SANDIA REPORT

SAND99–8223
Unlimited Release
Printed February 1999

Integrated Service Provisioning in an Ipv6 over ATM Research Network

Helen Chen, Jim Brandt, Eli Dart, Jerry Friesen, Jim Hutchins, Perry Robertson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND99-8223
Unlimited Release
Printed February 1999

Integrated Service Provisioning in an IPv6 over ATM Research Network

Helen Chen, Jim Brandt, Eli Dart, Jerry Friesen, Jim Hutchins, and Perry Robertson

Sandia National Laboratories
Albuquerque, NM 87185
Livermore, CA 94551-0969

Abstract

During the past few years, the worldwide Internet has grown at a phenomenal rate, which has spurred the proposal of innovative network technologies to support the fast, efficient and low-latency transport of a wide spectrum of multimedia traffic types. Existing network infrastructures have been plagued by their inability to provide for real-time application traffic as well as their general lack of resources and resilience to congestion. This work proposes to address these issues by implementing a prototype high-speed network infrastructure consisting of Internet Protocol Version 6 (IPv6) on top of an Asynchronous Transfer Mode (ATM) transport medium. Since ATM is connection-oriented whereas IP uses a connection-less paradigm, the efficient integration of IPv6 over ATM is especially challenging and has generated much interest in the research community. We propose, in collaboration with an industry partner, to implement IPv6 over ATM using a unique approach that integrates IP over fast ATM hardware while still preserving IP's connection-less paradigm. This is achieved by replacing ATM's control software with IP's routing code and by caching IP's forwarding decisions in ATM's VPI/VCI translation tables. Prototype “VR” and distributed-parallel-computing applications will also be developed to exercise the realtime capability of our *IPv6 over ATM* network.

TABLE OF CONTENTS

TABLE OF FIGURES

1. The Proposal

The Internet is currently connecting approximately 5 million hosts on 45 thousand interconnected networks covering 86 countries, and is continuing to grow. This unprecedented growth rate has presented a set of unique problems to the current version of Internet protocol (IPv4) [1]. Four of the biggest problems are: 1) the 32-bit address field, which is running out of space, 2) a three level (network, subnet, and host) addressing format that lacks the hierarchical structure to aggregate route entries in order to prevent routing table explosion, 3) the lack of Quality of Service guarantees that are necessary to support multimedia applications, and 4) the lack of scalability in its bandwidth capacity. These problems have spurred the proposal of innovative network technologies, which will support the fast, efficient and low-latency transport of a wide spectrum of traffic types. For example, the next generation IP (IPv6) [2] is designed with expanded address field (128 bits), and real-time capabilities that can rectify the immediate problems as well as meet future requirements. Using fast cell switching, ATM [3] offers high capacity, high link flexibility, and the ability to provide integrated services. While these ATM attributes can satisfy the performance requirements of future applications, in order to succeed in computer network, ATM must also provide good support to existing applications. Since the worldwide Internet is IP based, how to efficiently integrate IP and ATM is a hot research topic and has attracted much interest in the network community. This work will focus on the integration of the next generation IP and ATM.

Because ATM is connection-oriented whereas IP uses a connection-less paradigm, the efficient integration of IPv6 and ATM is especially challenging. Many proposals for supporting *IP over ATM* are under discussion in the networking community. ATM Forum's *LAN Emulation* (LANE) [4] proposes to emulate ATM as an IEEE 802 [5] based local area network. The *classical IP over ATM* [6] (CLIP) by the Internet Engineering Task Force (IETF) uses ATM as the physical layer for an IP network. Both of these proposals are interim solutions and will not scale to large networks as they rely on IP routers to reach different IP subnets, even though they are connected via the same ATM physical network. Consequently, they cannot take advantage of the fast ATM hardware and suffer from having a single point of failure at the IP router. In order to reap the benefits of the fast ATM switching hardware, the *Multiple Protocol over ATM* (MPOA) protocol [7] is currently being defined at the ATM Forum. This approach glues connection-less IP networks to the connection-oriented ATM network and treats the ATM backbone as an opaque cloud. In this configuration, all routers at the edge of the ATM cloud are one hop away from each other; they achieve connectivity using a fully meshed topology. The result is large routing tables with N^2 complexity in all edge routers. Perhaps the most significant flaw in the MPOA approach is the loss of flexibility and robustness inside the ATM cloud. If a path should fail within the ATM cloud, all existing virtual connections and their applications will be interrupted due to the connection-oriented nature of ATM. Not to mention the duplication (IP and ATM) in administering two sets of addresses, routing tables, and troubleshooting.

We propose, in collaboration with an industry partner, to implement IPv6 over ATM using a unique approach where we discard the ATM control software and integrate the fast ATM hardware directly with IP, thereby preserving the connection-less nature of IP. The huge success of IP in recent years is largely due to the simplicity and robustness of its connection-less paradigm. IP forwards data-grams in a hop-by-hop fashion without establishing an end-to-end connection. This approach eliminates the need for complicated signaling software and provides the fault tolerant feature of rerouting. Therefore, it is the goal of this design to preserve IP's connection-less paradigm.

This proposal will take standard ATM hardware and completely change its control software. In place of the ATM signaling (Q2931.Q) [8], ATM routing (PNNI) [9], and LANE software, we will load a standard IP router package into the switch control computer in order that each switch will operate in a connection-less manner. In addition, we will add IP flow management software to match IP flows with ATM's VPI/VCI [3] labels and a driver to control the switch hardware. The result is a router with attached switching hardware that can cache routing decisions in the switch virtual circuit (VC) translation table. We believe that this approach combines the simplicity, scalability, and robustness of IP with the speed, capacity, and integrated service capability of ATM.

IPv6 guarantees real-time performance by allowing applications to reserve bandwidth. It implements the real-time functionality by using either the *flow label* or the *priority* field in its header. A flow label is used to identify IP packets that have the same source-destination tuple and traverse a fixed communication path. Using this label, an IP stream that requires special treatment can be identified in order to receive its guaranteed QoS. An IP stream that desires to reserve resources can either use the RSVP [10] protocol or the hop-by-hop IPv6 extension header. While RSVP is used by the receiver initiated process, reservation in the forward direction can be achieved using the IPv6 hop-by-hop extension header. Alternatively, IPv6 can guarantee QoS via the marking of its priority field. This mechanism relies on an IP router to practice prioritized and fair queuing. Real-time applications must also cooperate by implementing either adaptive or hierarchical encoding. A real-time application that uses adaptive encoding throttles its rate in response to network's congestion indication. On the other hand, an application that uses hierarchical encoding allows the network to selectively drop packets based on the priority markings during congestion.

We intend to conclude this project with a *proof of concept* phase where the performance of this high-speed infrastructure will be evaluated using experimental procedures. While the IP performance can be evaluated using existing applications, the real-time performance requires the implementation of new applications and the understanding of their QoS requirements. We will prototype a few "VR" applications, which will be developed using the leading World Wide Web (WWW) technology, such as JAVA and VRML. Multimedia streams, over our high-speed network infrastructure, will also be inserted to provide audio as well as video to the virtual environment. We envision that such an environment can provide a 3-D perspective to an engineer who is using Pro-Engineer to design a mechanical part. Meanwhile, audio/video clips of engineering advisors, test data, and computer simulations could be displayed interactively using the high-speed multi-service network. We will also evaluate the speedup factor that this solution can offer to distributed, parallel, computing in solving large scientific problems. We believe that the real-time capability in IPv6 together with ATM's high bandwidth and low latency can provide acceptable cost/performance tradeoffs to traditional parallel computing methods. In addition, research tools will be developed to help characterize real-time traffic, understand their QoS requirements, and quantify the necessary network resources in order to meet those requirements.

2. Background Study

We present, in this section, the abstract of two background studies, the "IP-over-ATM architectures" [11] and the IPv6 experiences that we have gained through the collaborative experiments over the 6BONE [12] testbed.

2.1 "A Survey of IP over ATM Architectures", SAND 97-8273

This study contrasted the advantages and disadvantages of ongoing, standard as well as proprietary IP-over-ATM architectures. The following paragraphs present an abstract of the work, detailed analysis can be found in Sandia report 97-8273.

