Comparison of Performance of Virtual Coupling Schemes for Haptic Collaboration using Real and Emulated Internet Connections

Ganesh Sankaranarayanan and Blake Hannaford
BioRobotics Laboratory, Department of Electrical Engineering, University of Washington
Seattle, Washington 98195
Email: {ganeshs,blake}@u.washington.edu

***Abstract*— Networked haptic virtual environments (NHVEs) are those in which multiple users collaborate and experience force feedback at the same time. The robustness of such systems needs to be tested under various network conditions that closely mirror the Internet. Previously, we had proposed three virtual coupling schemes to maintain position coherency in a NHVE, which were tested using constant and then time-varying delays using the actual Internet through UDP packet reflectors. In this paper we present the results of comparing performance of the virtual coupling schemes for a time varying delay emulated using the popular network emulator NIST Net, with delay conditions that existed during our real Internet experiment to Italy. UDP was used for haptic data communication because of the high transmission rate requirements for NHVEs. Experiments were conducted for three fixed packet transmission rates of 1000, 500 and 100 Hz, and their performance compared using an independent-samples t-test to the data obtained using the Internet. Locally, the haptic update rate was maintained at 1000 Hz during the experiments. Our results show that the NIST Net was a suitable emulator for testing with lower packet transmission rates. At the transmission rate of 1000 Hz the performance of the virtual coupling schemes were significantly different from that of the actual Internet experiment.**

I. Introduction

Networked Haptic Virtual Environments (NHVEs) are suited for very diverse applications —from networked games to surgical [1] and maintenance procedure training [2]— since they allow multiple users to collaborate on a shared virtual space and experience force feedback at the same time. In this manner, they manage to enhance game realism in networked games, improve time management in aircraft assembly, and sharpen training skills for surgery.

There are, however, several obstacles in the implementation of NHVEs: the overall system needs to remain stable and the position coherency between copies of the virtual object must be maintained. The former is important especially in bilateral systems, since they are propense to destabilize rapidly when faced with time delays, and the latter is needed to prevent the virtual objects from drifting apart. In addition, the haptic update rate needs to stay at 1000 Hz to be able to feel stiffer objects [3], a condition that can consume a considerable portion of the network bandwidth.

In order to enable haptic collaboration, NHVEs are connected to each other in distributed (peer-to-peer) or centralized (client-server) architectures through either a dedicated network or the Internet. Therefore, testing the robustness of a NHVE system requires realistic network conditions, while a laboratory setting allows the system to be fine-tuned for performance. To this end, network emulators such as NIST Net, Dummynet and Hitbox are used to locally recreate network conditions. These systems can also be helpful to test teleoperation and other systems sensitive to delays. NIST Net is a Linux-based open source software package which allows the user to define the delay table and configurate the mean delay, standard deviation, bandwidth, packet drop percentage, packet duplication, etc., presenting both graphical and command line interfaces.

The client-server structure is capable of maintaining position coherency but adds round-trip delay in between the clients and the server, since the central server manages only one copy of the virtual object. On the other hand, in a peer-to-peer architecture each peer maintains a copy of the virtual object and updates it locally, resulting in half the delay but increasing the possibility of position discrepancy among the virtual objects.

A. Position Coherency

We define position coherency in a NHVE as the position difference between the multiple copies of a virtual object at a single instant of global time.

In our previous work we proposed *virtual coupling schemes* to enforce position coherency in a NHVE and tested it with constant time delays [4], compared two virtual coupling schemes (a peer to peer architecture and a client-server one) on a global scale Internet connection [5], and implemented network emulator capable of Internet-like features [6] for haptic data communications.

B. Goals of this study

Our objective for this paper is to compare the performances of the three virtual coupling schemes with a time varying network delay condition emulated using NIST Net to actual Internet network conditions at three fixed packet transmission rates. The delay value and the experimental parameters are chosen so that the experimental system would result in stable operation.

ROBOCOMM 2007, 14th − 16th Oct 2007, Athens, Greece.

ISBN 978-963-9799-08-0
DOI 10.4108/ICST.ROBOCOMM2007.2219

II. Background

Previous works in the area of NHVE can be classified according to the type of control or connection, and the type of force rendering. The first criterion differentiates between centralized or client-server architectures [7], [8], [9], [10], [11] and distributed or peer-to-peer [12], [13], [14], [15], while [4] and [16] utilize both centralized and distributed architectures to study NHVEs. The second characteristic contrasts continuous —or simultaneous— force rendering [8], [9], [10], [14], [4], [15] [16] where both users manipulate a virtual object at the same time, and one-at-a-time, [12], [13], [7]; in which each user takes turns to manipulate the virtual object.

The addition of haptics has several positive effects. In [17], [18], [19] it was proven useful in increasing the task performance and virtual presence felt by the users, who had to jointly manipulate a shared virtual environment and whose performances were analyzed. However, latency between users was not studied in this case. In contrast, [20] included network delay in a cooperative shared haptic virtual environment, where it was found to reduce task performance.

In addition, in [8] time varying delay was assumed to be a Gaussian distribution with an expectation value of δ. In [9] and [10] time varying delay was emulated by interference produced using two workstations with one of them sending fixed data size packets to the other, both sharing the same Ethernet hub with the client and server used for the experiments. Also, in [14] the actual Internet was used for the experiment: the bandwidth usage was reduced by sending packets only when at least one of the participants was in contact with the virtual object in the simulation.

