ELSEVIER

Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Future Generation Computer Systems 1008 (2003) 1–15

FGCS FUTURE GENERATION COMPUTER SYSTEMS

www.elsevier.com/locate/future

TeraVision: a high resolution graphics streaming device for amplified collaboration environments

Rajvikram Singh [a,1], Jason Leigh [a,*], Thomas A. DeFanti [a], Fotis Karayannis [b]

[a] *Electronic Visualization Laboratory, University of Illinois at Chicago, MC152, 1120 SEO, 851 S. Morgan Street, Chicago, IL 60620, USA*
[b] *Greek Research and Technology Network (GRNET), Greece*

Abstract

One of the common problems faced in amplified collaboration environments (ACEs), such as the Continuum, is termed the 'Display docking' or 'Display Pushing' problem where the visualization or the presentation generated on one or more computers, has to be distributed to remote sites for viewing by a group of collaborators. A typical image source in such a case could be computers ranging from laptops showing presentations, to compute clusters number crunching terabytes of data and rendering high resolution visualizations. In this paper, we present a platform independent solution which is capable of transmitting multiple high resolution video streams from such video sources to one or more destinations. The unique capability of this concept is that it is a completely hardware oriented solution, where no special software/hardware has to be installed on the source or destination machines to enable them to transmit their video. These multiple streams can either be independent of each other or they might be component streams of a video system, such as a tiled display or stereoscopic display. We shall also present results with testing on high speed dedicated long haul networks, and local area gigabit LANs with different Layer 4 protocols.

Keywords: TeraVision; Continuum; Amplified collaboration environment

1. Introduction and overview

Amplified collaboration environments (ACEs) are physical meeting spaces that enable distantly located groups to work in intensive collaboration campaigns that are augmented by advanced collaboration, computation, and visualization systems. One example of an ACE is the *Continuum* (Fig. 1) at the Electronic Visualization Laboratory [9], at the University of Illinois at Chicago. ACEs are based on the concept of the "War Room" or "Project Room" which have been shown to increase the productivity of collocated working teams by a factor of 2 [10]. The goal of the Continuum is to provide the same, if not greater, benefits for distributed teams. To this end, the Continuum integrates a broad range of technologies that include: multi-party video conferencing (via the AccessGrid [11]), electronic touch screens (for intuitive shared white-boarding), passive stereoscopic displays (such as the AGAVE, for displaying data sets in true 3D [3]), high resolution tiled displays (for displaying large visualizations or mosaics of visualizations), and PDAs and laptops for wireless control of these systems. It is anticipated that the Continuum will be a high performance front-end interface for the OptIPuter.

* Corresponding author.
E-mail addresses: rsingh@evl.uic.edu (R. Singh), spiff@evl.uic.edu, cavern@evl.uic.edu (J. Leigh).
[1] Tel.: +1-312-996-3002.

0167-739X/03/$ – see front matter © 2003 Published by Elsevier Science B.V.
doi:10.1016/S0167-739X(03)00074-8

Fig. 1. The continuum—an ACE.

The OptIPuter is a National Science Foundation funded project to interconnect distributed storage, computing and visualization resources using photonic networks. The OptIPuter project exploits the trend that network capacity is increasing, while at the same time plummeting in cost. This allows one to experiment with a new paradigm in distributed computing—where the optical networks serve as the computer's system bus; and compute clusters taken as a whole, serve as the peripherals in a potentially, planetary-scale computer. For example, a cluster of computers with high performance graphics cards would be thought of as a single giant graphics card in this context. We refer to these compute clusters as LambdaNodes to denote the fact that they are connected by multiples of light paths (often referred to as Lambdas) in an optical network. The challenge then is to optimize all the interconnected LambdaNodes to ensure that they are able to make maximal use of the network, i.e. so that the LambdaNodes are not the bottleneck in this architecture.

One can envision TeraVision as a hardware-assisted, network-enabled "Powerpoint" projector for distributing and displaying OptIPuter-based visualizations. A user who wants to give a presentation on his/her laptop, or stream output from one of the nodes of a graphics cluster simply plugs the VGA or DVI output of the source computer into the TeraVision box (called VBox for short). The box captures the signal at its native resolution, digitizes it and broadcasts it to other networked VBoxes (see Fig. 2).

Furthermore, using the VBox one can also transmit an entire tiled display provided that there are sufficient VBoxes at each end-point. Two VBoxes can be connected to the twin-heads of a stereoscopic AGAVE system to allow streaming of stereoscopic computer graphics. The VBoxes take responsibility for the synchronization for simultaneous capture of concurrent

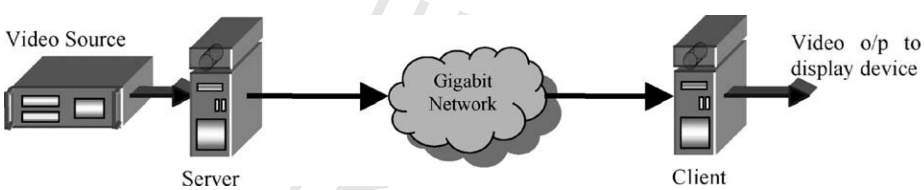


Fig. 2. Basic TeraVision setup. Note: The VBox acting as a server needs to have the video capture hardware for capturing the input video streams. The client on the other hand can be a Linux/Windows PC with a gigabit Ethernet adapter and a fast graphics card.

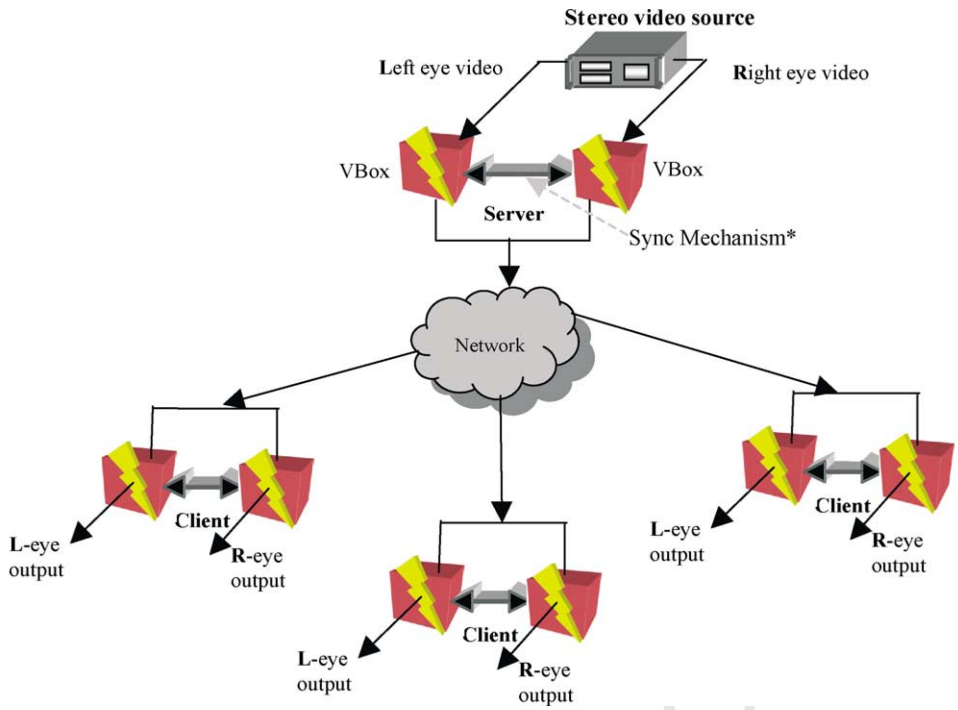


Fig. 3. TeraVision setup for streaming stereoscopic video. The sync mechanism is a dedicated, low latency channel used for synchronizing the video capture on the server side and video displays on the client side.

video streams on the server side and the synchronization for displaying the streams on the client side.