Both the IETF and the ATM Forum proposed overlay architecture to support IP over ATM. This architecture requires duplicate address space and routing protocols, obscuring the ATM infrastructure from IP, and incurring management overhead as well as complexity. It models the ATM infrastructure as a logical, non-broadcast, multi-access (NBMA) medium, where broadcast and multicast are emulated using a server-based mechanism and complex protocols. Therefore, the NBMA approach suffers from implementation complexity as well as performance bottleneck and single point of failure at the server. Nevertheless, the NBMA model allows location-independent network configuration over local and wide area, using its support for virtual local area networks (VLAN) across the ATM infrastructure.

Many ATM vendors implement the IETF Classical IP and the ATM Forum Local Area Network Emulation model. Both models lack the mechanism to take advantage of

ATM's cut-through paths, and therefore may experience performance bottlenecks at intermediate routers. In addition, they cannot access ATM's QoS capability and lack the multicast functionality frequently used by multimedia applications. In order to overcome these limitations, both IETF and ATM Forum are working on extensions to their current architecture. Ongoing work includes the IETF Multicast Address Resolution Server (MARS) [13] and the Next Hop Resolution Protocol (NHRP) [14], as well as the ATM Forum Multi Protocol over ATM (MPOA) approach.

Two proprietary IP-over-ATM approaches, Ipsilon's IP Switching [15] and Cisco System's TAG Switching [16], model the ATM infrastructure as point-to-point links. Since today's protocol already works on point-to-point links, they will continue to function over these solutions; thus, there is no need to emulate broadcast and multicast with new protocols and server machines. Moreover, by supporting standard routing packages in ATM switches, these models can preserve IP's connection-less paradigm and, at the same time, enjoy ATM's hardware speed. However, both models may have scalability problems in large Internet environments because of the large state information that they require. Details on these issues can be found in SAND 97-8273.

2.2 The IPv6 Study

In order to gain in-depth understanding of the IPv6 protocol, we participated in the 6BONE project, an informal collaborative effort sponsored by the Internet Engineering Task Force. The 6BONE testbed is a virtual IPv6 network layered on top of the current IPv4 Internet. This virtual network consists of islands of IPv6 capable workstations and subnets linked by virtual point-to-point links called "tunnels". These tunnels provide Internet-wide IPv6 transport in order to test the IPv6's network addressing policy, IPv4-to-IPv6 transition procedures, IPv6 routing, and its Application Programming Interface (API).

Over the course of this study, we were able to share valuable IPv6 experiences with researchers from 27 countries spanning North America, Europe, and Japan via the 6BONE mailing list. For example, earlier, we discovered a discrepancy of one API argument-passing specification between a public domain API library routine and the IETF draft document [17]. This finding was submitted to the 6BONE working-group in order to ensure source code portability of future IPv6 applications. New and modified application programming interfaces (API's) are now available to support existing IP applications and to provide new real-time applications with access to the *flow label* and the *priority* fields in the IPv6 header. Since IPv6 is very similar to IPv4, the core socket functions remain unchanged. The API's that changed include the addressing data structures, name-to-address translation functions, and address conversion functions. In addition, we followed the IPv6 standardization processes and adopted research software packages from the public domain whenever possible. Currently, most vendors implement a dual IP stack in their operating system (OS), thereby providing connectivity to IPv4 as well as IPv6 networks. The following IPv6 capable applications are also available through the public domain: telnet, FTP, SMTP (sendmail), Domain Name Service (DNS), Web Server and Client, traceroute, ping, finger, inetd, TTCP, routed, RIP, tcpdump, gated, TFTP, X11, RPC, NFS, POP, SSH, SNTP, TCP wrapper, PTCP, snoop, rlogin, rsh, rcp, rdist, tcptest, rdate, VIC, VAT, SDR, INN, SNMP, etc.

3. The Integrated Service Network

IPv6 was designed to solve the address exhaustion and the routing-table explosion problems in the global Internet. Unfortunately, IPv4's large installed base complicates its transition to IPv6. The current market chose to embrace an intermediate solution, the Classless Inter Domain Routing (CIDR) [18] that offers a temporary relief to the problems mentioned above. This trend influenced Ipsilon, our research partner, to also adopt this temporary solution in place of IPv6. Thus, we were forced to change our research direction to implement Differentiated Services on IPv4 Switching networks that are CIDR capable. The following sections describe the IPv4 Switching experiment and the new DiffServ architecture [19] being defined at the IETF.

3.1 The SC97 Demonstration

The objective of this exercise is to provide an operable interactive remote visualization environment (Figure 1) using the following leading-edge technologies: 1) a Silicon Graphics (SGI) ONYX2 graphics server, 2) scientific visualization applications, 3) an IPv4 switching networking environment, combining the robustness and simplicity of IP with the speed and quality of service (QoS) capabilities of ATM, and 4) a hardware-based MPEG-2 compression and decompression system.

Using a Long Link Error emulator (LLE), we simulated a wide area network by adding 30 ms of round-trip delay between Livermore, CA and the San Jose Convention Center. While our primary visualization hardware, an SGI ONYX2, was located at Sandia National Labs in Livermore, CA, users at the San Jose Convention Center received the visualization image and manipulated properties of the image as if the remote display were located at the Convention Center. This effect was achieved by transmitting MPEG-2 encoded screen images from the ONYX2 to be decoded and displayed on the workstation, an SGI O2, at the Convention Center. The SGI Networked Dual-head Software Demon (ndsd) provided users at the Convention Center with the ability to manipulate properties of the image on the ONYX display. Together with network quality of service guarantees, this approach enhances the performance by limiting the visualization response time to only the propagation delay between sites. This is a tremendous improvement compared to typical delays (up to 10's of minutes) of X-based remote visualization techniques; their large delay is the result of the cumulative round-trip times of a very large number of X requests and responses.

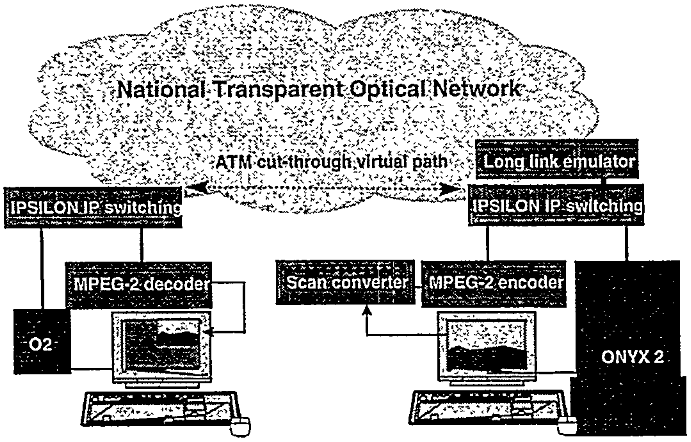

Figure 1. The SC97 demo topology

Our demonstration components are as follows.

- An SGI ONYX2 visualization server. The visualization server performs the compute-intensive post processing, rendering, and displaying of large data sets from previous scientific computations.
- The SGI Networked Dual-head Software Demon (NDSD). The NDSD accepts keyboard and mouse input from a networked machine, thereby allowing remote manipulations of local display.
- Ipsilon's IP switching network. Ipsilon's IP switching is a connectionless approach which integrates IP with fast ATM hardware. It combines IP's simplicity and robustness with ATM's speed and QoS capabilities. The varying traffic types generated by our application include the periodic bursts of the MPEG-2 encoded images and the small amounts of data associated with the mouse and keyboard movements.
- A Chromatek down converter. The Chromatek down converter converts the high-resolution RGB input of the ONYX console display into NTSC signal.
- An Optivision MPEG-2 system. The hardware-based MPEG-2 compression and decompression system takes the NTSC signal from the scan converter and compresses it using the MPEG-2 scheme before transmission. The resulting image quality is one half-digital studio quality video (CCIR-601) and the frame rate is 30 frames per second. The highest attainable compression ratio is about 30:1. At the client side, the compressed stream is recovered using an MPEG-2 hardware decoder.

- An SGI O2. The SGI O2 workstation provides remote users the display, and the control of the mouse and keyboard of the ONYX server.