In [21] the end-to-end packet delay and loss behavior of the Internet was studied using a UDP probe packet at different data rates in which they found rapid fluctuations of queuing delays over small intervals, and the compression of the probe packets and packet loss were random, unless the probe packets used a large amount of the available bandwidth. In [22] the Internet delay and loss behavior was studied using TCP packets to model out-of-order packets and reordering. They found that a large percentage of packet reordering was common throughout the Internet and it was correlated to routing fluctuations.

Also, in order to provide fairness among all the participants in a client-server architecture and to mitigate the effects due to delay jitter, media synchronization techniques available in video and audio streaming [23] were applied to haptic media in [9], [8], and [10]. In [15], a motion synchronization technique based on a feedback controller was used to maintain consistency in a two-user peer-to-peer system. Their system was effective upto a roundtrip delay of 300 ms. Both [1] and [14] used the actual Internet in implementing their system.

Furthermore, in our previous work [5], virtual coupling schemes were tested for time varying delays using the Internet for three fixed transmission rates (1000, 500 and 100 Hz). UDP packet reflector program was hosted at the servers of our collaborators and routed the haptic UDP data packets between two users. The network topology that was used during the experiment is shown in Fig. 1. Particularly, the network route for the packets between University of Washington (UW) and Scuola Sup. Sant'Anna (SSSUP) is through NLR (the National Lambda Rail), GEANT2 and GARR networks (both European gigabit research networks); a switching node in New York City connects GEANT2 to NLR. The packet reflector locations along with the local network connection constituted four time varying delay conditions whose values, obtained during the experiments using UDP packets, are shown in Table I.

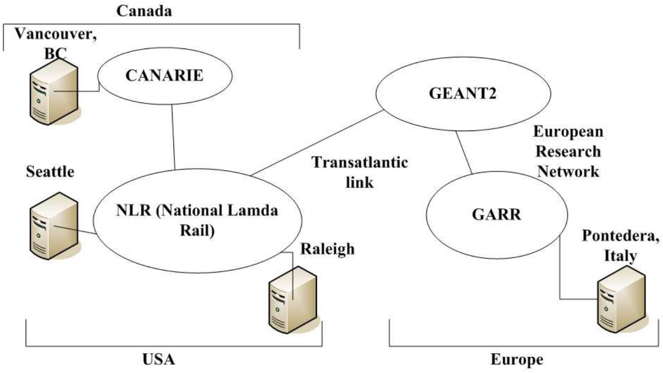


Fig. 1. Internet experiment network topology

TABLE I

Delay Condition with Packet Reflector Location

Delay Condition	Mean (ms)	Std (ms)	Hosts
1	1.27	0.35	Local
2	43.8	0.53	UBC, Canada
3	74.2	0.63	NCSU, USA
4	203.9	2.52	SSSUP, Italy

III. Methods

A. Virtual Coupling Schemes

In this paper we adopt three virtual coupling schemes to enforce position coherency in a networked haptic virtual environment that support two users with one degree of freedom. The purpose of these schemes is minimizing the error between the positions of the virtual copies of the rigid bodies, x_2 and x_3, while x_1 and x_4 represent the position of the haptic devices.

To this end, the haptic handles in Figs. 2, 3, and 4, H_{1-1} and H_{1-2} represent the virtual copies of the haptic device positions, and O_{1-1}, O_{1-2} represent the virtual copies of the rigid body with which the two users interact. K_{VC1}, B_{VC1}, K_{VC2} and B_{VC2} are the virtual couplings, and the interaction between the human operator and the haptic device is represented by an impedance type haptic display. Also, viscous damping B_d is added between the cube and the surface to keep the mass from drifting off when users are not in contact.

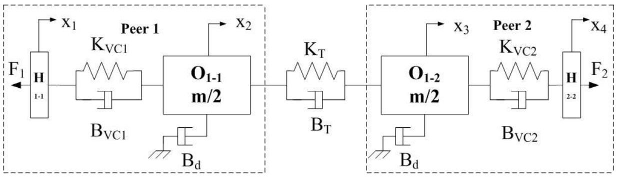


Fig. 2. Virtual Coupling Scheme 1

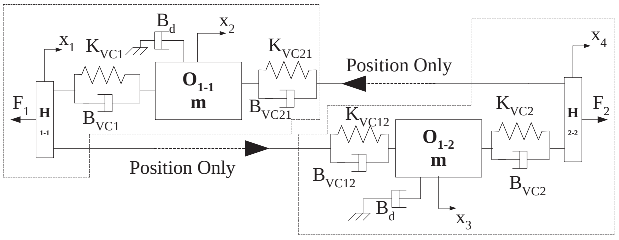


Fig. 3. Virtual Coupling Scheme 2

As we can see in the figures, the dotted lines separate the local and the remote user in a communication network. In the following equations, $T_1(t)$ and $T_2(t)$ represent the time varying delays between peers and clients for the communication path left to right and right to left.

