter-operate with networks supported by GMPLS signaling. We also plan to provide level-2 peering with NSF CHEETAH network [2] using MSPP at ORNL, and lever-3 peering with ESnet and CERN in Chicago and with Internet2 in Atlanta.

While USN is being rolled out, we conducted preliminary experiments to understand the properties of dedicated channels using a smaller scale connection from Oak Ridge to Atlanta. While this 1Gbps channel is limited in its capacity, span and capabilities, these experimental results provided us with valuable insights into both channel and host aspects of supporting data transfers over dedicated channels. For USN dedicated channels, which are of much larger capacity and longer distance, we expect our qualitative results to hold although the actual loss and jitter levels might be quite different:

(a) The throughput profile will be qualitatively similar in that losses will be non-zero and random at various sending rates, and jitter levels could be significant for control streams.
(b) Host components play a significant role in the performance seen at the applications level.
(c) Achieving data transfer rates close to the channel capacities would require a careful selection of control parameters and appropriate implementation of protocols.

Our future plans include testing both protocols and applications over USN channels that connect ORNL supercomputer sites to user sites. In particular, our plans include developing and testing interactive visualization, monitoring and steering modules for Terascale Supernova computations [19] executed on ORNL supercomputers from remote locations connected via USN channels.

ACKNOWLEDGMENTS

This research is sponsored by High-Performance Networking Program of Office of Science, U. S. Department of Energy under Contract No. DE-AC0500OR22725 with UT-Battelle, LLC.

REFERENCES

[1] CERN: The World's Largest Particle Physics Laboratory. http://http://public.web.cern.ch/Public/Welcome.html.

[2] End-To-End Provisioned Optical Network Testbed for Large-Scale eScience Application, http://www.ece.virginia.edu/ mv/html-files/ein-home.html.

[3] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGraw-Hill Book Co., New York, 1990.

[4] High-performance networks for high-impact science, 2002. Report of the High-Performance Network Planning Workshop, August 13-15, 2002, http://DOECollaboratory.pnl.gov/meetings/hpnpw.

[5] Network provisioning and portocols for DOE large-science applications, 2003. Report of DOE Worksshop on Ultra High-Speed Transport Protocol and Dynamic Provisioning for Large-Scale Applications, August 10-11, 2003, http://www.csm.ornl.gov/ghpn/wk2003.html.

[6] Doe science networking - roadmap to 2008, 2003. Report available at: http://www.es.net/hypertext/welcome/pr.Roadmap/index.html.

[7] The earth simulator center. http://www.es.jamstec.go.jp/esc/eng/ES/index.html.

[8] Energy sciences network. http://www.es.net.

[9] A. Falk, T. Faber, J. Banister, A. Chien, R. Grossman, and J. Leigh. Transport protocols for high performance. *Communications of the ACM*, 46(11):43–49, 2003.

[10] W. C. Grimmell and N. S. V. Rao. On source-based route computation for quickest paths under dynamic bandwidth constraints. *Journal on Foundations of Computer Science*, 14(3):503–523, 2003.

[11] H. J. Kushner and C. G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, 1997.

[12] NSF Shared Cyberinfrastructure Division PI Meeting, February 18-20, 2004, http://http://hpn.east.isi.edu/nsf-sci.

[13] Nsf grand challenges in escience workshop, 2001. Final Report: http://www.evl.uic.edu/activity/NSF/final.html.

[14] N. S. V. Rao, J. Gao, and L. O. Chua. On dynamics of transport protocols in wide-area internet connections. In L. Kocarev and G. Vattay, editors, *Complex Dynamics in Communication Networks*. 2004.

[15] N. S. V. Rao, Q. Wu, S. M. Carter, and W. R. Wing. Experimental results on data transfers over dedicated channels. In *First International Workshop on Provisioning and Transport for Hybrid Networks: PATHNETS*, 2004.

[16] N. S. V. Rao, Q. Wu, and S. S. Iyengar. On throughput stabilization of network transport. *IEEE Communications Letters*, 8(1):66–68, 2004.

[17] Spallation neutron source. http://www.sns.gov.

[18] Internet2. http://www.internet2.edu.

[19] Terascale supernova initiative. http://www.phy.ornl.gov/tsi.

[20] User Controlled LightPath Provisioning, http://phi.badlab.crc.ca/uclp.

[21] Q. Wu. *Control of Tranport Dynamics in Overlay Networks*. PhD thesis, Louisiana State University, 2003.