3.2 The Differentiated Service Architecture

The IETF DiffServ architecture is designed to offer differentiated services to satisfy the performance requirements of a diverse class of applications in the fast growing Internet. This architecture is based on a simple model where traffic entering a network is classified and conditioned at the boundary of the network, and assigned to different behavior aggregates, each identified by a single differentiated service (DS) codepoint. Within the core of the network, packets are forwarded according to the per-hop behavior (PHB) associated with the DS codepoint. The PHB is the means by which a node allocates resources to a behavior aggregate, and it is on top of this basic hop-by-hop resource allocation mechanism that useful differentiated services are constructed. DiffServ achieves scalability by decoupling and implementing the complex traffic classification and conditioning functions at the network boundary, thereby allowing nodes in the core to simply forward packets according to the DS codepoint marked in their header

The DiffServ architecture consists of functional components to perform packet-classification, traffic conditioning (i.e., metering, marking, shaping, and policing), and per-hop forwarding behaviors. Figure 2 depicts the logical view of a packet classifier and traffic conditioner, where the packet classifier selects packets in a traffic stream based on the content of some portions of the packet header such that individual packets can be marked and steered for further processing. A traffic profile describes the temporal properties of a traffic stream; it is used to meter packets to determine whether they are in- or out-of-profile according to their contractual agreement. The results of metering will be passed to the traffic marker for appropriate DS codepoint assignments. Based on the DS codepoint, the traffic conditioner can either delay or drop some or all of the packets in a traffic stream in order to bring it into compliance with its traffic profile. While the first action is appropriate for elastic applications that are insensitive to delay and jitter, dropping is preferred for real-time applications that have stringent timing requirements.

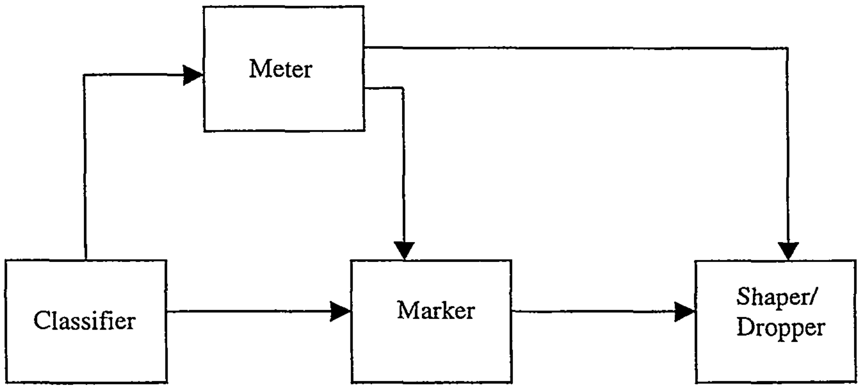

Figure 2. Logical view of a Packet Classifier and Traffic Conditioner.

3.3 The DiffServ Testbed

Over the past several years, Gigabit Ethernet [20] has gained more market than ATM as a solution for local area networks. As a result, this trend further influenced Ipsilon's business direction away from the original, ATM-based, IP Switching approach. We, therefore, had to modify our research objective yet a second time, instead of implementing integrated services over an IP Switching network, we choose to adopt the emerging IETF Differentiated Service architecture (DiffServ) over networks, independent of their physical transport mechanisms.

To this end, we constructed a Fast Ethernet [21] based testbed implementing the Class Based Queuing (CBQ) [22] traffic management mechanism designed by Van Jacobson of Cisco systems. This testbed consists of two 400 MHz Pentium-2 PCs running a modified FreeBSD 2.2.6 kernel by Sony. This kernel implements a built-in framework that can support alternative queuing mechanisms [23] in order to achieve resource sharing and to guarantee quality of service. Currently, it supports only Class Based Queuing. CBQ enforces its control policies based on a configuration file. A typical configuration file contains CBQ commands to setup a number of traffic classes with unique identifiers, the associated resources, priorities, and filters (used to classify outgoing data packets). Future releases will include RSVP, Weighted-Fair-Queuing (WFQ) [24], etc. To avoid congestion, the kernel queue management system uses the random early discard (RED) algorithm [25] to drop packets when queue sizes reach a preset threshold, such that the probability of packet-drops from a connection is roughly proportional to its share of the bandwidth. Additionally, CBQ employs an explicit congestion notification (ECN) [26] mechanism to quench a greedy source, when the congested condition can not be relieved through RED alone.

Preliminary results show that the CBQ's packet processing can keep up with the 100 Mbps link speed, however, a 10-microsecond latency was added to UDP echo packets. Figure 3 depicts the test configuration file we used to evaluate the performance of CBQ's bandwidth guarantees. As shown, a default class was assigned with a guaranteed bandwidth of 5 Mbps, and was allowed to borrow otherwise unused bandwidth to achieve efficient link utilization. The remaining 4 classes were assigned 10, 20, 30, and 40 Mbps of peak bandwidth and increasingly higher priorities, respectively. Note that these classes are not allowed to borrow additional resources. For this test, we started 5 Netperf streams, one per service class, from a source machine that also functioned as the CBQ traffic shaper and conditioner. As shown in Figure 4, the default flow was started to run for the entire duration of this test, approximately 20 seconds. Subsequently, the 10-, 20-, 30-, and 40-Mbps classes were launched, at 2-second intervals, to run for about 10 seconds each. Our results indicate that each stream received its guaranteed bandwidth and the default stream was able to soak up whatever bandwidth unused during the entire test, thereby achieving near full link utilization.

```
# Author: Eli Dart, eddart@ca.sandia.gov
# Interface specifications
#    Device          Bits/sec  total   Scheduler-type
interface  de0       bandwidth 100000000   cbq

# Class specifications
# Define the root class first -- all others borrow
from it
class cbq de0 root_class NULL priority 0 admission
none pbandwidth 100

# The control class contains the protocols RSVP, IGMP
and ICMP.
# Control class must come before default class!
class cbq de0 ctl_class root_class priority 1
admission none pbandwidth 5 control

# The default class
class cbq de0 def_class root_class borrow root_class
priority 2 admission none pbandwidth 5 default

#class cbq de0 tcp_class root_class priority 2
admission none pbandwidth 10
#filter de0 tcp_class 0 0 0 0 6

# Other classes
class cbq de0 port_1112 root_class priority 5
admission none pbandwidth 5
filter de0 port_1112 0 1112 0 0 0
filter de0 port_1112 0 0 0 1112 0

class cbq de0 port_2223 root_class priority 5
admission none pbandwidth 10
filter de0 port_2223 0 2223 0 0 0
filter de0 port_2223 0 0 0 2223 0

class cbq de0 port_3334 root_class priority 5
admission none pbandwidth 20
filter de0 port_3334 0 3334 0 0 0
filter de0 port_3334 0 0 0 3334 0

class cbq de0 port_4445 root_class priority 5
admission none pbandwidth 30
filter de0 port_4445 0 4445 0 0 0
filter de0 port_4445 0 0 0 4445 0
```