1) Scheme1: In scheme 1(Fig. 2)we present a peer-to-peer architecture with two virtual copies of the rigid bodies connected by a virtual coupling whose network is implemented through a position-position architecture, analogous to that in [24]. This virtual coupling transmits forces between the virtual copies and allows tracking between them. To guarantee the consistency of the total mass felt by each user, the masses of the virtual objects were divided between their virtual copies. Equations 1 to 4, give the control force acting on the virtual objects along with the reaction force applied to each user.

$$F_1 = K_{VC1}(x_2(t) - x_1(t)) + B_{VC1}(\dot{x_2}(t) - \dot{x_1}(t)) \tag{1}$$

$$\begin{aligned}\frac{m}{2}\ddot{x_2}(t) - B_d\dot{x_2}(t) &= K_T(x_3(t - T_2(t)) - x_2(t)) \\ &+ B_T(\dot{x_3}(t - T_2(t)) - \dot{x_2}(t)) - F_1\end{aligned} \tag{2}$$

$$F_2 = K_{VC2}(x_3(t) - x_4(t)) + B_{VC2}(\dot{x_3}(t) - \dot{x_4}(t)) \tag{3}$$

$$\begin{aligned}\frac{m}{2}\ddot{x_3}(t) - B_d\dot{x_3}(t) &= K_T(x_2(t - T_1(t)) - x_3(t)) \\ &+ B_T(\dot{x_2}(t - T_1(t)) - \dot{x_3}(t)) - F_2\end{aligned} \tag{4}$$

2) Scheme2: Scheme 2 (Fig. 3) represents a peer-to-peer architecture in which the position of the users are coupled to both local and remote copies of the virtual object. A local virtual coupling joins the position of the remote user to the virtual object at their respective ends, while the positions of the users are transmitted across the network. Equations 5 to 8 describe the scheme 2.

$$F_1 = K_{VC1}(x_2(t) - x_1(t)) + B_{VC1}(\dot{x_2}(t) - \dot{x_1}(t)) \tag{5}$$

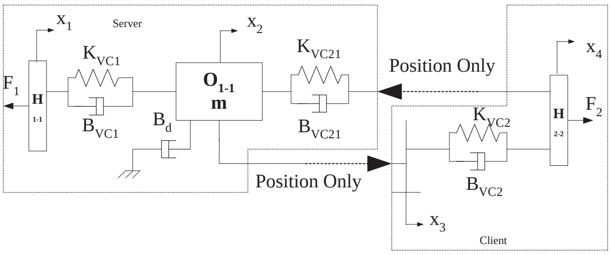


Fig. 4. Virtual Coupling Scheme 3

$$\begin{aligned}m\ddot{x_2}(t) - B_d\dot{x_2}(t) &= K_{VC21}(x_4(t - T_2(t)) - x_2(t)) \\ &+ B_{VC21}(\dot{x_4}(t - T_2(t)) - \dot{x_2}(t)) - F_1\end{aligned} \tag{6}$$

$$F_2 = K_{VC2}(x_3(t) - x_4(t)) + B_{VC2}(\dot{x_3}(t) - \dot{x_4}(t)) \tag{7}$$

$$\begin{aligned}m\ddot{x_3}(t) - B_d\dot{x_3}(t) &= K_{VC12}(x_1(t - T_1(t)) - x_3(t)) \\ &+ B_{VC12}(\dot{x_1}(t - T_1(t)) - \dot{x_3}(t)) - F_2\end{aligned} \tag{8}$$

3) Scheme3: In scheme 3 (Fig. 4) we introduce a client-server architecture, in which user 1 is located on the server and user 2 is located remotely. The server maintains the only copy of the virtual object, while each user transmits their position through the network to the server, with the server performing collision detection between the position of the users and the virtual object. As a result, each client is then updated with the new position of the virtual object. The position update of the virtual object is transmitted back to user 2 who then uses a local virtual coupling to generate the appropriate reaction forces. Equations 9 to 12 describe this scheme.

$$F_1 = K_{VC1}(x_2(t) - x_1(t)) + B_{VC1}(\dot{x_2}(t) - \dot{x_1}(t)) \tag{9}$$

$$\begin{aligned}m\ddot{x_2}(t) - B_d\dot{x_2}(t) &= K_{VC21}(x_4(t - T_2(t)) - x_2(t)) \\ &+ B_{VC21}(\dot{x_4}(t - T_2(t)) - \dot{x_2}(t)) - F_1\end{aligned} \tag{10}$$

$$x_3(t) = x_2(t - T_1(t)) \tag{11}$$

$$F_2 = K_{VC2}(x_3(t) - x_4(t)) + B_{VC2}(\dot{x_3}(t) - \dot{x_4}(t)) \tag{12}$$

B. Generation of Random Delay Variables

The NIST Net software generates delay random variables with specified mean and standard deviation based on the input delay distribution table.

Let X be the normalized random variable having a probability density function $f(x)$. Then the cumulative distribution function *cdf* of the continuous random variable X is given by equation 13.

$$F_X(x) = P[X < x] = \int_{\infty}^{x} f(t)dt \tag{13}$$

Assume that inverse of the *cdf* exists and is given in equation 14.

$$Z_Y(y) = F_X^{-1}(x) \tag{14}$$

Let U be a uniformly distributed random variable in the interval [0,1]. Then, a desired random variable Y can be obtained as shown in equation 15. The proof of this is straightforward and it is shown in equation 16.

$$Y = Z(U) \tag{15}$$

$$\begin{aligned} P[Y \leq x] &= P[Z(U) \leq x] \\ &= P[U \leq F_X(x)] = F_X(x) \end{aligned} \tag{16}$$

In NIST Net, $F_X(x)$ is the *cdf* of the normalized delay distribution which is obtained through measurements of a network like Internet. Let the new distribution that is generated using $F_X(x)$ have a desired mean delay $D\ ms$ and standard deviation $\sigma\ ms$. Then, the NIST Net generated delay time is given in equation 17.

$$Y = D + Z(U) * \sigma \tag{17}$$

For a more detailed implementation of correlation, random packet drop, and other parameters refer to [25].