The most basic TeraVision setup (Fig. 2) consists of a server and a client connected over gigabit networks. The diagram shows a projector with every VBox to denote display capability of the unit. The server has the video capture hardware for capturing high resolution VGA or DVI inputs and the client can receive the streams and display them at various resolutions. The client can be either a Windows or a Linux PC and does not require any specialized hardware for displaying the incoming video streams. So, even though the diagram depicts the server and client to be symmetrical, they need not be. A client may only need the video capture hardware if it wants to act as a video server during a collaborative session. This will be explained in later sections.

Fig. 3 depicts a situation where two TeraVision servers are used for streaming stereoscopic video to multiple client sites. The two streams (left and right eye video) are synchronized during capture on the servers and then again on the clients before the display.

Similarly, multiple TeraVision boxes can be used for streaming the component video streams of a tiled display. Fig. 4 shows a tiled display being streamed using multiple VBoxes at a site. As in the previous case, all the servers synchronize with each other to capture the component streams. And the clients synchronize before displaying all the component streams simultaneously.

2. Hardware description

Fig. 5 shows the hardware block diagram of two VBoxes, using an Ethernet channel to synchronize the capture of two independent video streams. Many such VBoxes can be connected together and synchronized in the same fashion to capture multiple video streams.

The prototype VBoxes are Pentium 4s at 1.5 GHz with 512 MB of RAM each. The graphics cards are Radeon 8500s and the motherboard supports both 32 and 64 bit PCI slots. They have a 100 BaseT Ether-

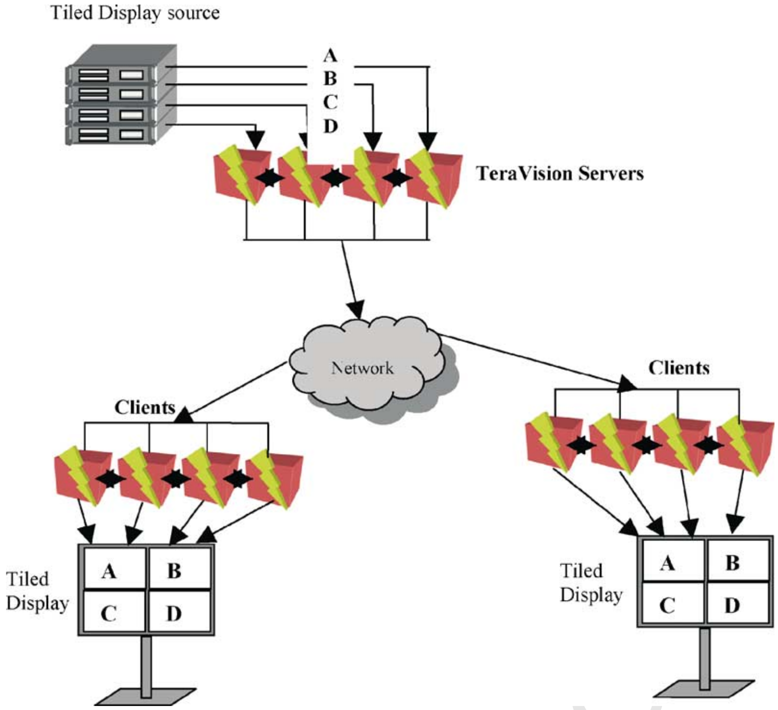


Fig. 4. Using VBoxes to stream a tiled display.

net adapter which is solely dedicated for providing the synchronization mechanism between the boxes. The sync channel needs to have low latency to be effective. Thus the network connections for the sync channels have to be either through cross-coupled cables (peer-to-peer) or through a switch carrying low network traffic.

Foresight Imaging's I-RGB-200 [6] video capture card is used for the video data acquisition. According to the specifications of this frame grabber, it is capa-

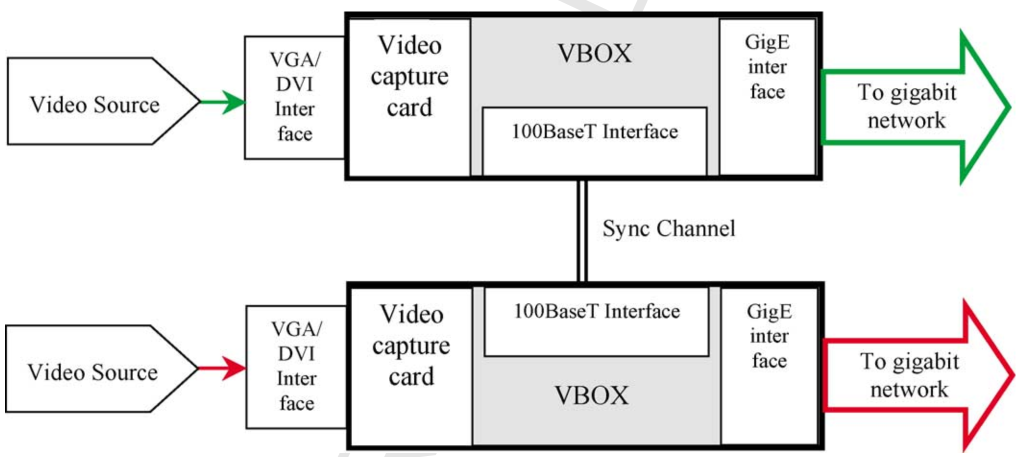


Fig. 5. Hardware block diagram. The figure shows two VBox servers synchronizing the capture through an Ethernet link. Many such VBoxes can be connected and synchronized at the same time.

ble of performing video capture at 1600 × 1200 × 75, 1280 × 1024 × 85 and 1024 × 768 × 60 Hz. The capture resolution is up to 24 bits per pixel. The card is able to sustain a 120 Mbps transfer over the PCI bus to copy captured video data to main memory. The card occupies one 32 bit, 33 MHz PCI slot on the motherboard. Only the VBoxes acting as servers need to have the capture hardware.