Figure 3. Test CBQ configuration

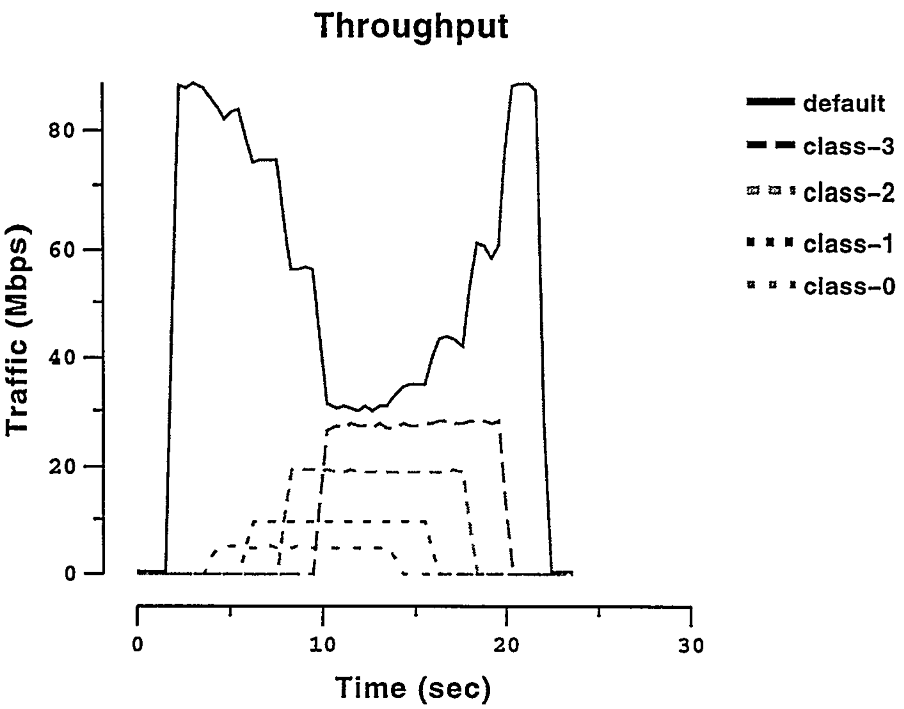

Figure 4. CBQ Bandwidth guarantees

4.0 Scientific Visualization Applications

The Department of Energy (DOE) Advanced Strategic Computing Initiative (ASCI) is chartered with the stewardship of the nation's nuclear stockpile using computer simulation, creating a science-and-engineering problem of unprecedented size and complexity. Likewise, the Strategic Simulation Initiative (SSI) will also apply tera- and peta-scale computing resources to atmospheric and combustion simulation by the year 2004. These simulation studies will produce results that are voluminous, multivariate, and complicated, and consequently difficult to understand. Scientific visualization transforms raw data into graphical representation, exploiting a person's high-bandwidth visual system and remarkable pattern recognition ability. It is the best means available today for making sense of large, complex scientific datasets. However, because of their size and complexity, those problems will challenge existing visualization, data management, system architecture, and networking technologies. New approaches are essential to making advances in these endeavors. We will describe, in the following sections, the current and developing visualization techniques at Sandia.

4.1 Client-server mode

Given sufficient network bandwidth, the client-server based visualization package (Figure 5) such as Ensight can provide acceptable performance to interactive users. However, this model precludes remote users from accessing server machine's powerful graphics hardware. Using Ensight, the server machine post processes user data and transports the resulting geometry data to the remote desktop for rendering and display. Therefore, the visualization performance and quality are limited by the graphics hardware of the desktop.

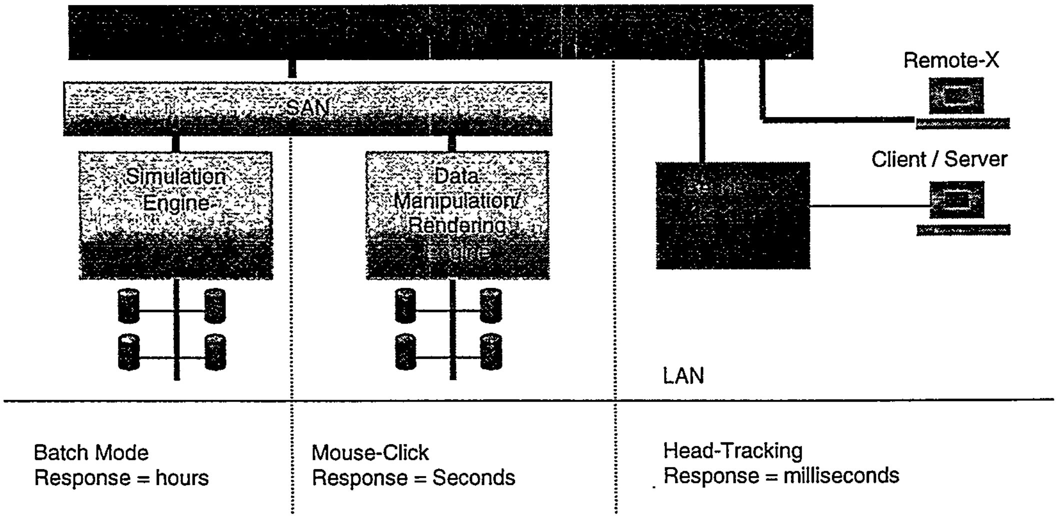

Figure 5. The existing scientific visualization architecture

4.2 Remote X-Display

For historical reasons, many of Sandia's interactive visualization tools (e.g. CTH plot, BLOT, MUSE-EIGEN, etc.) are not distributed. These tools provide graphics display to remote users using remote X-windows (Figure 3). Depending on the platforms used, either the display bitmaps or GL commands are sent over the network, which can be bandwidth intensive and cause a performance bottleneck. In addition, because the rendering is done in software at the server, its performance will further degrade with respect to interactive response time. Because interactive visualization sessions tend to exchange a large number of X- and GL-commands, interactive users will experience substantial latency due to the large round trip times typical of wide area networks. Furthermore, the visualization quality is limited by the graphics hardware of the desktop machine.

4.3 Remote Console Experiments

Two experiments were conducted in an attempt to improve the performance of remote visualization over LAN or WAN. Both approaches strive to achieve this goal by taking advantage of the graphic server's hardware speed, and by eliminating cumulative round trip delays caused by the numerous X and GL commands exchanged during interactive visualization sessions. These experiments create a virtual console environment (Figure 6), which gives remote users the ability to manipulate the properties of the visualization image on the server display as if it were local.

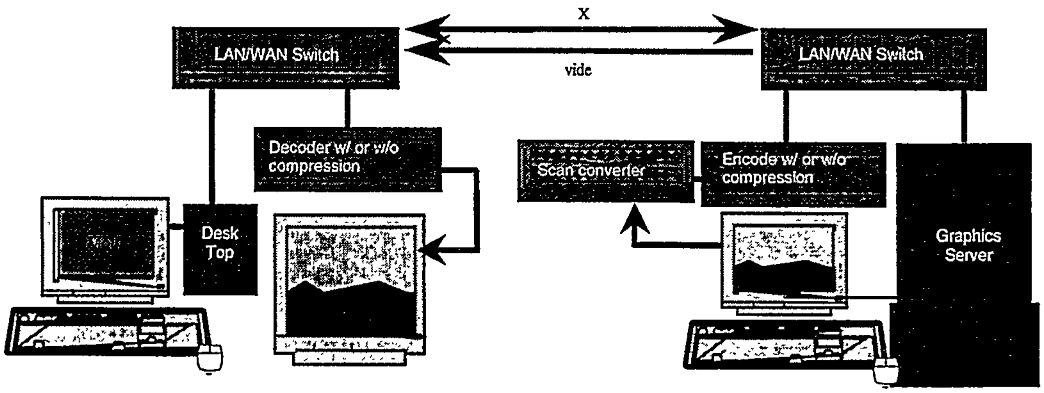

Figure 6. The remote console architecture

4.3.1 The MPEG-2 experiment

A scan converter took the analog signal of the graphic server's display and converted it to NTSC frames. The NTSC frames were then digitized, compressed, and streamed over the network using an MPEG-2 encoder. At the receiving end, an MPEG-2 decoder reversed the process and sent the recovered analog signal to the monitor of a remote desktop. Meanwhile, a software utility, SGI's ndsd, was used to transfer the control of the mouse and keyboard from the server to the same desktop, thereby allowing the remote user to manipulate the server's display as if it were local. While this approach is innovative, we found that its display quality was not acceptable because of the loss of resolution during the conversion from the display (1280x1024) to NTSC (640x480), and the degradation caused by the MPEG-2 compression algorithm. The MPEG-2 impact was especially deleterious to the text display because of its tendency to lose high-frequency information, which blurs the text's edges. Furthermore, MPEG-2's inherent delay (>250ms) introduced synchronization problems between the server's mouse movements and the delayed screen display. Since the mouse commands were transmitted via a separate TCP stream over the WAN, its susceptibility to network delay further exacerbated this problem.

4.3.2 The motion JPEG project

This project was designed to remove the synchronization problem between the mouse commands and the screen display in order to reduce the quality degradation of displayed text experienced by the MPEG-2 experiment. This experiment required modification to the CTH plot software to separate the graphical images and the control graphical user interface (GUI) in different windows. This approach allowed graphics frames to be compressed using motion JPEG, independent of the control GUI. Because motion JPEG did not exhibit inter-frame dependency, it eliminated the excessive encoding latency experienced by MPEG-2 and therefore the synchronization problem. Furthermore, this separation enabled the control GUI to be delivered using remote X-window, which offered high-quality text display.

4.4 Ongoing Research Projects - Transmission of Computer Visualization

While the motion JPEG project has addressed two of the three performance issues experienced by the MPEG-2 experiment, it still suffered from the loss of resolution caused by the conversion from the display to NTSC signal, and the lossey JPEG and

MPEG-2 compression algorithm. The following research project develops methods and designs to transmit images of high-resolution computer screen, up to 1280x1024 pixels, across either a high-speed local area network such as Gigabit Ethernet or a wide area network such as ATM. This work concentrates on aspects of video capture and the subsequent regeneration.