IV. Experiment Setup

The experimental collaborative haptic system consists of a collaboration software framework that implements the virtual coupling schemes and the network emulator setup which enables experiments using the emulator.

A. Collaboration Software Framework

The collaboration software framework consists of two separate programs. The *haptic controller* program (written in C++) manages communication with the haptic device, performs collision detection and dynamic updates of the virtual object as well as renders the appropriate forces to the user, and also performs haptic UDP communications at variable rates. An OpenGL graphical display application based on Qt 4.1.1.1 was used to manage the graphic display. The display is updated from the *haptic controller* at 30 Hz through a TCP/IP link. UDP is used for haptic data communication because of the high packet transmission rate requirements. Each UDP haptic data packet has a unique packet sequence number generated in the haptic servo loop and a checksum field generated using a simple checksum algorithm. The out-of-sequence and corrupt packets are detected by the *haptic controller* program and are dropped. The timestamp field in the haptic data packets are used to compute network delay characteristics.

B. Network Emulator Setup

Figure 5 shows the network emulator setup, in which WS1 and WS2 are the two workstations used in the experiment by user 1 and 2 respectively. WS1 is an AMD Opteran 1.5 GHz with 1 GB RAM and WS2 is an Intel Pentium 4 2.66 GHz with 1GB RAM with both running Fedora Core 4 Linux. WS3, an AMD Athlon 1.2 GHz computer with 256 MB RAM and two 10/100 Mbs network cards, is a router that runs NIST Net for network emulation. Both workstations are situated in the same laboratory but in different cubicles. Both the *haptic controller* and graphic applications are run on WS1 for user 1 and on WS2 for user 2. Additionally, a TCP/IP link between the two users was set to initiate target tracking during the experiments. A PHANToM Omni —the commercially available 6 DOF sensing and 3 DOF force-rendering device— was used for all the experiments.

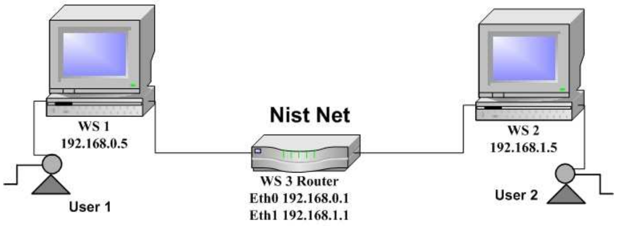


Fig. 5. Network emulator experiment setup

The snapshot of the haptic virtual environment as seen by the user 1 at WS1 is shown in Fig. 6. It consists of a cube that is restricted to one degree of freedom movement along the floor in a three dimensional room. Blue and yellow spheres represent the local and remote user positions. The haptic virtual environment also contains a target sphere that moves at a constant speed of 1.333 mm/s along a solid white line. The task consists of jointly moving the cube along the line and making the target align at the center of the cube at all times. User 1 was allowed only to touch the left side of the cube and user 2, the right side. Collison detection between the position of the user and the cube was computed and newtonian dynamics was used to update the position of the cube. The collision between the walls and the cube also restricts the latter to be in the visible scene at all times. A similar virtual environment with its display rotated (for user 2 to see the right side of the cube) is shown in WS2.

Additional attributes were added to the haptic virtual environment to aid the users during the experiments. The color of the cube —which is initially white— is changed to blue when the users make contact, as long as the reaction force applied reaches or exceeds 10% of 3.3N (the maximum recommended continuous force for the Omni haptic device). The color of the cube changes to red when the user reaction force is greater than 3.3N, while the color of the target sphere changes from green to purple during tracking and the solid white line hints the user as to when the target is going to change direction.

V. Experiment Procedure

From our previous global scale Internet haptic experiment we chose to present, using the network emulator, the delay mean and standard deviation values corresponding to the packet reflector location at Italy. The delay distribution curve for packets from WS2 to WS1 from one of the experiments to Italy was given as input to NIST Net. From our previous use of NIST Net for haptic data communication [6], we

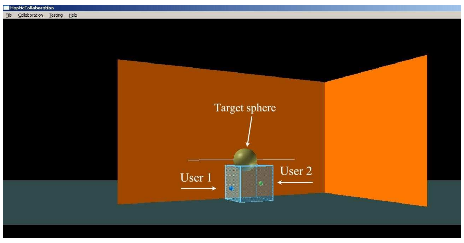


Fig. 6. Snapshot of the haptic virtual environment at WS1

found that during the emulation a large number of out-of-sequence packets were generated, which were not common in the Internet. We had mitigated the problem by modifying the output FIFO of the NIST Net to force the datagram packets from swapping while it was waiting at the output buffer. The drawback of this modification is that an additional queuing delay was added to each packet. Several trials were conducted by transmitting UDP haptic data packets between WS2 and WS1 and the emulator delay parameters were chosen so that they would give the closest possible match to the data from the Italy experiment, as shown in Table I. The emulator delay parameters used in the experiment are shown in Table II, while additional NIST Net parameters like random drop and duplicate packets were not utilized. The delay conditions were applied only to the UDP communication between WS1 and WS2 and the target initiation signals between WS1 and WS2 that uses TCP were not delayed.