The gigabit Ethernet adapters used for streaming the video streams are the Intel Pro/1000 cards which use optical fiber interfaces for connecting to the network and 64 bit, 66 MHz slot for interfacing with the PC. Initially, the prototypes were tested using back to back (peer-to-peer) dedicated links and then later on long distance links between Chicago–Amsterdam, Amsterdam–Greece and Chicago–Greece.

3. Software: design and implementation

The TeraVision software was originally written for the Linux OS but later ported to Windows. The Windows version was then modified to integrate the I-RGB-200 video capture card. Currently, we have a Windows server and both Windows and Linux versions of the client.

3.1. Concepts

Server: This is a process that acts as the provider of video streams. Clients can connect to it and request TCP/UDP streams.

Client: This is the process that needs to connect to the server to get the video streams. It is also responsible for displaying the streams.

Master: A process (server or client) running as a master is responsible for providing sync messages to all slave processes connected to it. All slave processes have to wait for the sync 'pulse', before they can transmit or receive a video frame.

Slave: The slave processes are started by giving them the IP address of a master process (server or client). The slaves connect to the master and wait for sync messages before they can either transmit (in case of a server) or receive (in case of a client).

Hence for a typical TeraVision setup, there is a master server and one or more slave servers, which constitute the senders. And similarly there is a master client and one or more slave clients, which constitute the receivers. Fig. 6 further depicts the concept.

3.2. TeraVision server

The I-RGB-200 frame grabber card uses DMA to transfer the captured frames to a set of circular buffers specified by the user in the user space. Since the gigabit Ethernet adapters also use DMA for transferring large chunks of data from system memory to the LAN card's on board buffers, this becomes a serious point of contention because of the limited bandwidth of the 64 bit PCI bus. Thus the performance of the PCI bus limits the overall performance of the system.

The server software (Fig. 7) is threaded with one of the threads acting as the producer. It is responsible for filling up a common circular buffer with captured frames. Another thread acting as the consumer, tries to empty the circular buffer and transfer the data to the network as fast as the system and the network can allow it to. Frames are dropped on the fly if the network is slower than the capture rate.

Whenever the networking thread (consumer) gets the CPU, it simply picks up the latest frame in the circular buffer and pushes it out of the network. The reasoning behind this approach is that if the network is faster than the capture rate, all frames will be transmitted. However, if it is slower than the capture rate, the consumer thread will run at intervals decided by the network throughput (assuming there are no other CPU intensive tasks on the system). Thus the OS' scheduler indirectly affects the frame decimation.

The server can accept video frames either from a video capture card or disk files. The user may also choose to transmit video via TCP or UDP streams. Future versions will incorporate options for using RBUDP [1] and multicasting (over UDP). Plans for integrating a compression module are also underway.

The UDP module in TeraVision takes the video frame data and splits it up into UDP packets. It marks every UDP packet with a header, which allows the receivers to re-assemble the video frame in the correct way even if there are packet losses, duplication or out of order packets in the network or host machines. This simple 'protocol' for handling video streams on UDP has been implemented using scatter-gather techniques to minimize memory copies.

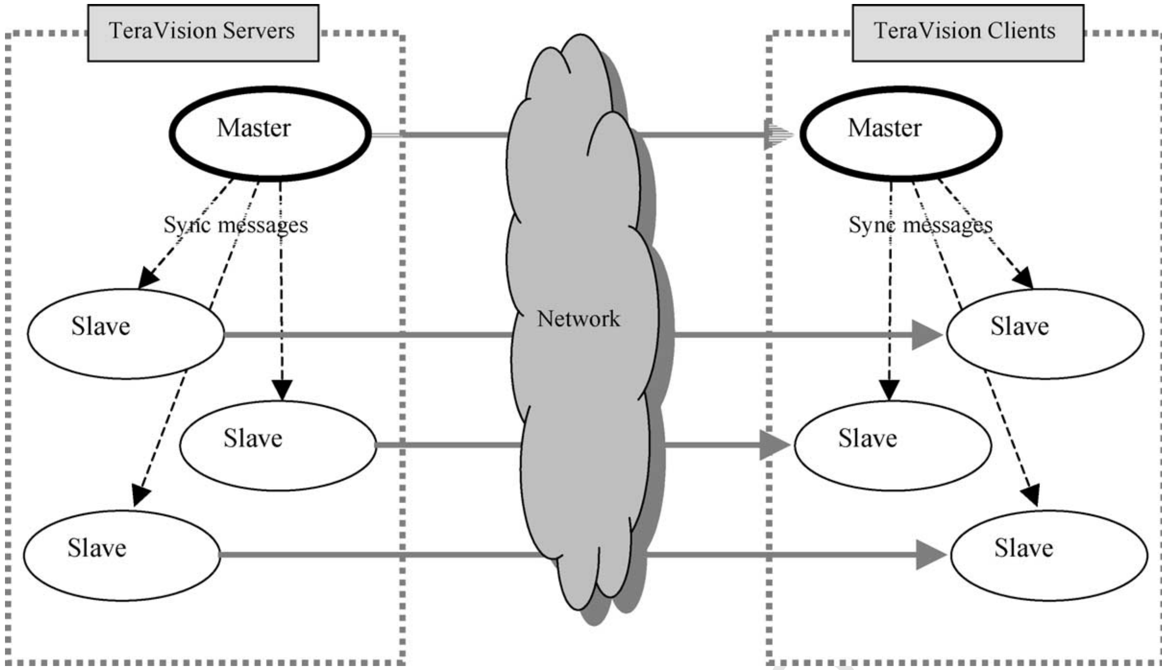


Fig. 6. Master–slave concept. The server (or client) can consist of many processes, where one process acts as a master and the rest as slaves. The master process provides the sync messages to all the slaves for synchronizing the capture (on the server side) and the display (on the client side).

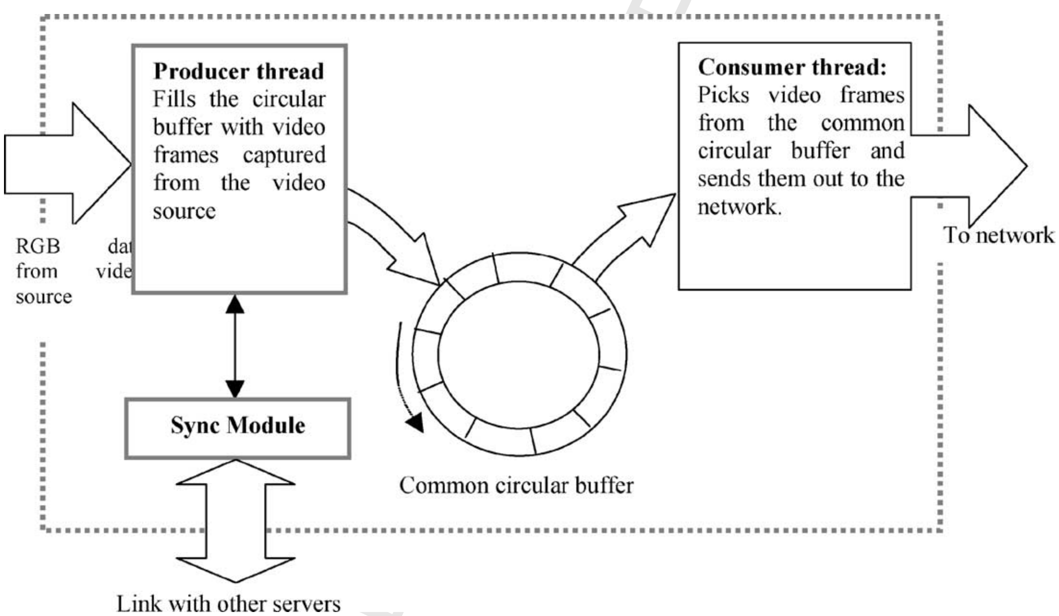


Fig. 7. TeraVision server design.

