# End-to-End Performance Measurement of Internet Based Medical Applications

**P Dev, PhD[1], D Harris, PhD[2], D Gutierrez, MS[2], A Shah, MS[2], S Senger, PhD[3]**
**[1]SUMMIT, Stanford University, Stanford, California, USA and**
**[2]Dept. of Electrical Engineering, Stanford University, Stanford, California, USA and**
**[3]Dept. of Computer Science, University of Wisconsin, La Crosse, Wisconsin, USA**

## Abstract

*We present a method to obtain an end-to-end characterization of the performance of an application over a network. This method is not dependent on any specific application or type of network. The method requires characterization of network parameters, such as latency and packet loss, between the expected server or client endpoints, as well as characterization of the application's constraints on these parameters. A subjective metric is presented that integrates these characterizations and that operates over a wide range of applications and networks. We believe that this method may be of wide applicability as research and educational applications increasingly make use of computation and data servers that are distributed over the Internet.*

## Introduction

An Internet based application is one in which the user interacts with a program on one computer and, through it, reaches out over the Internet to a program or data on another computer. Examples are a Web browser being used to retrieve pages from a Web server, or a data analysis program accessing a warehouse of data on a database server. In situations where multiple users collaborate on a task, these multiple programs may interact with each other or may all make use of data on a single server.

The performance of these programs depends on the combined performance of the program on each computer as well as on the performance of the intervening network. The performance of the program itself is based on the processing power of the computer and the graphics system, the speed of disk access, and the throughput of the network card. The performance of the network depends on its bandwidth or throughput, as well as factors such as network delay, loss of data packets, and network jitter leading to alteration in the order of arrival of data packets.

In this paper, we focus on the impact of network parameters, and show how we design an evaluation system that generates an integrated measure of the performance of an application in the context of varying network conditions.

## Method

We conducted experiments to investigate the relationship between individual perception of application performance and experimenter controlled network parameters. We developed a subjective performance measure that supports analysis of the performance of individual applications, and comparison across applications. To develop this performance measure, we drew on the experience of audio and video telephony developers on subjective perception of telephony quality under varying network conditions.

### *Application: The Remote Stereo Viewer*

We developed an image serving application that delivers a sequence of linked images based on user movement of the mouse cursor. The simplest sequence was a set of rotated views of a human skull taken at intervals of 5 degrees of rotation. The images were delivered to the viewer in pairs, 5 degrees apart, so that they could be viewed in stereo depth using commercially available glasses that deliver the correct image of the pair to each eye. For comfortable viewing, smooth movement of the cursor should result in smooth rotation of the skull. Jerky rotation can also result in a loss of stereo perception.

A more complex set of images was that of the dissected hand (fig. 1). A fresh cadaver hand was placed on a turntable, and photographed at 5 degree intervals of rotation. It was removed from the turntable, a layer of tissue dissected, and the hand was photographed again. The process was repeated for six levels of dissection, with the last showing the bones of the hand. A total of 504 images were available for interactive viewing. Typical resolution is/was 1200x1500 pixels. Lateral movement of the mouse caused rotation of the hand and forward movement displayed dissections of increasing depth.

Other image sets include additional dimensions of interactive control corresponding to rotation about extra axes or zooming.

Each image were stored in JPEG format on the server. Users accessed these images through a client program on their individual computers. The client program monitored mouse movement and sent the control signals to the server. A monitoring program on the server received these signals, retrieved the appropriate pair of JPEG image files, and transmitted them to the client. The images were decompressed at the client and displayed to the left ad right eye respectively.

The client and server use a TCP connection to exchange information about the image set contents and to communicate requests. The actual transport of image data is handled using a reliable UDP layer protocol that supports recovery of lost of packets. Image decompression at the client occurs in parallel with image transport. The image transport rate is controlled to be comparable to the image decompression rate. The transport produces bursts of 30-40 Mbps. The UDP layer implementation allows image data to be transported over a multicast address to multiple collaborating clients and voids some overhead of the TCP layer.

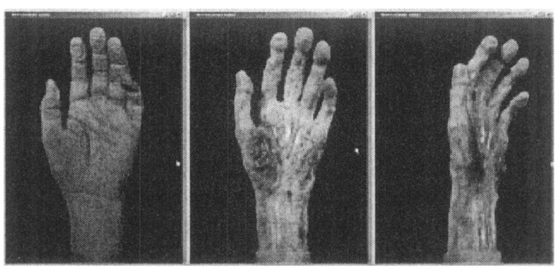

**Figure 1.** Images of the dissected hand. a) the fresh cadaver hand. b) the dissected hand with skin and fat removed. c) the same dissected hand, rotated.