TABLE II
NIST NET EMULATOR DELAY PARAMETERS

Delay Condition	Mean (ms)	Std (ms)
WS1 to WS2	197.0	2.4
WS2 to WS1	197.0	2.4

Ten subjects, eight male and two female, with ages ranging from 22 to 50 years old were selected for these experiments. The experimenter acted as user 1 and each subject as user 2. One of the subjects was left-handed and the rest were right-handed. Through an instructional video, each subject was given a detailed explanation of what they were going to do in the experiment. They were then given five practice trials to get comfortable with the virtual environment and the interaction with the cube, while their performance was monitored. After five trials, if they were able to comfortably track the target, they proceeded to the actual experiment. The subjects were asked to touch and apply forces on the right side of the cube as the experimenter interacted on the left side. They maintained contact with the cube at all times and waited for the color of the target to change to start moving. Whenever the color of the cube changed to red, they were instructed to apply less force on the cube until the color reverted back to blue. The order of the schemes and the packet transmission rate were randomized during the experiment to reduce the learning effect.

The experiment was conducted for the three virtual coupling schemes using the parameters shown in Table III. Every scheme was tested for three packet transmission rates: 1000, 500 and 100 Hz with a total of 9 trials for each subject. The position and velocity of the two cubes, along with the position of the target and forces rendered to each user, the number of corrupt, as well as out-of-sequence and lost packets, and the delay in ms computed from the time stamp information, were recorded in a file for analysis. Each objective measure was then compared to those obtained using the Internet experiment to Italy through an independent-samples t-test. For three virtual coupling schemes and three packet transmission rates, a total of nine independent-samples t-tests were performed for each measure.

TABLE III
VIRTUAL COUPLING PARAMETERS

Parameters	Values	Schemes	Selection
m	0.25Kg	1, 2, 3	experimental
B_d	0.025 $\frac{Ns}{mm}$	1, 2, 3	experimental
K_T, K_{VC21}, K_{VC12}	2.0 $\frac{N}{mm}$	1,2	experimental
B_T, B_{VC21}, B_{VC12}	0.3 $\frac{Ns}{mm}$	1,2	experimental
K_{VC1}, K_{VC2}, K_{VC12}, K_{VC21}	0.5 $\frac{N}{mm}$	1,3	device manual
B_{VC1}, B_{VC2}, B_{VC12}, B_{VC21}	0.003 $\frac{Ns}{mm}$	1,3	device manual

VI. RESULTS

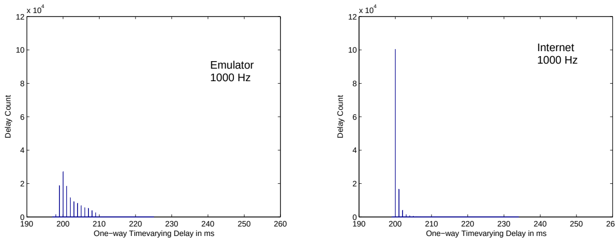
The histograms of the emulated one-way delay between WS1 and WS2 and the data from the Italy experiment for packet transmission rates of 1000, 500 and 100 Hz are shown in Figs. 7, 8 and 9 respectively. The independent axis represents the time in ms and the dependent axis, the count of the one-way delays in ms measured at WS1. All the delay values had a peak value around their respective means and a long tail towards increasing delay.

Table IV show the mean delay values and the standard deviation obtained using the emulator for the three packet transmission rates.

TABLE IV
ONE-WAY EMULATED DELAY MEASURED AT WS1

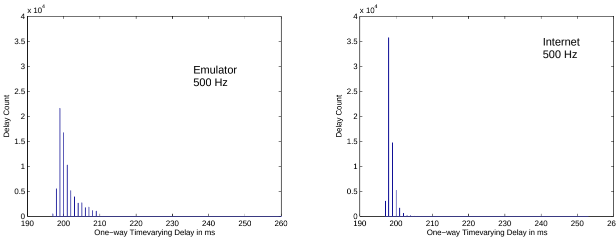
Packet Rate (Hz)	Mean (ms)	Std (ms)
1000	203.27	2.75
500	202.21	2.52
100	201.2	2.23

Figure 10 shows a typical output of target tracking from a single trial.



(a) **(b)**

Fig. 7. Histogram of one-way time varying delay for 1000 Hz packet transmission rate using NIST net, and with data from the Italy experiment

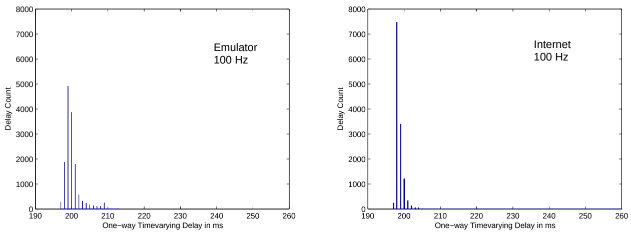


(a) **(b)**

Fig. 8. Histogram of one-way time varying delay for 500 Hz packet transmission rate using NIST net, and with data from the Italy experiment

Figure 11 shows the RMS position error between x_2 and x_3 from the experiments using the NIST Net and the Internet for the three packet transmission rates of 1000, 500 and 100 Hz. The vertical error bars represent the standard deviation around mean error values. In the experiment using the NIST Net, the RMS position error ranged from 0.82 mm to 0.83 mm for scheme 1, from 1.04 mm to 1.12 mm for scheme 2, and from 0.73 mm to 0.80 mm for the scheme 3. In contrast, in the experiment using the Internet, the RMS position error ranged from 0.67 mm to 0.86 mm for scheme 1, from 0.93 mm to 1.13 mm for scheme 2, and from 0.76 mm to 0.83 mm for the scheme 3. The independent-samples t-test between the emulator and the Internet experiment show that the RMS position errors for scheme 1 were significantly different, at 1000 Hz with a p value of 0.007.