3.3. TeraVision client

The clients, at the time of writing this paper, are available for both Windows and Linux. The display ends are responsible for receiving the incoming network data and displaying them on projectors or monitors.

Since data is consistently coming in from the network, the clients' software also needs to be threaded, so that the display may run simultaneously with the network. If reliable transport protocols like TCP are used for sending the video streams, care has to be taken as to not stop the network streams as it might give the impression of network congestion to the sending machine, causing TCP to back up. And if unreliable transport layer protocols like UDP are used, again the network cannot be ignored as it may cause large packet losses due to socket buffer overflow.

Thus an ideal solution for this would be to let the networking code and the display code run as threads, independent of each other. Similar to the server's design, the client software also has two main threads running as producer–consumer with a common circular buffer. The network thread (producer) is responsible for picking up the incoming data from the network and filling the common circular buffer. The display thread (consumer) empties out this buffer and pastes the frames on the screen.

In case of the network throughput being faster than the display speed, frames are dropped from the common buffer. The master client makes this decision and then lets all the slave clients know which frames are to be finally displayed during the synchronization.

3.4. Sync module

The sync module is present on both the server and client ends. It enables the master processes to send synchronization messages to the slave processes. In the prototype boxes, the sync modules use a dedicated Ethernet adapter on the PCs to transmit the synchronization messages. A dedicated link ensures low latency for the sync messages. The software uses TCP/IP to send the messages between machines.

On the server side, the sync module is used for synchronizing multiple servers before they capture video frames. On the client side, the sync module provides a mechanism for the master to specify to its slave, which frames to display simultaneously. This is important as frames might be needed to be dropped, in case the network throughput exceeds the display speed. It also ensures that the frames are pasted on the screens simultaneously, which is extremely important for stereoscopic or tiled display streams.

One can run the servers and clients with synchronization or without. It was noticed in the prototypes that switching on the synchronization, decreased the server throughput as now critical CPU time was used for sending and waiting for sync messages using blocking I/O calls.

4. Tests and observations

Tests were run for both TCP and UDP streams and the results are shown below. We experimented with various socket and TCP flow window sizes. The TCP flow Windows were calculated based on the round trip times. UDP packet sizes were also varied. For all experiments the Ethernet cards, intermediate routers and switches were configured to use the standard 1500 byte MTUs. The tests were done initially for a LAN setup, which provided near ideal network conditions as there are minimal packet losses and very low transport delays. Thus they helped in identifying the upper performance limit of the systems in terms of throughput and frames per second.

The second set of tests were done over LFNs (long-fat networks). These networks provide a very different scenario as there are packet losses and long round trip delays, which affect the performance of acknowledgment-based reliable transport protocols such as TCP. One has to either manually tune the TCP stacks or rely on some sort of auto-tuning provided by the OS to get good performance.

The video sources in all the experiments were PCs running MS Windows. The display of the PCs were set to run at 1024×768 at 60 Hz. The pixel depth was 32 bits per pixel and the both VGA and DVI outputs of the sources were tested with the TeraVision hardware.

4.1. Gigabit LAN tests

The prototype TeraVision boxes were tested on two types of LAN configurations:

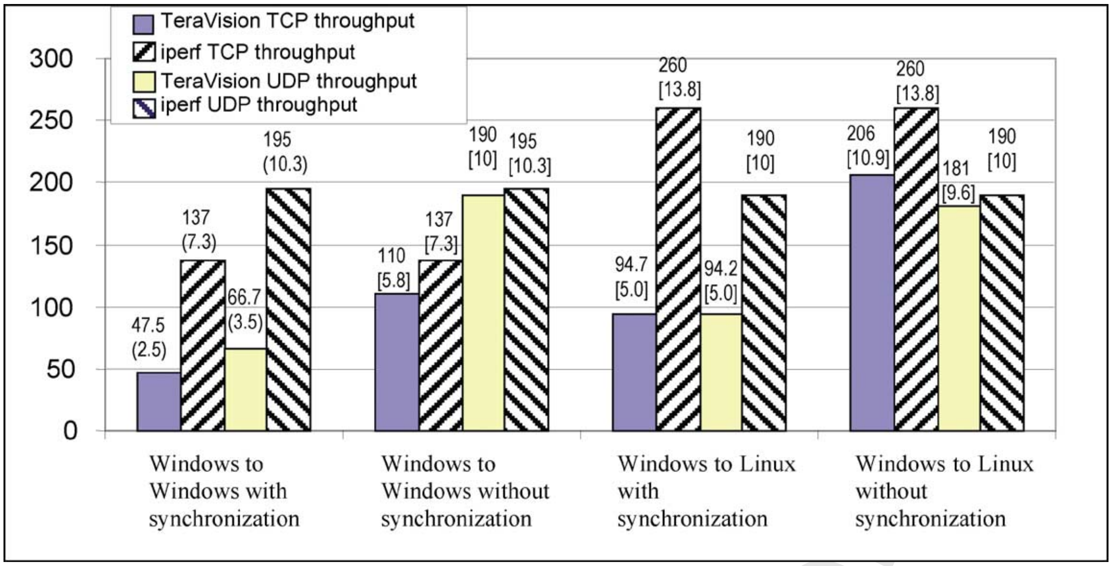


Fig. 8. TeraVision throughput with TCP and UDP streams on gigabit LAN. The effective frames per second are indicated in square brackets along with the observed throughput in Mbps. The UDP tests were done with 1000 byte packets. The observed loss was 0%.

- Back to back/peer-to-peer mode, in which the servers and clients are connected to each other directly using cross-connect cables.
- Through a network switch, where the VBoxes had to share the medium with Ethernet traffic from other machines.