***Application: The Haptic AutoHandshake***

The *Handshake* application is intended to train students remotely in various surgical procedures (fig 2). This is done by placing a haptic device at each end-point and having the instructor guide the movements of the student remotely. (A haptic device is one that delivers force feedback to the user so that the user has a sensation of touching and feeling a virtual object.) The *AutoHandshake* is a variant used when no instructor is available and the movements have been pre-recorded on a simulator located remotely. In this case, the simulator replays the proper movement that the student should make, and the student follows the movement using the local haptic device. Deviation from the pre-recorded movement results in corrective forces applied to the user's device. Network delay (latency) causes delay in the corrective force from the server, and can cause oscillation of the device.

This application uses a TCP connection to exchange setup information but then exchanges haptic data using UDP packets. The UDP packets are unreliable but because of the latency requirements of the application, retransmission of lost packets has little value. Unlike the JPEG server, the AutoHandshake is a low-bandwidth application, using only up to 128 Kbps.

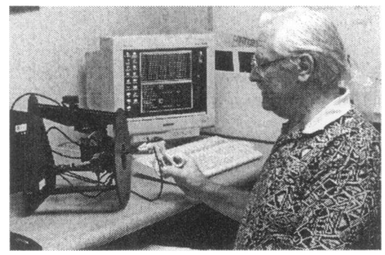

**Figure 2.** The user holds a haptic device that resembles a conventional surgical laparoscopic tool.

***Network Characterization***

Network performance can be described quantitatively using four parameters. These are: (i) throughput, the available bandwidth, (ii) packet loss, how many data packets are lost during transmission due to congestion, link failures or other problems, (iii) delay, the time it takes for a packet to get from one end-point to the other and (iv) jitter, the statistical variance of the delay. These parameters vary from network to network, and even on a particular network may vary at different times of the day. For example, depending on the route a packet takes to get from one point to another, it might need to pass through a different number of routers along the way leading to different delays in each case. Similarly, the packet loss seen on a network may significantly vary during the day. Routers have buffers to store incoming packets temporarily in the event the router is busy. During periods of high network traffic, these buffers may fill up resulting in packet delay, packet jitter, and even discarding of new incoming packets.

To measure various network connections, we used a commercial software package called Chariot, from NetIQ Corp. Small memory resident programs called Performance Endpoints are installed on the two computers that are located on the endpoints of the connection being evaluated. A single Controller module resides on one of the endpoints or on a third computer. This Controller module communicates with the endpoints and instructs them to generate network transmissions from a simulated application, whose only function is to generate network traffic. The simulated network application can generate traffic that varies in terms of the network transport protocol used (TCP, UDP, RTP), data rate, session duration, packet size, etc. As the network traffic is generated, the endpoints measure the average throughput, packet loss, delay and jitter and report these back to the Controller.

### *Network Emulation*

The network emulation consists of two endpoint computers and a third intervening computer that simulates the network. The two endpoint computers are the server hosting the application and the client computer, or two peer computers, depending on whether the application is a client-server application or a peer-to-peer application. The endpoints are linked to the emulator computer by gigabit fiber-optic connections.

The actual network emulation occurs on the computer in between the two endpoints. This computer manipulates the data traffic going between the two endpoints. A public-domain network emulator software, NISTNet, is used for this purpose. This software allows the emulator computer to be programmed to behave like different networks by allowing the values of packet loss, delay and jitter to be changed in a controlled fashion. Thus, using the values of the packet loss, delay and jitter obtained from the real networks as a reference, the emulator parameters are varied, while running real applications, to determine the boundaries of acceptable performance.

### *Measuring Application Performance*

The performance ratings were gathered using the absolute category rating (ACR) method, a subjective rating method that has been used to measure perception of performance of Internet based audio and video telephony applications [1-4]. In our ACR experiment, subjects use the applications described above (Image Server and Haptic Handshake) over a simulated network that may or may not be degraded. They are then asked to rate the quality of the experience, without reference to a known standard. A score of 5 indicates "Excellent: network degradation is not noticeable"; a score of 4 indicates "Good: network degradation is noticeable, but not annoying"; a score of 3 indicates "Fair: network degradation is evident and annoying"; a score of 2 indicates "Poor: network degradation is severely affecting the user experience and the application is minimally useable"; and 1 indicates "Bad: network degradation has made the application unusable".

The performance of the emulated network is degraded by changing the values for packet loss, delay or jitter. The amount of the parameter change is selected at random and is not known to the user. Users are asked to grade their perceived performance of the program according to the scale above. It is worth noting here that the users were familiar with the application under normal local Internet network conditions.