Figure 12 shows the peak position error between x_2 and x_3 from the experiments using the NIST Net and the Internet for the three packet transmission rates of 1000, 500 and 100 Hz. In the experiment using the NIST Net, the peak position error ranged from 1.80 mm to 2.21 mm for scheme 1, from 2.74 mm to 3.39 mm for scheme 2, and from 1.92 mm to 2.31 mm for the scheme3, while in the experiment using the Internet, the peak position error ranged from 1.59 mm to 2.00 mm for scheme 1, from 2.76 mm to 3.26 mm for scheme 2, and from 1.93 mm to 2.52 mm for the scheme 3. The independent-samples t-test between the emulator and the Internet experiment show that the peak position errors for scheme 1 were significantly different, at 1000 Hz with a p value of 0.029.



(a) **(b)**

Fig. 9. Histogram of one-way time varying delay for 100 Hz packet transmission rate using NIST Net, and with data from the Italy experiment

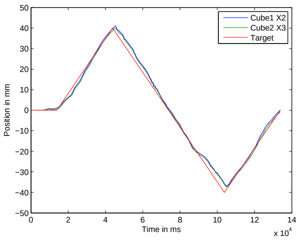


Fig. 10. Target tracking output for a trial (Scheme 1, 1000 Hz, Subject 6)

Figure 13 shows the RMS force applied to user 1 for all the three schemes, from the experiments using the NIST Net and the Internet for the three packet transmission rates of 1000, 500 and 100 Hz. The lower and upper force thresholds used for changing the color of the cube are shown in horizontal dotted lines. In the experiment using the NIST Net, the RMS force ranged from 1.63 N to 1.66 N for scheme 1, from 1.44 N to 1.50 N for scheme 2, and from 1.45 N to 1.56 N for the scheme 3, while in the experiment using the Internet, the RMS force ranged from 1.36 N to 1.69 N for scheme 1, from 1.17 N to 1.51 N for scheme 2, and from 1.02 N to 1.50 N for the scheme 3. All the force values were within the upper and lower force thresholds. The independent-samples t-test between the emulator and the Internet experiment show that the RMS force for all the three schemes were significantly different, at 1000 Hz with a p value of 0.007, 0.015 and 0.013 respectively.

Figure 14 shows the RMS force applied to user 2 for all the three schemes, from the experiments using the NIST Net and the Internet for the three packet transmission rates of 1000, 500 and 100 Hz. In the experiment using the NIST Net, the RMS force ranged from 1.68 N to 1.73 N for scheme 1, from 1.52 N to 1.61 N for scheme 2, and from 1.49 N to 1.60 N for scheme 3 respectively, while in the experiment using the Internet, the RMS force ranged from 1.33 N to 1.66 N for scheme 1, from 1.32 N to 1.57 N for scheme 2, and from 1.20 N to 1.55 N for the scheme 3. All the force values were within the upper and lower force thresholds. The independent-samples t-test between the emulator and the Internet experiment show that the RMS force for scheme 1 and scheme 2 was significantly different, at 1000 Hz with a p value of 0.004 and 0.034 respectively. The RMS force applied

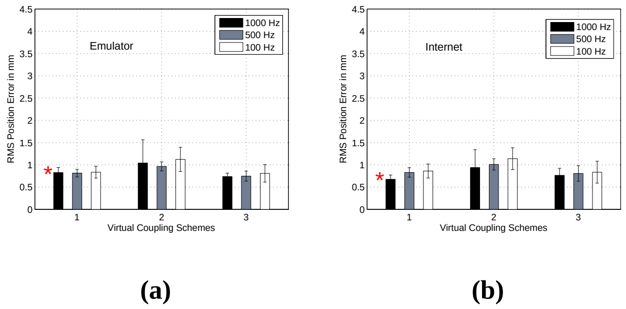


Fig. 11. RMS position error between cube 1 and cube 2 for all the schemes using NIST Net, and with data from the Italy experiment

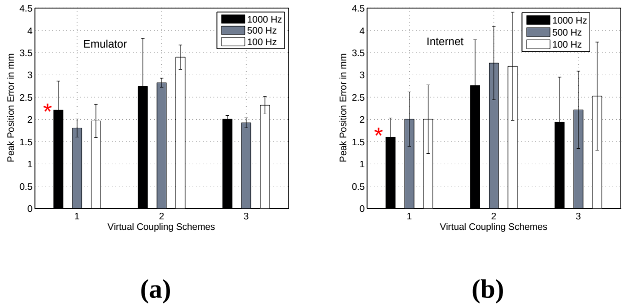


Fig. 12. Peak position error between cube 1 and cube 2 for all the schemes using NIST Net, and with data from the Italy experiment

to user 2 for scheme 3 at 1000 Hz had a p value of 0.093 and was considered not significant.