The bandwidth requirements for full resolution remote visualization are typically on the order of one gigabit per second (see below):

$$DataRate(bps) = 24bpp \times (1280 \times 1024 pixels/frame) \times 30 fps = 943{,}718{,}400 \text{ bps}$$

It should be apparent that this is a high bandwidth problem and will require specialized hardware as well as high capacity networks such as Gigabit Ethernet. In wide area networks, data compression must be utilized in order to keep interactive visualization in real time.

Video Source

SGI (Silicon Graphics) developed specialized hardware to render computer generated graphics. A typical high-resolution screen has 1280x1024 pixels and 24-bit per pixel (bpp) to represent true color (3 colors per pixel x 8 bits per color). The SGI video card outputs screen data through several means including a high-speed parallel digital data, NTSC, and RGB port. There are eight video ports, which may be programmed for output of a large variety of screen sizes and locations, in order that a selected screen section has the resolution to match a particular application. For instance, data from each quadrant of a screen can be output from four separate ports each with a resolution of 640x512 pixels. Because of the close match to NTSC (640x480), we can reduce the loss due to format conversion. The following section describes the Super Computing 98 demonstration that used this technique to offer a virtual console environment to remote users.

SC98 Remote Visualization Demonstration

A full screen demonstration of remote visualization was performed at the Super Computing 98 Conference in Orlando, FL. The hardware configuration is shown in the figure below.

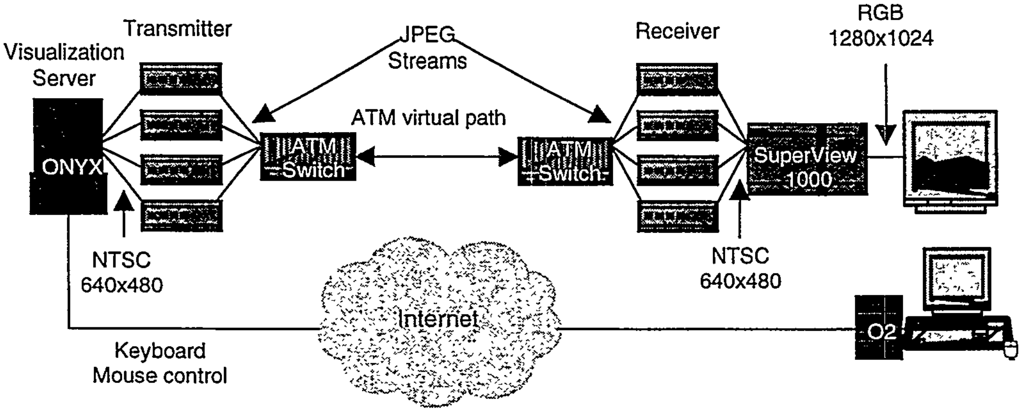

Figure 7. The SC98 demo setup

From the server end, four 640x512 RGB [27] video outputs, one from each quadrant of the screen, were converted to NTSC using four RGB Spectrum RGB/Videolink 1650 scan converters. These NTSC streams were then digitized and placed inside ATM cells for transmission using four Fore Systems JPEG encoders. At the receiving end, the NTSC streams were regenerated using four JPEG decoders. They were then combined and converted to RGB using a RGB Spectrum Superview 1000 video combiner for full resolution display. Equipment used in this demonstration was loaned from RGB Spectrum and Fore Systems. Tom Tarman, department 4616, used the external RGB video scan converter and combiner to demonstrate this capability so that we didn't have to build special hardware. However, the total cost, which remoted only a single computer screen, is prohibitively expensive (Video combiner = $17,000.00 and Video Scan Converter = $6,995.00), especially considering the number of engineers that would benefit from access to this technology.

Moreover, the quality of the resulting video suffered, due to the scan conversion from 640x512 RGB to NTSC; NTSC had effectively 240 vertical dots because it interlaces its 480 horizontal lines every other frame. Although, the video quality was usable at the receiver end, it could never be as good as the original. In addition, the JPEG compression algorithm threw away high frequency information, which blurred the edges of the text. Since we also had difficulties getting the colors of the four quadrants to match, we believe that engineers would not be able to work for an extended period of time using this screen.

Personal Computer Interface (PCI) Video Hardware

In order to remote console display in high-resolution, this project funded the design of a PC-based external device that consists of a custom PCI card and a high-speed network interface. Where the custom PCI card will be used to receive and digitize video input and the network adapter to transmit the digitized signal over high-speed networks. We will also need a Linux driver to move data directly from the video card to the network adapter, without intermediate memory copies. The bandwidth of the PCI bus is over 1 Gbps as shown below.

$$PCIBandwidth = 33MHz \times 32bits = 1.056GHz$$

Thus, it has the capacity to deliver 30 frames of full resolution video per second (~944 Mbps) to a network interface. Because of the close margin, we must take into consideration the PCI bus overhead in the development of the software driver. Linux was chosen as the operating system because of its efficiency and availability of open drivers.

Digital Video Receiver

Initially, we selected to receive high-resolution video data from the SGI Digital Video Port 2 (DVP2). Our interface to the DVP2 consists of three, 10-bit bipolar RGB words at PECL levels (PECL is Positive-ECL or Emitter Coupled Logic) and a unique biphase clock. It was designed to drive a 50-ohm cable some length to a recovery board. As the design of the digital video receiver card was completed, we were informed that the DVP2 card is no longer available. We, therefore, turned our attention to design a card that would source video signal from the NTSC port.

NTSC Design

This card was designed to have four NTSC inputs, one from each quadrant of the computer screen. These four input streams are combined and sent over the network to another PC where the video streams will be separated to regenerate four NTSC outputs. Thus, this approach will require an external video combiner and scan converter to recover the 1280 x 1024 RGB signals for display. Alternatively, a software driver running in the receiving PC could take the data from the network card, convert it to RGB and move it to a high resolution video card. As demonstrated at SC98, the initial NTSC scan conversion renders the quality of the display unsuitable for the remote visualization application. Which led us to the RGB design as described below.

RGB Design

In order to eliminate the loss due to scan conversion, we decided to input the RGB signals directly. This design digitizes RGB signals using a three-in-one video Analog-to-Digital Converter (ADC). This same card was designed with a video Digital-to-Analog Converter (DAC) in order to provide also the function to regenerate the RGB screen for use at the receiver end. The advantage of this design is its use of the RGB input. Since all computers implement RGB, this design can be used for not only remote visualization and PC screen sharing, but also inexpensive, high-quality, desktop video conferencing. Additionally, we intend to add Wavelet compression to our design, whereby we can choose to turn on or off as well as to adjust the compression scale to match a desirable data rate. Wavelet compression, as described in the next section, is considered the front runner of compression algorithms. It is being recommended to become the next generation JPEG standard (JPEG2000). A schematic of this design is filed under R56581.

Wavelet Compression Demo

Compression of video signals comes in two general forms, Wavelet and JPEG. Both techniques can be used to compress a digital NTSC video stream by a factor of 10 – 20 without appreciable loss in quality. In this realm the compression is often referred to as near lossless. It is not lossless in that repeated compression/decompression cycles on the same data set would result in lower and lower quality images. Most integrated circuit (IC) implementations of these algorithms use NTSC input due to its large commercial market. However, as the HDTV market grows, it is expected that, in the foreseeable future, these algorithms will also be integrated in re-programmable logic to compress RGB.

Analog Devices developed a series of integrated circuits that implement Wavelet compression for NTSC video. We acquired, from them, a demo board (Figure 8) that contains their ADV611 video compression chip and used it to view sample remote visualization data. The output was compared to JPEG compressed video and was shown to be of higher quality at very low data rates and of equal quality at higher data rates.