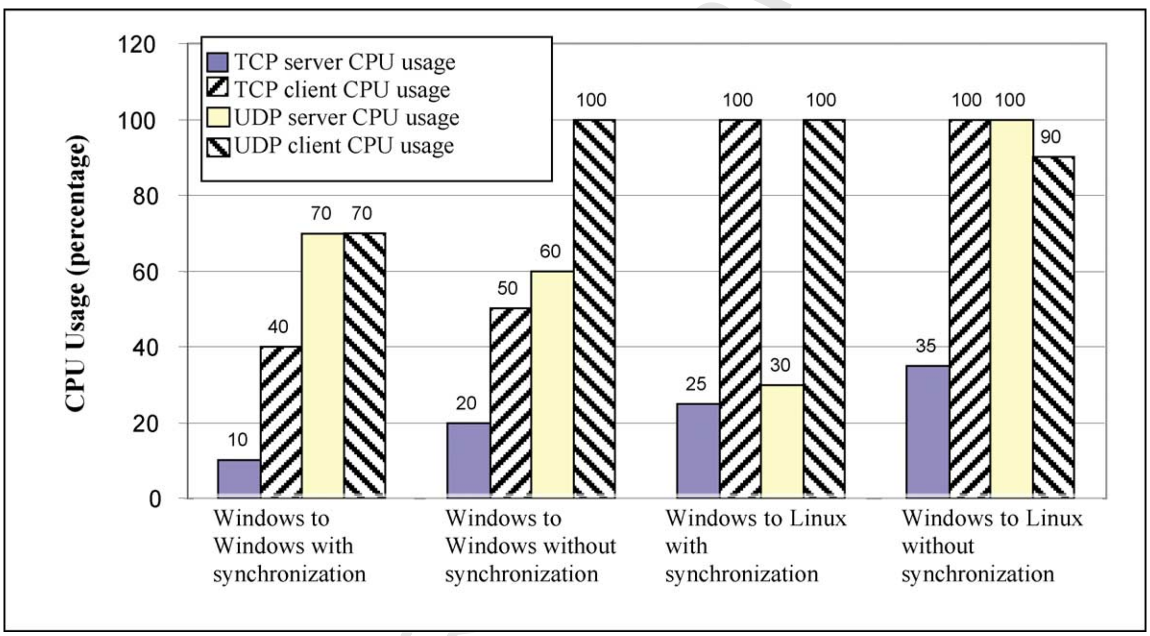


Fig. 9. CPU usage on the TeraVision servers and clients. The UDP tests were done with 1000 byte packets. The 0% loss was observed for UDP.

Since the machines were placed so close to each other, the TCP flow control window does not affect performance significantly. As shown in Figs. 8 and 9 the throughput achieved by TCP streams was close to the ones attained by UDP. The effective frames per second are indicated within square brackets along with the observed throughput (in Mbps) in Fig. 8. We also noticed that the Linux OS is more efficient in receiving incoming network traffic (Fig. 8). Fig. 9 shows the CPU utilization of the VBoxes for TCP and UDP streams. We notice that CPU usage is higher for Linux, indicating that high priority is given to the network sub-system in the OS.

The 1000 byte UDP packets were used for all the tests as they seem to give the best throughput for Windows. In all the LAN experiments, no significant packet losses were observed. The streams always show high packet losses when they are started but the losses diminish almost immediately, as the operating systems adjust internal buffers to minimize the loss.

4.2. Over LFNs

During iGrid 2002, a TeraVision experiment was setup where video was streamed between Amsterdam and Chicago and also between Greece and Amsterdam. Subsequent experiments were performed between Greece (GRNET) and Chicago (EVL). The following graphs show the data that was collected for tests done between GRNET (Greece) and EVL (Chicago) (Fig. 10).

The sending machines in this set of experiments used the Windows XP operating system and thus we notice the UDP throughput in these tests is considerably higher than the previous tests on gigabit LAN (Fig. 8) where the sending machines were running Windows 2000. However, we notice that the TCP throughput has decreased considerably (Fig. 11). The TCP stacks on the machines at both ends were tuned for long-fat networks. The TCP flow Windows were adjusted to the bandwidth-delay product of the network. Ideally, if the TCP flow Windows are set to the bandwidth-delay product, the line utilization should be 100% and TCP should perform as well as UDP. However, the performance of TCP streams is extremely poor on LFNs, as we can see from the graphs.

Fig. 11 shows the throughput achieved by TCP and UDP streams over the LFN. The UDP streams showed 0% loss in all the tests. Since the main difference for TCP packets is that the sending machine has to wait for the acknowledgments after sending data equal to the flow window size, we believe that it is the acknowledgements that hurt the performance of TCP streams. The buffers on the intermediate network nodes (routers) seem to queue the acknowledgment packets, slowing down the throughput of the

```
1   <1 ms  <1 ms  <1 ms  195.251.26.230
2   <1 ms  <1 ms  <1 ms  koletti-acropolis-PoS.athensMAN.grnet2.gr
[195.251.24.234]
3   <1 ms  <1 ms  <1 ms  grnet.gr1.gr.geant.net [62.40.103.57]
4   62 ms  62 ms  62 ms  gr.uk1.uk.geant.net [62.40.96.98]
5   69 ms  69 ms  69 ms  uk.fr1.fr.geant.net [62.40.96.89]
6   78 ms  78 ms  77 ms  fr.de1.de.geant.net [62.40.96.49]
7   78 ms  78 ms  78 ms  de1-1.de2.de.geant.net [62.40.96.130]
8  167 ms 167 ms 167 ms  abilene-gtren-gw.de2.de.geant.net [62.40.103.254]
9  171 ms 171 ms 171 ms  wash-nycm.abilene.ucaid.edu [198.32.8.45]
10 171 ms 180 ms 171 ms  198.32.11.126
11 184 ms 175 ms 175 ms  nycmng-washng.abilene.ucaid.edu [198.32.8.84]
12 195 ms 195 ms 195 ms  chinng-nycmng.abilene.ucaid.edu [198.32.8.82]
13 195 ms 195 ms 195 ms  chin-chinng.abilene.ucaid.edu [198.32.11.109]
14 291 ms 212 ms 196 ms  mren-chin-ge.abilene.ucaid.edu [198.32.11.98]
15 196 ms 196 ms 196 ms  131.193.80.78
```

Fig. 10. Traceroute from GRNET to EVL. The routes are symmetrical in both directions.

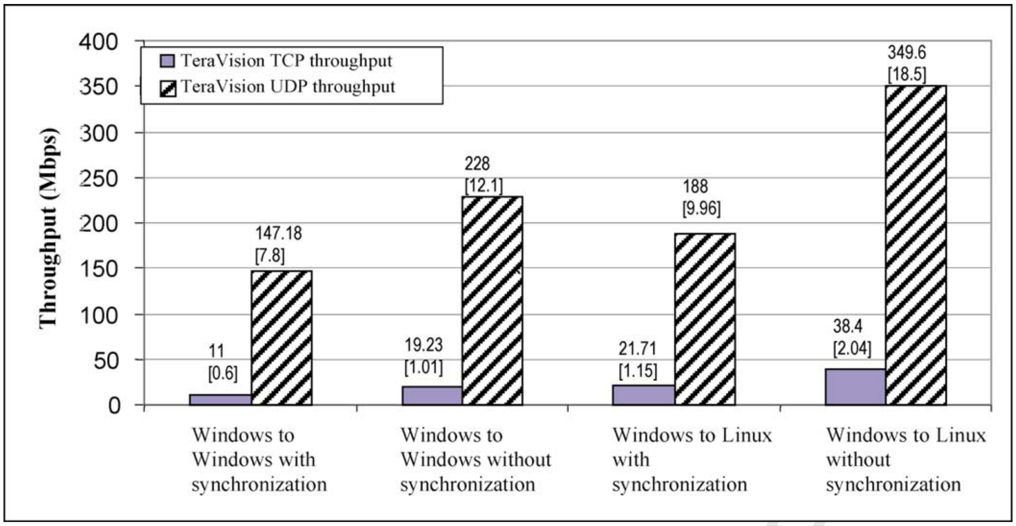


Fig. 11. TeraVision throughput on UDP and TCP streams between GRNET and EVL. The effective frames per second are indicated in square brackets along with the observed throughput in Mbps. The UDP tests were done with 1000 byte packets. The 0% loss was observed for UDP.