Figure 15 shows the network packet loss characteristics as a percentage of total packets used in the experiments using the NIST Net and the Internet computed at WS1. The horizontal axis is the packet transmission rate in Hz and the vertical axis is the packet loss percentage. The network data from all the schemes for each packet transmission rate was combined to compute the percentage of out-of-sequence packets and the percentage of packets lost in transmission. The out-of-sequence packet percentages from the Internet experiments were 0.0465%, 0.02% and 0.0172% for 1000, 500 and 100 Hz packet transmission rates respectievly. From the experiments using the NIST Net, it was 11.56%, 5.67% and 0.62% respectively. Overall, the percentage of out-of-sequence was insignificant in the Internet experiment compared to NIST Net. The percentages of packets that were lost (transmitted from WS2 but never arrived to WS1) from the Internet experiment were 2.90%, 4.05% and 2.8% for 1000, 500 and 100 Hz packet transmission rates respectively. For the experiments using the NIST Net, it was 21.81%, 6.87% and 2.29%. Overall the percentages of packets that were lost in the transmission were significantly higher in the experiments using the NIST Net for the 1000 Hz packet transmission rate. For the other two packet transmission rates, the loss percentage were within 2% difference.

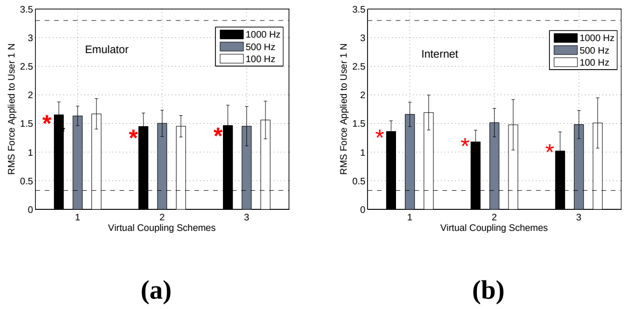


Fig. 13. RMS force user applied to user 1 for all the schemes using NIST Net, and with data from the Italy experiment

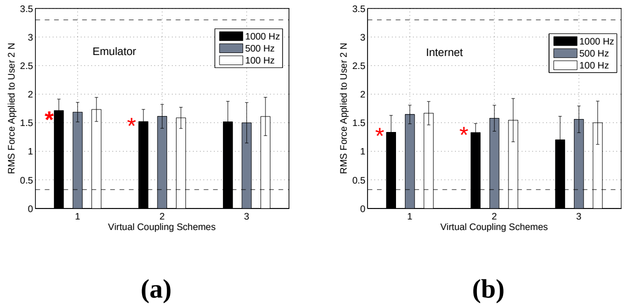


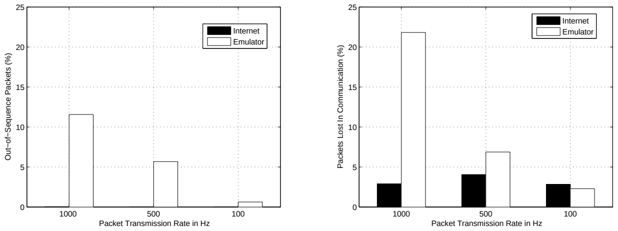
Fig. 14. RMS force user applied to user 2 for all the schemes using NIST Net, and with data from the Italy experiment

VII. Discussion

From the experimental results, the objective performance measures of the virtual coupling schemes obtained by using the network emulator were found to be significantly different from that of the data obtained using the actual Internet experiment for the packet transmission rate of 1000 Hz. However, for the packet transmission rates of 500 and 100 Hz, the t-test did not report any significant differences between the two. Virtual coupling scheme 1 is found to be more sensitive to emulated network delays compared to the other two virtual coupling schemes. This is not surprising since scheme 1 is the most sensitive to delay conditions because of the direct coupling between the copies of the virtual object.

The number of out-of-sequence packets was considerably higher while using the network emulator as compared to the experiment using the Internet. We had already adopted the modified FIFO in the NIST Net implementation to mitigate this problem, which greatly helped in reducing this number but proved not enough for NHVE application at 1000 Hz packet transmission rates. The number of packets that were lost in the transmission were also higher with the 1000 Hz transmission rate. These losses had considerable effect on the force applied to the user because of the zero-order hold (ZOH) in the virtual coupling schemes.

In conclusion, network emulators appear suitable for testing networked haptic virtual environments at lower packet transmission rates. Careful consideration of design parameters and system sensitivity to packet loss are to be considered to test the system at higher packet transmission rates.



(a) **(b)**

Fig. 15. Comparison of out-of-sequence packets, and packets lost in transmission at WS1

In this work the virtual coupling schemes were tested for just one emulated delay condition. In the future, we plan to compare the results using other network conditions obtained using actual Internet experiments. Subsequently, we plan to study the virtual coupling schemes for more than two users at a lower packet transmission rates using the network emulator.

ACKNOWLEDGMENT

We would like to thank Oliver Tonnet from SSSUP for hosting the packet reflector network. Funding from Boeing, US Army, Medical Research and Materials Command, grant number DAMD17-1-0202 and TATRC, Telemedicine and Advanced Technology Research Center.

REFERENCES

[1] C. Gunn, M. Hutchins, D. Stevenson, M. Adcok, and P. Youngblood, "Using collaborative haptics in remote surgical training," in *Proc.First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems(WHC'05)*, 2005, pp. 481–482.

[2] W. McNeely, K. Puterbaugh, and J. Troy, "Six degree-of-freedom haptic rendering using voxel sampling," in *Proc. ACM SIGGRAPH*, 1999, pp. 401–408.