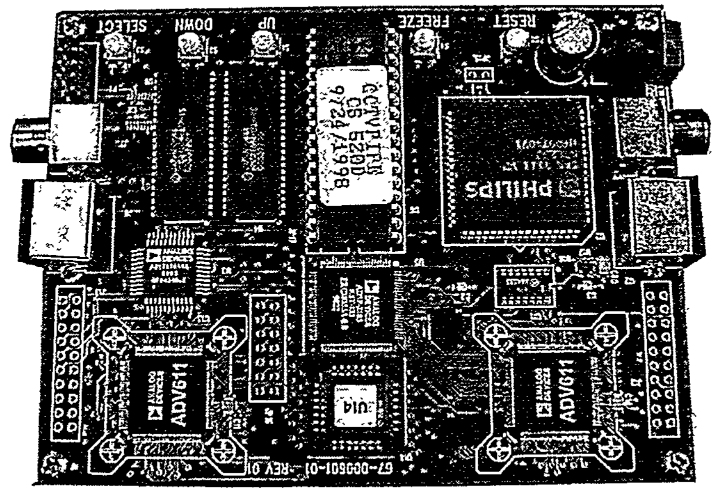

Figure 8. The CCTV Videopipe

5. Traffic Modeling Methodology

This section presents our traffic characterization and modeling research to address the networking issues raised by the ASCI and SSI applications. In particular, their visualization requirements where the extrapolated data-rate is approximately 100 gigabits per second (Gbps). We approach this problem by first understanding the traffic characteristics, such that resource requirements can be accurately assessed, and the quality of service (QoS) ensured against competing streams. We construct realistic traffic models based on empirical data in order to capture the application's true nature. By embedding statistic probes inside these models, we create benchmark tools that can be used to evaluate the network performance in terms of throughput, delay, jitter, and loss. These modeling tools can be used to drive and benchmark the quality of service (QoS) performance in simulation as well as physical networks.

Previous studies [28, 29] indicate that simple analytical expressions based on Poisson processes and maximal rate streams do not accurately model the bursty, self-similar nature of network traffic. Therefore, we propose to model the traffic of our scientific visualization applications based on detailed knowledge of the application and certain probability distributions based on measurements. The following paragraphs describe our measurement-based modeling methodology that we will employ to simulate their traffic.

We analyzed the trace of a visualization session both in client/server and in remote-x mode, and made the following observations. The bi-directional trace of a client/server-based session (Figure 9) consists of interleaved periods of interactive and bulk transfers. Where the interactive traffic represents the control messages exchanged between the client and the server for data manipulation, and the bulk-transfer is the geometry data to be rendered and displayed at the client. Thus, its traffic characteristics can be described using the measured distributions of the following variants:

1) The number-distribution of interactive periods of a typical session.
2) The number-distribution of interactive packets per interactive period.
3) The size-distribution of the interactive packets
4) The inter-arrival time-distribution of the interactive packets, which represents the processing overhead at the server and the processing and/or human's think-time at the client.
5) The number-distribution of bulk-transfers of a typical run.
6) The size-distribution of each bulk-transfer.

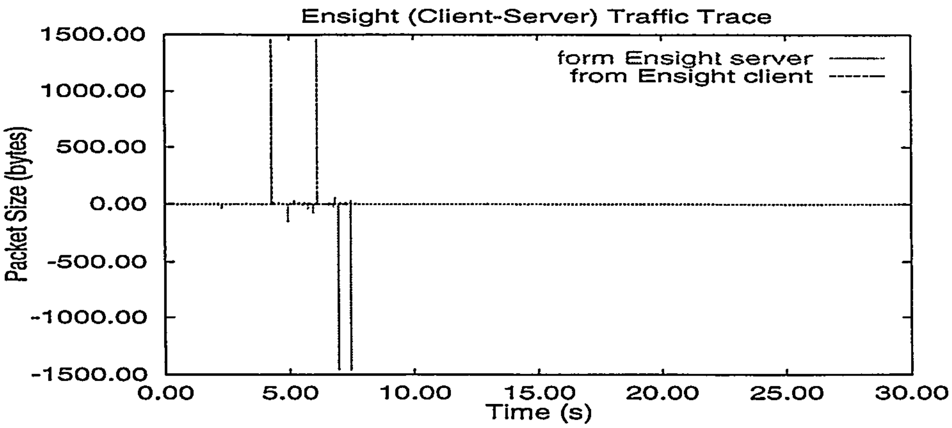

Figure 9. Bi-directional traffic trace of a visualization session in Client/Server mode

Our traffic model generates random numbers for each of the above variants based on their probability distribution, which we extracted from our traffic trace. This model uses the inverse transform method [30] to "map a uniformly distributed 0-1 random number through the y-axis of the cumulative distribution function onto the x-axis in order to obtain a random value for the variant" (Figure 10). In our case, we build a histogram of individual data points for each variant, sum them into bin-heights to create the distribution function, and store them in arrays; the content of the array elements are the values of the cumulative distribution. In order to generate a random variant, we first generate a uniformly distributed random number between 0 and 1. We then perform a binary search to locate the index, k, of the cumulative distribution into which this random number falls. Finally we linearly interpolate between x[k] and x[k +1] to determine the value for our random variant, x.

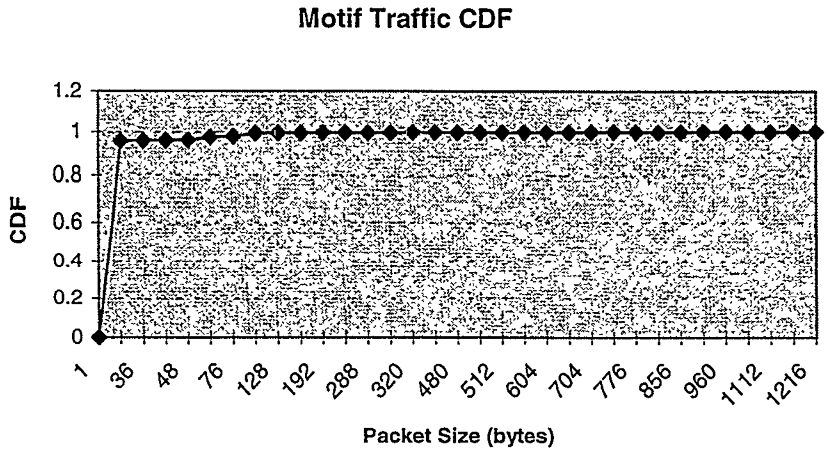

Figure 10. Cumulative Distribution Function of Packet sizes

Figure 11 depicts a bi-directional traffic trace of the same visualization session using the remote-X approach. As shown, its traffic can be described in a similar manner, because it also exhibits interleaved interactive and bulk transfers, though with a much higher demand on link bandwidth. In this case, the interactive packets represent the traffic generated by a user's mouse-clicks and keystrokes, and the bulk-transfers are the bitmap or GL-commands for the remote display. Note that following each bitmap transfer from the server, an equal amount of data is sent in the opposite direction. This traffic is the X-window backing store maintained by the visualization application; it is used to repaint the screen, in case part or the entire display is obscured.

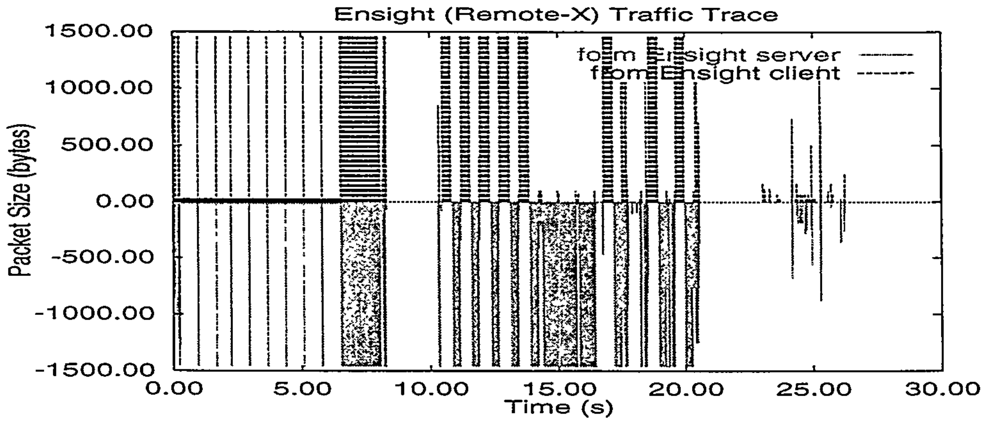

Figure 11. Bi-directional traffic trace of a visualization session in Remote-X mode

As shown in Figure 2, the remote console architecture delivers the application's control messages and graphical images in separate streams. In this case, we propose to model the control stream using the interactive variants described previously and its display traffic as real-time, continuous video streams. The packet-size distribution of the video trace (Figure 12) captures its bursty characteristic, which resulted from compression. In the absence of compression, we will model the video traffic as a continuous stream transmitted at a constant bit-rate.