TCP streams. But since there is 0% packet loss for UDP, a selective acknowledgment scheme would be more suitable for reliable transmission on such networks. The future versions of TeraVision will incorporate RBUDP [1], which uses SACK (Selective Acknowledgement) packets for enabling reliable transfer.

4.2.1. Between Greece and Chicago (GRNET and EVL)

The tests over the LFNs were done at iGrid 2002 and between GRNET and EVL. The following traceroute and graphs show the results for the EVL–GRNET tests.

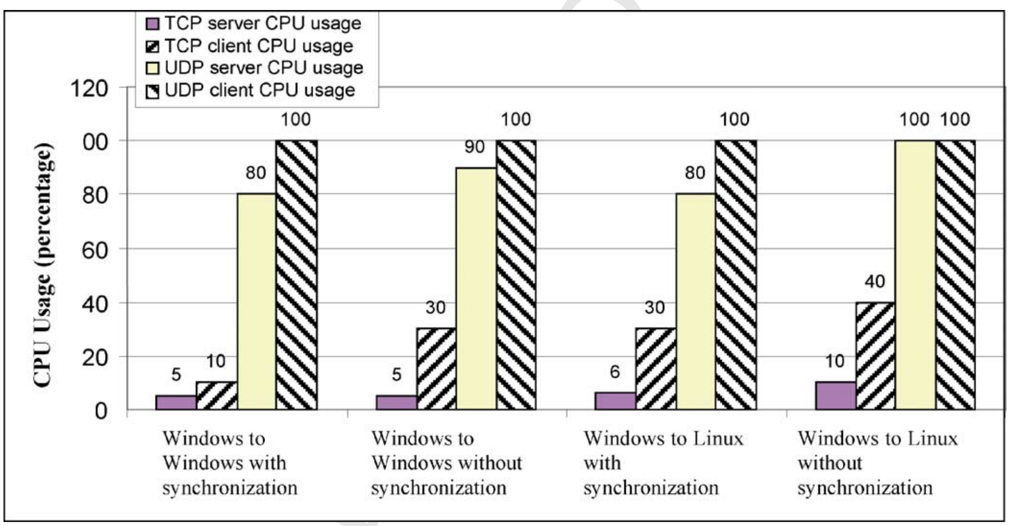
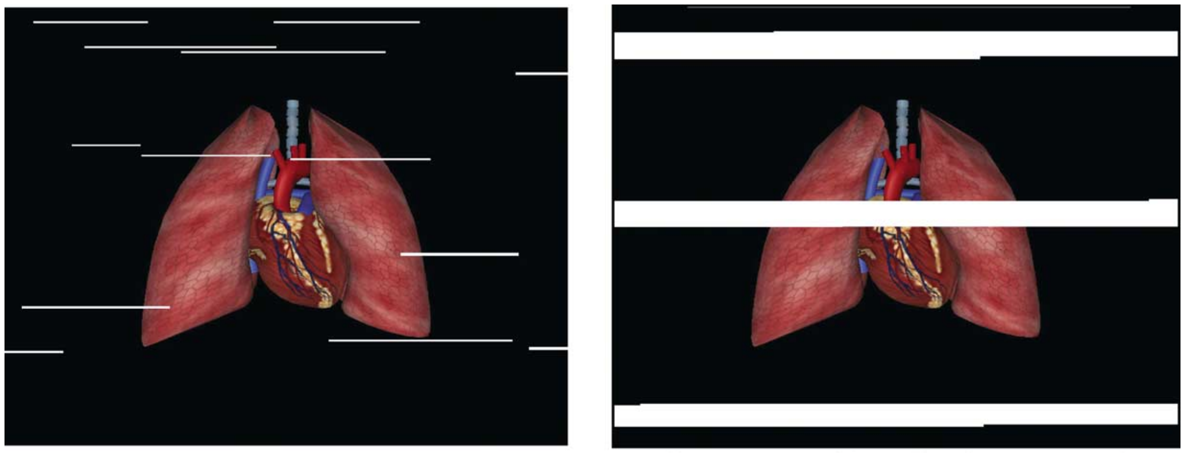


Fig. 12. CPU usage with UDP and TCP streams between GRNET and EVL observations.



Client output with network losses

Client output with packet losses on end hosts.

Fig. 13. Images of a client showing the result of network loss vs. packet loss at the end hosts.

From our tests, the UDP streams seem to be the most apt method for streaming large data over LFNs. However, because of the packet losses, typical of UDP streams, the resultant image has missing pixels which cause undesirable streaks across the image (Fig. 12). However, it was noticed that there was a certain pattern in the manner in which the network, i.e. the routers, lost data and the way the end hosts lost the data. The white streaks represent lost data packets. When the network loses the packets, the packets dropped by the routers are random and intermittent. Since each UDP packet typically is between 500 and 1500 bytes, the resultant image has small streaks which appear at random positions on the screen (Fig. 13).

However, when the end hosts lose the data, it is generally due to buffer overflows, either in the OS or the driver. Thus it causes large contiguous chunks to be missing from the resultant image (Fig. 13). The losses also do not seem to be random and occur at regular intervals. Such losses are observed immediately at the beginning of a session, just when the streaming is started. The OS then adjusts its buffer sizes to minimize the loss and image smoothens out after a few seconds.

5. Limitations imposed by the hardware

The original goal was to be able to achieve a network streaming rate of 30 fps per VBox, but even under the best conditions, we have been able to touch ~15 fps. The reason for this can be explained as follows. We shall assume ideal conditions and take technical specifications as given by the hardware manufacturers.

The main point of contention in the hardware is the PCI bus. Since there is a single bus that is shared both by the capture card and the network adapter, the performance of the bus decides the performance of the system. The PCI bus on the motherboards of the PCs support 32 bit, 33 MHz PCI slots for the video capture card and 64 bit, 66 MHz PCI for the gigabit Ethernet adapters.