To test the Remote Stereo Viewer, the hand image set was used (figure 1). Six test subjects familiar with the Remote Stereo Viewer were asked to rate the application's performance as the values of packet loss, delay and jitter were randomly changed. Similarly, for the AutoHandshake application, a subjective evaluation was performed using six test subjects that were familiar with the application and its capabilities under no network stress. A Phantom Desktop haptic device, developed by SensAble Technology, was used during these tests. The AutoHandshake evaluation involved having the user make a movement with the Phantom device. The server repeated the movement for the user to follow. The user could then determine if the accuracy and smoothness of the movement being repeated was equal to that of the original.

## Results

### *Network characterization*

Network characterization, using the Chariot software, was performed for four different network connections:

a) two endpoints connected across a 100 Mbps local area network, (100LAN)
b) two endpoints connected across a wireless local area network (WLAN) using the IEEE 802.11 standard,
c) across an Internet connection with one endpoint at Stanford University and the other at the University of Wisconsin – LaCrosse, and
d) across an Internet connection with one endpoint at Stanford University and the other at Sweden's Royal Institute of Technology.

The latter two connections go through NGI research and education infrastructures, which provide much higher performing connections than across the commercial Internet. Table 1 shows the results that characterize the four networks using the throughputs and protocols used by our two applications.

### *Application performance measurement*

Application performance was estimated using a subjective performance metric, as network parameters were varied.

Figure 3 shows how the Remote Viewer performance degrades with increasing *packet loss*, reaching "Bad" performance levels at about 0.8% packet loss.

| | 10 Mbps, Reliable UDP (Remote Viewer) | | | | 128 Kbps, UDP (AutoHandshake) | | | |
|---|---|---|---|---|---|---|---|---|
| | WLAN | 100 Mbps LAN | Wisconsin | Sweden | WLAN | 100 Mbps LAN | Wisconsin | Sweden |
| **Packet Loss (%)** | N/A | 0.000 | 0.006 | 0.130 | 0.000 | 0.000 | 0.000 | 0.024 |
| **Delay (ms)** | N/A | 0.200 | 29.500 | 82.500 | 1.750 | 0.200 | 29.500 | 82.500 |
| **Jitter (ms)** | N/A | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

**Table 1.** Characteristics of four different networks, using throughput and protocols characteristic of two of our applications. Measurements were averaged over different times of the day. (WLAN is a wireless local area network.)

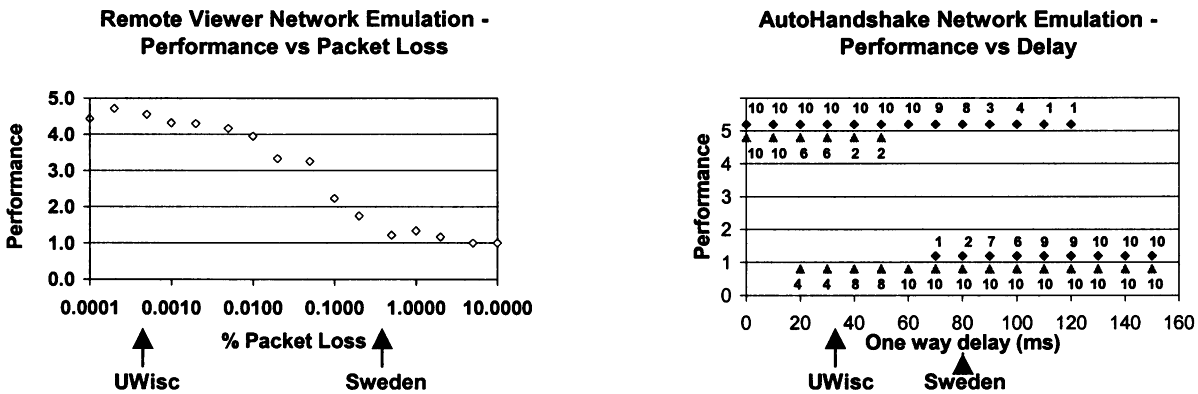


**Figure 3.** Subjective performance estimates for two different applications operating over a simulated network. Changes in simulated packet loss and delay were applied in random order. (a) Six subjects estimated performance of the Remote Viewer application over 17 trials each. (b) Performance of the Autohandshake was estimated over 10 trials for an abrupt (triangle marker) and a gentle (diamond marker) movement. Parameter values for two networks from Table 1 are indicated with markers (UWisc and Sweden).

Since the application retransmits packets that are in error, packet loss causes the images to be delayed. The application's performance also degrades gradually with increasing *delay*, but it never reaches the "Bad" level, because the image may be delayed but it is never lost. The application is fairly insensitive to reasonable values of *jitter*. The jitter-induced delay is barely noticed by the end user compared to the inherent delay in decoding and displaying the images.