[3] C.B.Zilles and J. Salisbury, "A constraint-based god-object method for haptic display," in *Proc.IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh,PA, 1995, pp. 146–151.

[4] G. Sankaranarayanan and B. Hannaford, "Virtual coupling schemes for position coherency in networked haptic environments," in *Proc. BioRob 2006. The First IEEE/RAS-EMBS International Conference On Biomedical Robotics And Biomechatronics.*, Pisa,Italy, 2006, pp. 853–858.

[5] G. Sankaranarayanan and B. Hannaford, "Experimental internet haptic collaboration using virtual coupling schemes," in *review for IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, California, USA, 2007.

[6] G. Sankaranarayanan and B. Hannaford, "Measurement and emulation of time varying packet delay with applications to networked haptic virtual environments," in *proceedings of ROBOCOMM, the first International Conference on Robot Communication and Coordination*, Athens, Greece, 2007.

[7] H.R.Choi, B.H.CHoi, and S.M.Ryew, "Haptic display in the virtual collaborative workspace shared by multiple users," in *Proc. IEEE International Workshop on Robot and Human Communication(RO-MAN'97)*, 1997, pp. 478–483.

[8] K. Hikichi, H. Morino, Y. Yasuda, I. Arimoto, and K. Sezaki, "The evaluation of adaptation control for haptics collaboration over the internet," in *Proc. IEEE Communications Quality and Reliability(CQR) International Workshop*, May 2002.

[9] Y. Ishibashi, T. Kanbara, T. Hasegawa, and S. Tasaka, "Traffic control of haptic media in networked virtual environments," in *Proc. IEEE Workshop on Knowledge Media Networking*, 2002, pp. 11–16.

[10] Y. Ishibashi, T. Hasegawa, and S. Tasaka, "Group synchronization control for haptic media in networked virtual environments," in *Proc. IEEE International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems(HAPTICS'04)*, 2004, pp. 106–113.

[11] M. Y. Sung, Y. Yoo, K. Jun, N.-J. Kim, and J. Chae, "Experiments for a collaborative haptic virtual reality," in *Proc. 16th International Conference on workshop on Artificial Reality and Telexistence-ICAT'06*, Washington, DC, USA, 2006, pp. 174–179.

[12] J. P. Hespanha, M. McLaughlin, G. S. Sukhatme, M. Akbarian, R. Garg, and W. Zhu, "Haptic collaboration over the internet," in *Proc.Fifth PHANToM Users' Group Workshop(PUG00)*, Aspen,CO, 2000, pp. 9–13.

[13] P. Buttolo, R. Oboe, and B. Hannaford, "Architectures for shared haptic virtual environments," *Computer and Graphics*, vol. 21, no. 4, pp. 478–483, 2000.

[14] J. Kim, H. Kim, H. K. Tay, M. Muniyandi, M. A. Srinivasan, J. Jordan, J. mortensen, M. oliveira, and M. Slater, "Transatlantic touch: A study of haptic collaboration over long distance," *PRESENCE*, vol. 13, no. 3, pp. 328–337, 2004.

[15] J. Cheong, S.-I. Niculescu, A. Annaswamy, and M. A. Srinivasan, "Motion synchronization in virtual environments with shared haptics and large time delays," in *WHC '05: Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 277–282.

[16] M. Fotoohi, S. Sirouspour, and D. Capson, "Multi-rate control architectures for dextrous haptic rendering in cooperative virtual environments," in *Proc. IEEE Conference on Decision and Control*, San Diego, CA, USA, 2006, pp. 4478–4483.

[17] M. N. Masahiro Ishii and M. Sato, "Networked spidar: A networked virtual environment with visual, auditory, and haptic interactions," *PRESENCE*, vol. 3, no. 4, pp. 351–359, 1994.

[18] C. Basdogan, C. hao Ho, M. A. Srinivasan, and M. Slater, "An experimental study on the role of touch in shared virtual environments," *ACM Transactions on Computer-Human Interaction*, vol. 7, no. 4, pp. 443–460, 2000.

[19] E.-L. Sallnas, K. Rassmus-Grohn, and C. Sjostrom, "Supporting presence in collaborative environments by haptic force feedback," *ACM Transactions on Computer-Human Interaction*, vol. 7, no. 4, pp. 461–476, 2000.

[20] M. O. Alhalabi, S. Horiguchi, and S. Kunifuji, "An experimental study on the effects of network delay in cooperative shared haptic virtual environment," *Computer and Graphics*, vol. 27, no. 2, pp. 205–213, 2003.

[21] J.-C. Bolot, "End-to-end packet delay and loss behavior in the internet," in *SIGCOMM '93: Conference proceedings on Communications architectures, protocols and applications*. New York, NY, USA: ACM Press, 1993, pp. 289–298.

[22] V. Paxson, "End-to-end internet packet dynamics," *IEEE/ACM Trans. Netw.*, vol. 7, no. 3, pp. 277–292, 1999.

[23] Y. Ishibashi and S. Tasaka, "A comparative survey of sychronization algorithms for continuous media in network environments," in *Proc. IEEE Local Computer Networks LCN 2000*, Tampa,Florida, 2000, pp. 337–348.

[24] R. J. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," *IEEE Trans. Automat. Contr.*, vol. 34, no. 5, pp. 494–501, May 1989.

[25] M. Carson and D. Santay, "Nist net: a linux-based network emulation tool," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 3, pp. 111–126, 2003.