Let us assume that the video card is capturing 1024×768 at 24 bpp, frames at 15 fps, which amounts to 35.4 MB of data. At the specified transfer rate of 120 Mbps, it would take ~0.3 s to DMA all the data from the card's onboard buffers to the PC's main memory. This data then has to be broken into UDP or TCP packets with appropriate computations. Assuming one memory copy by the protocol stack in the OS, and a 400 MHz FSB on the PCs, it would take approximately ~0.05 s for the memory copy. Then the data has to be sent out to the gigabit Ethernet adapter. But even though the gigabit LAN adapter interfaces through a 64 bit, 66 MHz bus, it can only consume data at 1 Gbps (or 125 Mbps), which is the specified network throughput. Thus, even if the card DMAs all the data from the kernel space to the onboard buffers, it will take ∼0.3 s.

Thus the aggregate time taken for streaming a 30 fps stream is $0.3+0.05+0.3 = 0.65$ s. Thus, even though it seems theoretically possible to stream 15 frames under a second, this figure is for near ideal conditions, where we have not taken into account factors like other devices sharing the bus (like a 4X AGP video adapter) and the OS/software overheads. The CPU usage touches 100% at 15 fps on the server side, further proving that the system is out of computing resources to do anything better.

6. Future work

6.1. Replace CAVERNsoft with QUANTA

The present version of TeraVision uses CAVERNsoft [8] for providing all the networking APIs. Future versions of the software will use the QUANTA [2] toolkit (the successor to CAVERNsoft). QUANTA is a networking middleware being developed at EVL and provides scientific applications with a high-level way to specify their data delivery requirements (such as bandwidth, latency, jitter, reliability). It then transparently translates them into the appropriate transmission protocol and network QoS services to achieve the optimum performance. QUANTA consists of a collection of novel networking protocols designed to handle a wide variety of extremely high bandwidth application traffic flows. One such protocol is Reliable Blast UDP (RBUDP).

6.2. Incorporate RBUDP

TeraVision is intended to be a graphics streaming device for scientific visualization applications. And typical scientific visualizations cannot tolerate artifacts in the resultant image. Thus UDP is far from being an ideal solution for TeraVision. We need a reliable transport layer, which can provide the performance of UDP but with the reliability of TCP. EVL has been working on such a streaming protocol called the RBUDP [1]. RBUDP uses a scheme of selective acknowledgments, where the sender sends a burst of UDP packets and the receiver acknowledges only the packets which are not received. The sender then re-transmits the missing packets. RBUDP has shown excellent results for LFNs and the performance is close to UDP streams. Thus future versions of TeraVision will incorporate RBUDP as an alternative transport layer protocol for streaming.

6.3. Real-time compression

Work is underway to integrate a compression module in the server and client code. We are working on an optimized version of RLE (run length encoding) compression which can make use of the SIMD (single instruction multiple data) instructions on the CPU to compress (and de-compress) the captured frames in real-time. The idea is to shift the load from the PCI bus to the CPU. By reducing the amount of data being sent and received on the PCI bus, we hope to increase the frames per second being streamed. Threaded code ensures efficient utilization of multiple CPUs.

6.4. Multicasting

The prototype boxes can only transmit point to point, using UDP or TCP. To distribute the video stream to multiple sources simultaneously (as in the case of collaborative use scenarios), multicast must be employed. However, multicast, like UDP, is an unreliable protocol. The protocol that we know that holds the most promise is RBUDP, however, RBUDP is a point to point protocol. At the data rates generated by the TeraVision boxes, *broadcast* RBUDP is impractical as a single TeraVision box does not have the capacity to serve more than one end-point. We believe that a combination of Forward Error Corrected Multicast and light-weight real-time compression might hold the solution.

6.5. Tighter synchronization

In one set of our experiments, we streamed stereoscopic animation using two servers and two clients. In this setup one stream carries the left eye information and the other carries the right eye information. The two streams have to be tightly synchronized together. If the streams are off by even a few milliseconds, there is a noticeable glitch in the resultant 3D video. There was a glitch visible in the video, which indicated that the synchronization was not close enough.

The prototype software uses blocking TCP/IP calls for sending synchronization pulses between the processes. The video streams are synchronized when they are captured and then synchronized again before they are pasted on the display. The servers and clients can be run either with or without the synchronization switched on. When the synchronization is switched off, the two streams run independent of each other and the resultant video is out of sync. When the synchronization on the servers and clients is switched on, the video appears to be better, but the frame rate drops drastically (Fig. 11).

Since the systems are heavily loaded, the OS scheduling and queuing greatly affects the transfer of the sync messages. We plan to experiment with raw network data packets and OOB (out of band) data to tighten the synchronization. Another option is to make the synchronization run at real-time priority and switch off all possible queuing in the TCP/IP stack and the Ethernet driver and hardware.

6.6. *Floor control*

Ideally, a VBox should be able to act as a server and a client. The future versions of TeraVision would let many clients connect to a server and receive data. But if a client wants to then act as a server, he/she can ask for a floor control lock. Essentially, the server is sent a message, requesting it to release the lock to the client. The user on the server may then decide to honor or ignore the request.

In case the user honors the request, the server process shuts down the transmission and starts up a client process. All the clients then continue to receive the data from the 'new' server. We have already implemented code for a distributed mutex, which can be used as the floor control lock during collaboration.

7. Recommendations

7.1. *PC architecture*

The PC architecture seems to be inherently limited for real-time streaming applications such as TeraVision. The CPU, I/O devices and memory share the same bus, causing bottlenecks. One solution would be to provide multiple data paths between the various components on the motherboard. Some of the upcoming technologies such as Infiniband [5] promise to let computer architectures have such a design (Fig. 14).

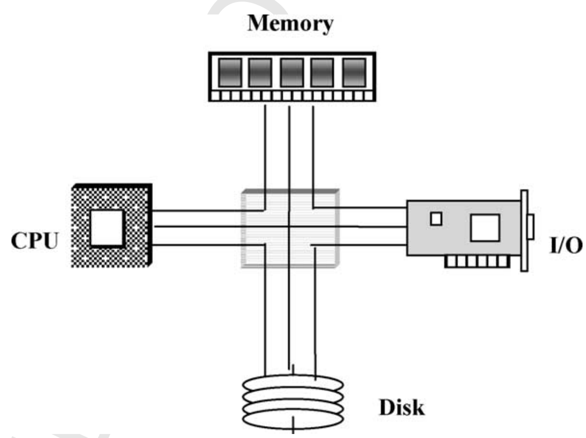


Fig. 14. Multiple data paths between the various components in a PC would ensure better performance.