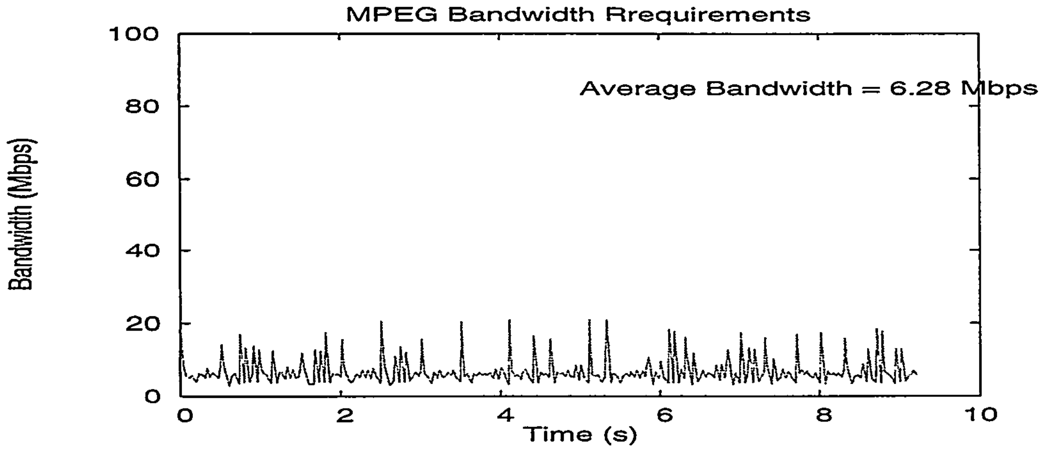

Figure 12. MPEG-2 encoded video stream

In the past year, we built several modeling and benchmark tools based on measured distributions of our scientific visualization applications. They will be useful to benchmark simulation as well as physical networks to evaluate the performance of new protocols, algorithms, and architectures.

6. Future Work

In addition to the visualization applications, we will also model and analyze popular best-effort applications used on the Internet. These models will be based on empirical data gathered from Sandia's production networks. In addition, we will derive load distributions of these applications as a function of day and time. This work can be used to supplement our existing tools in order to benchmark future ASCI and SSI systems in their entirety.

The ESnet [31] DiffServ project plans to turn on the Differentiated Service (DiffServ) capability in their cisco routers, thereby allowing network researchers to evaluate different resource allocation and fair scheduling algorithms for the Per-Hop-Behavior (PHB) of its differentiated services. This environment also offers us an opportunity to experiment with traffic classification and conditioning mechanisms at the ESnet boundary. Based on the information from our traffic modeling study, we can accurately specify a traffic profile for our application. Accurate traffic profiles enable the proper assignment of a DS codepoint such that its associated PHB, defined by ESnet, can deliver the expected quality of service. We will drive our realistic traffic model to benchmark the quality of service against the expected performance. Of course, we will verify our research results against the performance of actual interactive visualization sessions. As described earlier, our traffic models include real-time video streams as well as elastic data applications. Because of timing constraints, real-time streams prefer expedited services from ESnet. Because queuing will introduce delay and jitters, assured services drop out-of-profile packets instead. Elastic data applications, on the other hand, are sensitive to loss and, therefore, will need ESnet's assured services. Rather than dropping, the assured services re mark and queue out-of-profile packets as best-effort traffic. In addition to the traffic management study mentioned above, we plan to participate in ESnet's dynamic resource allocation experiment, using the resource-broker interface that the DiffServ project plans to develop.

The U. S. government Defense Advanced Research Projects Agency (DARPA) recently selected the National Transparent Optical Network Consortium [32] (NTONC} to develop an advanced research network as part of its Next Generation Internet (NGI) SuperNet program. The goal is to provide a testbed for complex optical networking technologies, a facility for development and demonstration of high-bandwidth network applications, and a platform for research and development of integrated network management capabilities. This testbed will be a transparent, fiber optic network running from San Diego to Seattle, with nodes in Los Angeles, San Francisco, and Portland. Additionally, there is an optically switched ring around the San Francisco Bay (Figure 8).

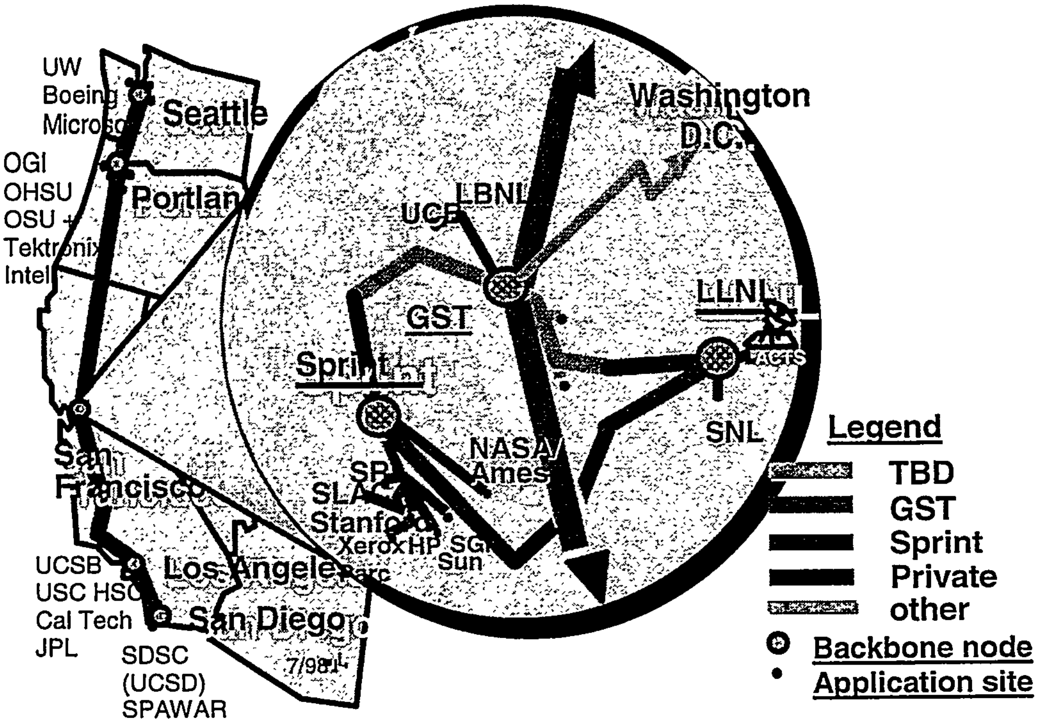

Figure 8. The National Transparent Optical Network Topology

This network uses four wavelengths on a single fiber to achieve a total capacity of 10 billion bits per second (Gbps), where 2.5 Gbps each will be allocated to support a dedicated transport mechanism as shown in Figure 9. They are ATM/SONET [33], Packet/SONET, switched SONET, and a Circuit on Demand testbed. Thus, NTON can offer us the opportunity to study effects of these transport mechanisms on QoS performance. For example, ATM schedules its load in units of small fixed-size cells, verses a weighted fair queuing algorithm employed by packet switches to minimize jitter caused by variable-length packets. Furthermore, NTON's large capacity allows us to repeat the ESnet DiffServ study using our traffic models that are scaled up to more closely approximate the sizes of future ASCI and SSI applications.