The AutoHandshake application is resilient to packet loss, starting to degrade only at values higher than 1%. Packet loss causes loss of a few points that describe the path that the haptic tool should be following. Since the sampling of points along the movement path is high, these missed points are averaged from the ones that do arrive, making the application insensitive to packet loss. However, network delay (fig 3b) has a significant impact. The AutoHandshake application either works well or it does not work at all. As network delay increases, the application becomes unstable and therefore unusable. The results differ depending on whether the movement is gentle or abrupt. The AutoHandshake performance is also sensitive to jitter, reaching "Bad" performance levels at around 15 ms of jitter. Jitter causes degradation because the arrival of points out of order causes the path to loose smoothness, which is perceived by the end user.

***End-to-end performance requirement***

Each application imposes a minimum performance requirement on the network. As we see above, the Remote Viewer application requires a high bandwidth but is not very sensitive to packet loss, delay or jitter on the network. The AutoHandshake, on the other hand, is very sensitive to network delay and jitter, but is less sensitive to packet loss, and does not require high bandwidth. Table 2 summarizes the requirements that these two applications impose on the network. These requirements have been selected as the values where the evaluated performance is rated "good" or better. It is worth noting here that this evaluation has been made on a per parameter basis, that is, each parameter was treated independently. The results that different mixtures of these parameters may bring have not yet been studied.

| | Remote Stereo Viewer | AutoHandshake |
|---|---|---|
| Bandwidth | 40 Mbps | 128 Kbps |
| Packet Loss | < .01 % | < 10% |
| Delay (one way) | < 100 ms | < 20 ms (abrupt movement)<br>< 80 ms (gentle movement) |
| Jitter | Not sensitive to jitter | < 1 ms |

**Table 2.** Summary of minimum end-to-end performance requirements for network.

### *Application performance on real networks*

We can estimate how these applications may perform on real networks by correlating the data in tables 1 and 2. In figure 3, the network connection (UWISC) from Stanford University to University of Wisconsin, will support good performance of the Remote Viewer application, but will support only inconsistent use of the AutoHandshake with abrupt movements because the delay to Wisconsin exceeds the requirement of the application. The network connection from Stanford University to KTH, Sweden (SWEDEN) will not support the Remote Viewer due to packet loss and delay, and will not allow the normal operation of the AutoHandshake application, even with gentle movements. It is worth noting here that both of these connections use NGI network infrastructure, but the application requirements are higher than this network can support.

## Discussion

We have characterized the network requirements for two applications and have indicated how they might perform over four different types of networks. In actual practice, we expect these applications to be used by multiple students at the same time. These applications have been designed for collaborative use, where a group of students, or a teacher and many students interact together on an image collection or a graphic and haptic three-dimensional model of anatomy. Therefore, it is desirable if the performance characterization presented above can be extended to describe a collaborative learning environment.

### *Network Weather prediction*

In the future, we expect to be able to answer the following type of question: "How many remote sites can I teach today if I use a collaborative application that accesses a library of interactive images?" Each application will be characterized in terms of its network requirements. Simultaneous collaborative use of the application will also be characterized. Programs that monitor network traffic continuously will predict the expected traffic at the time of the class. An analysis of predicted traffic, together with the known requirements of the application, will answer the above question. An answer may be: "At 3pm today, you can teach only 5 sites if you use the image server with stereo images. If you restrict use of stereo to one site at a time, and allow the rest to view images without stereo, you can expand your class to 12 sites."

## Conclusions

We have presented a method to obtain an end-to-end characterization of the performance of an application over a network. This method is not dependent on any specific application or type of network. We therefore believe that this method may be of wide applicability as research and educational applications increasingly make use of computational and data servers that are distributed over the Internet.

## Acknowledgements

This work was partially supported by NLM Contract N01-LM-0-3506 for the Next Generation Internet. A preliminary report was presented at EduCause 2002.

## References

1. Webster, A., Jones C., Pinson M., Voran S. and Wolf S. "An objective video quality assessment system based on human perception," SPIE Human Vision, Visual Processing, and Digital Display IV, February 1993.

2. ITU-T Recommendations: P.800. Methods for subjective determination of transmission quality. Recommendations of the International Telecommunications Union (Telephone transmission quality, telephone installations, local line networks), 1996.

3. ITU-T Recommendations: P.910. Subjective video quality assessment methods for multimedia applications, Recommendations of the ITU (Telecommunication Standardization Sector), 1999.

4. Jones, C. and Atkinson, D.J. "Development of Opinion-Based Audiovisual Quality Models for Desktop Video-Teleconferencing". 6th IEEE International Workshop on Quality of Service, Napa, California. May 1998.