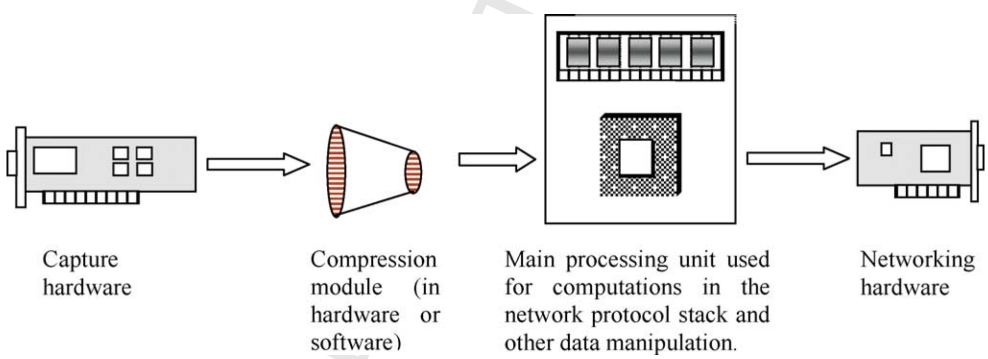


Fig. 15. A pipelined architecture would ensure that the data paths between modules are independent of each other.

The other option is to have a pipelined architecture between the peripheral hardware, where data is sent from one module to the other over dedicated channels and there is no contention. As shown in Fig. 15, the data paths between the capture hardware, compression module, networking module are dedicated and independent of each other.

8. Conclusion

TeraVision is a graphics streaming system, which is capable of streaming multiple synchronized video streams over high speed networks. It currently uses TCP and UDP for sending the network data. Currently for LFNs, TCP fails to give acceptable performance whereas UDP provides performance at 15 fps when there is sufficient bandwidth to deliver the image frames. Future versions of TeraVision will incorporate RBUDP, compression and multicasting options. Eventually, the entire networking layer in TeraVision will be replaced by QUANTA [2].

TeraVision prototypes were demonstrated successfully during iGrid 2002 [7]. For the purpose of testing, TeraVision boxes have been installed in Greece and the New Media Innovation Center in British Columbia, Canada, are building there own TeraVision boxes. Argonne National Labs will also soon have one to help stream high resolution graphics for weather simulations.

Uncited reference

[4].

Acknowledgements

We would like to thank Cees de Laat at University of Amsterdam for providing the end-points at SARA in Amsterdam to perform experiments during the development of TeraVision. Alan Verlo, Linda Winkler, Lance Long, Paul Wielinga and Hans Blom provided us with diligent support to help debug and solve various network issues. Also the valuable inputs by Javier Girado, Yong-Joo Cho, Andrew Johnson, Naveen Krishnaprasad, Luc Renambot, Vikas Chowdhry, Shalini Venkataraman, Atul Nayak and Greg Dawe are greatly appreciated. We are very grateful to Maxine Brown and Laura Wolf for their time and effort in helping to organize iGrid 2002 and the TeraVision demo at iGrid 2002. We also thank Chris Scharver and Helen Kostis for their help with testing and demonstrating TeraVision during iGrid 2002.

The virtual reality and advanced networking research, collaborations, and outreach programs at the Electronic Visualization Laboratory (EVL) at the University of Illinois at Chicago are made possible by major funding from the National Science Foundation (NSF), awards EIA-9802090, EIA-0115809, ANI-9980480, ANI-0229642, ANI-9730202, ANI-0123399, ANI-0129527 and EAR-0218918, as well as the NSF Information Technology Research (ITR) cooperative agreement (ANI-0225642) to the University of California at San Diego (UCSD) for “The OptIPuter” and the NSF Partnerships for Advanced Computational Infrastructure (PACI) cooperative agreement (ACI-9619019) to the National Computational Science Alliance. EVL also receives funding from the US Department of Energy (DOE) ASCI VIEWS program. In addition, EVL receives funding from the State of Illinois, Microsoft Research, General Motors Research, and Pacific Interface on behalf of NTT Optical Network Systems Laboratory in Japan.

References

[1] E. He, J. Leigh, O. Yu, T.A. DeFanti, Reliable blast UDP: predictable high performance bulk data transfer, in: Proceedings of the IEEE Cluster Computing 2002, Chicago, IL, September 2002.

[2] E. He, J. Alimohideen, N.K. Krishnaprasad, J. Leigh, O. Yu, T.A. DeFanti, L. Smarr, QUANTA: A Toolkit for High Performance Networking Applications.

[3] J. Leigh, G. Dawe, J. Talandis, E. He, S. Venkataraman, J. Ge, D. Sandin, T.A. DeFanti, AGAVE: access grid augmented virtual environment, in: Proceedings of the AccessGrid Retreat, Argonne, IL, January 2001.

[4] W.R. Stevens, Unix Networking Programming, Networking APIs: Sockets and XTI, vol. 1, 2nd ed., Addison-Wesley, Reading, MA, 1998, 357 pp.
[5] InfiniBand Trade Association. http://www.infinibandta.org.
[6] ForeSight Imaging. http://www.foresightimaging.com.
[7] iGrid 2002. http://www.igrid2002.org.
[8] K. Park, Y. Cho, N. Krishnaprasad, C. Scharver, M. Lewis, J. Leigh, A. Johnson, CAVERNsoft G2: a toolkit for high performance tele-immersive collaboration, in: Proceedings of the ACM Symposium on Virtual Reality Software and Technology 2000, Seoul, Korea, October 22–25, 2000, pp. 8–15, in press.
[9] The Continuum Project. http://www.evl.uic.edu/cavern/continuum.
[10] J.S. Olson, L. Covi, E. Rocco, W.J. Miller, P. Allie, A room of your own: what would it take to help remote groups work as well as collocated groups? in: Short Paper of the Conference on Human Factors in Computing Systems, CHI'98, 1998, pp. 279–280.
[11] The Access Grid. http://www.accessgrid.org.

Rajvikram Singh is a Master of Science student, majoring in computer science, at the University of Illinois at Chicago. He is also a Research Assistant at the Electronic Visualization Laboratory and works with Jason Leigh's group. His current research interests include developing applications and protocols for high-speed optical networks, concurrent computing, system programming and real-time systems. He can be contacted at: Electronic Visualization Laboratory, Engineering Research Facility (ERF), University of Illinois at Chicago, Chicago, IL.

Jason Leigh is an Associate Professor in the Department of Computer Science at the University of Illinois at Chicago; and a Senior Scientist at the Electronic Visualization Laboratory (EVL). Leigh is a Co-chair of the Global Grid Forum's Advanced Collaborative Environments research group; and a co-founder of the GeoWall Consortium.

Fotis Karayannis received his Dipl-Ing. degree from the Department of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Greece, in 1994. He completed his PhD in January 1999 in the Computer Science Division of the same university in the fields of Integrated Communications and Management of Broadband Networks. He has been a Research Associate at the NTUA Telecommunications Laboratory from 1994 to 1999 participating in different European communications projects. From 1999 to 2000, he has been working for the Greek Research and Technology Network (GRNET) as the Technical Planning and Development Manager dealing with the development of the Next Generation Internet network and Grid services for the Greek Academic and Research Institutes. He is the technical coordinator of the HellasGrid national Grid task force.

Thomas A. DeFanti is a Director of the Electronic Visualization Laboratory (EVL), a distinguished Professor in the Department of Computer Science, and Director of the Software Technologies Research Center at the University of Illinois at Chicago.