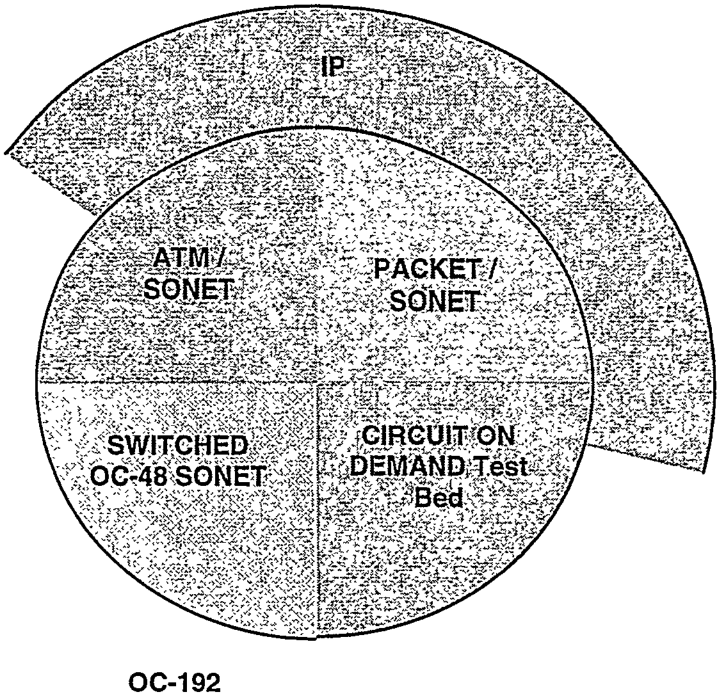

Figure 9. The 10 Gbps NTON Infrastructure

Acknowledgment

The authors would like to thank Rich Palmer, Phil Bryson, Hilary Jones, and Bruce Mah for their valuable technical as well as editorial input.

APPENDIX A

Acronyms

6BONE	A testbed sponsored by the Internet Engineering Task Force to test the Internet Protocol version 6.
ADC	Analog to Digital Converter.
API	Application Programming Interface
ASCI	Advanced Scientific Computing Initiative
ATM	Asynchronous Transfer Mode
CBQ	Class Based Queuing
CIDR	Classless Inter Domain Routing
CLIP	Classical IP over ATM protocol
DAC	Digital to Analog Converter
DiffServ	Differentiated Services
ESNET	Energy Science Network
GL	Graphical Language
HDTV	High Definition Television
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
JAVA	Machine independent, objected oriented programming language designed by Sun Microsystems
IP/IPv6	Internet Protocol/Internet Protocol version 6
JPEG	Joint Photographic Experts Group
LANE	Local Area Network Emulation
MARS	Multicast Address Resolution Server
MPEG	Motion Picture Expert Group
MPOA	Multi Protocol over ATM
NBMA	Non Broadcast Multi Access
NHRP	Next Hop Resolution Protocol
NTON	National Transparent Optical Network

NTSC	National Television System Committee
PCI	Personal Computer Interface
PNNI	Private Network-to-Network Interface
QoS	Quality of Service
RSVP	Resource reservation protocol
SONET	Synchronous Optical Network
SSI	Strategic Simulation Initiative
VLAN	Virtual Local Area Network
VPI/VCI	Virtual Path Identifier/Virtual Circuit Identifier
VRML	Virtual Reality Modeling Language
WAN	Wide Area Network
WFQ	Weighted Fair Queuing

APPENDIX B

References

[1] ISI, USC, "DOD Standard Internet Protocol", RFC 760, January 1980

[2] R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, December, 1995

[3] M. de Prycker, "Asynchronous Transfer Mode - solution for broadband ISDN", Ellis Horwood Limited, ISBN 0-13-053513-3, 1991

[4] The ATM Forum Technical Committee, "LAN Emulation over ATM Specification Version 1", ATM Specification, February 1995

[5] IEEE 802

[6] M. Laubach, "Classical IP and ARP over ATM", RFC 1577, January 1994

[7] The ATM Forum Technical Committee, "Multi-Protocol Over ATM Version 1", ATM Forum/BTD-MPOA-MPOA-01.11 Draft, February 1997

[8] The ATM Technical Committee, "ATM User Network Interface (UNI) Specification Version 3.1", Prentice Hall, 1995, pp.151-294

[9] The ATM Technical Committee, "Private Network-Network Interface Specification Version 1.0", af-pnni-0055.000, March, 1996

[10] R. Braden, L. Zhang, , S. Berson, , S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP), Version 1 Functional Specification", Internet Draft, November 1996

[11] H. Y. Chen, R. Tsang, J. M. Brandt, J. A. Hutchins, " A Survey of IP Over ATM Architectures", SAND 97-8273, July 1997

[12] http://www.ietf.org/internet-drafts/draft-ietf-ngtrans-6bone-6bone-routing-01.txt, May 1998

[13] G. Armitage,, "Support for Multicast over UNI3.0/3.1 based ATM Networks", RFC 2022, November 1996

[14] J. Luciani, D. Katz, D. Piscitello, B. Cole, "NBMA Next Hop Resolution Protocol", Internet draft, October 1996

[15] P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, G. Minshall, "Transmission of Flow Labelled IPv4 on ATM Links Ipsilon Version 1.0", RFC 1954, May 1996

[16] Y. Rekhter, B. Davie, D. Katz, E. Rosen, "Tag Switching Architecture - Overview", Internet Draft, January 1997

[17] http://www.ietf.org/internet-drafts/draft-ietf-ipngwg-bsd-api-new-05.txt, December 1998

[18] V. Fuller, T. Li, J. Yu, K. Varadhan, "Supernetting, an Address Assignment and Aggregation Strategy", RFC 1338, June 1992

[19] R. Seifert, "Gigabit Ethernet – Technology and Application for High-Speed LANs", Addison-Wesley, 1998

[20] R. Breyer, S Riley, "Switched and Fast Ethernet", Ziff-Davis Press, 1996

[21] S. Floyd, V. Jacobson, "Link-sharing and Resource Management Models for Packet Networks", IEEE/ACM Transactions on Networking, Vol.1. No. 4, August 1993, pp. 397-413.

[22] http://www.csl.sony.co.jp/person/kjc/programs.html, 1998

[23] J. C. R. Bennett, H. Zhang, "WFQ: Worst-case Fair Weighted Fair Queuing", Proceedings of INFOCOM 96, March 1996, pp. 120-128

[24] S. Floyd, V. Jacobson, "Random Early Discard for Congestion Avoidance", IEEE/ACM Transactions on Networking, Vol. 3. No.4, August 1995, pp.365-386

[25] http://www.ietf.org/internet-drafts/draft-kksjf-ecn-03.txt, April 1999

[26] S. Blake, et al., "An Architecture for Differentiated Services", RFC 2475, December 1998

[27] F. Fluckiger, "Understanding Networked Multimedia – Applications and Technology", Prentice Hall, 1995, pp. 498-502

[28] P. B. Danzig, S. Jamin, "tcplib: A Library of TCP Internetwork Traffic Characteristics", USC-CS-91-495, 1991

[29] V. Paxson, "Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic", ACM/Sigcomm, Computer Communication Review Vol. 27 No. 5, October 1997, pp. 5-18

[30] R. Jain, "The Art of Computer Systems Performance Analysis", John Wiley & Sons, 1991

[31] http://www.es.net

[32] http://www.llnl.gov/NTON

[33] W. Stalling, "Networking Standards – A Guide to OSI, ISDN, LAN, and MAN Standards", Addison-Wesley, 1993, pp. 300-306

UNLIMITED RELEASE

INITIAL DISTRIBUTION:

2	MS 0874	P. J. Robertson, 1716
1	MS 9003	D. R. Henson, 8900
1	MS 9011	P. W. Dean, 8903
1	MS 9011	P. R. Bryson, 8910/8980
2	MS 9011	J. M. Brandt, 8910
10	MS 9011	H. Y. Chen, 8910
2	MS 9214	E. D. Dart, 8910
2	MS 9011	J. A. Hutchins, 8910
1	MS 9012	S. C. Gray, 8930
1	MS 9019	B. A. Maxwell, 8940
1	MS 9011	J. Meza, 8950
1	MS 9011	M. Rogers, 8960
1	MS 9011	J. A. Larson, 8970
2	MS 9012	J. A. Friesen, 8990
1	MS 9001	T. O. Hunter, 8000 Attn: R. C. Wayne, 2200/8400 D. L. Crawford, 5200 M. E. John, 8100 L. A. West, 8200 W. J. McLean, 8300 P. N. Smith, 8500 P. E. Brewer, 8600 T. M. Dyer, 8700 L. A. Hiles, 8800
3	MS 9018	Central Technical Files, 8940-2
1	MS 0899	Technical Library, 4916
1	MS 9021	Technical Communications Dept. 8815/Technical Library, 4916
1	MS 9021	Technical Communications Dept. 8815 for DOE/OSTI
1	MS 0188	D. Chavez, LDRD Office, 4001