12th 2008 IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications

Evaluating ALPHAN with Multi-User Collaboration

Hussein Al Osman, Mohamad Eid, and Abdulmotaleb El Saddik
Multimedia Communications Research Laboratory - MCRLab
School of Information Technology and Engineering - University of Ottawa
Ottawa, Ontario, K1N 6N5, Canada

ABSTRACT

In our previous work we introduced a novel application layer protocol, named ALPHAN, for haptic data communication. In this paper, we present a thorough evaluation of the protocol using a multi-user collaborative haptic application. The benchmark application consists of a simple game where three users attempt to lift a 3D triangular shape and place it in a triangular hole. The performance metrics and the test bed of the protocol evaluation are also discussed. It is found that a delay of 150 ms or higher caused the participating users not even to feel the existence of each other. Also a comparison between the two users and three users scenarios is considered. Finally, we comment on our findings and provide directions for prospective research.

KEYWORDS: Tele-Haptics, networking protocol, XML descriptions, collaborative haptics.

1. INTRODUCTION

The term haptics, originating from the Greek language, refers in general to the study of touch behavior. In recent years, the term has been associated mostly with technology that interfaces users to various computing devices through force feedback. Such haptic interfaces give users the ability to sense and manipulate virtual environments or remote real environments through touch [1].

Recently, there has been a trend towards introducing collaboration to the haptic paradigm. The aim is to allow users in geographically distant locations to collaborate, through the sense of touch, to achieve a common goal. Examples of such applications include graphical user interfaces (GUI's) [2], distributing training or Tele-mentoring [3], scientific data visualization [4], entertainment and gaming [5], computer-mediated social interaction [6], visual art and museums [7], surgical simulations and rehabilitation [8-9], among others. Such field of research is referred to as collaborative haptic audio visual environments (C-HAVE) in this paper.

Typically, C-HAVE application developers are faced with a number of challenges at both the application level and the networking level. Firstly, at the application level, consistency assurance, access control, transparency, and stability are all common issues that have to be dealt with. On the communication level, Quality of Service (QoS) parameters such as the network latency, jitter, packet loss, scalability, and compression are key aspects that have been investigated and researched. A unique feature that characterizes haptic data is the need for simultaneous and interactive input and output with the haptic device at extremely high update rate (1 KHz), which is by far below the typical latency for the Internet. Ideally, these updates are transferred over the network at the same rate they are generated. On the other hand, these updates are characterized by their small size.

This paper evaluates the performance of the ALPAHN protocol when it comes to supporting multi-user collaborative haptic virtual environments. The remainder of the paper is organized as follows: section 2 summarizes related and previous work related to multi-user collaborative haptic applications and networking. Section 3 briefly introduces ALPHAN. In section 4, we introduce our multi-user haptic benchmarking application and present the results we collected. Section 5 states our conclusion and provides further insight into our future work.

2. RELATED WORK

There have been several attempts to study how the network conditions affect user interaction and cause instability in collaborative scenarios [10, 11, 12, 13]. It is worth mentioning that most of the studies have been focused on only network delays, because they are the main cause of instability. In general, delays variance can severely affect cooperative task on the same object by different users and the user-user interaction.

In a two-user manipulation of shared virtual objects over a network but without force feedback, it was shown how jitter has a great impact on coordination performance and task difficulty perception [14]. In [15], the effects of delay and jitter were studied during a cooperative task with no force feedback. Two types of interaction were considered: predicting other's movements and coordinating actions. The results showed how even small delays can affect time completion time and how jitter has a greater impact on predicting other's actions.

Wang and his colleagues [16] describe an experiment where two users were asked to move the MPB Device [17] as if they were shaking hands (mutual touch). During this experiment, both users were in Canada, a user in Ottawa and the other in Kitchener. The end-to-end delay ranged from 30 to 50 ms, but researchers artificially added another 100 ms. Two types of experiments were carried out: without any time-delay compensation or with a compensator. Without any compensation, the system was highly unstable in terms of force feedback. With the time-

1550-6525/08 $25.00 © 2008 IEEE
DOI 10.1109/DS-RT.2008.23

delay compensation technique, in turn, the system provides better transmission of forces, resulting in better performance. This technique consists of a combination of predictive and filtering techniques to compensate for network delay. They concluded that not only is haptic feedback affected, but also visual feedback when increasing network delay and as a result, task performance is degraded.

In a later research, the effects of varying simulated delay on a cooperative task has been studied and evaluated in [10]. During this experiment, the same time-delay compensator was used. In presence of large delays and visual-haptic feedback, virtual scene divergences were found at each user. This suggested the need of further research on predictive filtering to achieve a trade-off between visual and haptic feedback.

As per haptic data communication, several communication protocols/mechanisms have been tested and evaluated including TCP and UDP, Synchronous Collaboration Transport Protocol (SCTP), Smoothed SCTP, Light TCP [18]. First, TCP and UDP have been shown not efficient to use for haptic data communication. The SCTP sends "normal messages" unreliably whereas key messages are sent reliably using sequence numbers. Smoothed SCTP is heavily based on SCTP, yet it provides a mechanism for jitter smoothing. The Light TCP is inspired from TCP yet it supports the notion of key and non-key updates. To the best of our knowledge, none of these protocols were tested with more than two users haptically collaborating with each other. The contribution of our work is to uncover the effects of network conditions in the case of multiple users (three users).

3. The ALPHAN Protocol

In general, a communication protocol serves two main purposes. Firstly, a protocol permits for the standardization of the communication between participating network end points. Secondly, a protocol allows application developers to avoid the repetitive implementation of the same functionality in similar applications.

The ALPHAN protocol (Application Layer Protocol for HAptic Networking) [19] is designed with the aforementioned two objectives in mind by providing a general framework through which C-HAVE applications can communicate. The protocol is strategically placed on the application layer, which allows it to be easily customizable in order to meet the different application requirements.

ALPHAN operates on top of the UDP since the latter does not impose any reliability or flow control schemes that do not fit the C-HAVE applications requirements. Additionally, ALPHAN supports the notion of key updates that is widely supported by most of the haptic communication transport layer protocols, by implementing an application layer reliability mechanism that is only applied to such updates, while "normal updates" remain unaffected. ALPHAN also makes use of the Multiple Buffering (MB) scheme. In this scheme, every object in the C-HAVE environment is attributed a sending buffer. The multi-buffering concept permits the decoupling of update transmission for different objects, In case they need to be prioritized based on user and/or application preferences.

4. Application Implementation

The purpose of this study is to examine the performance of the ALPHAN protocol for C-HAVE applications that require tight collaboration between multiple users. Our previous work in [20] studied the behavior of ALPAHN for collaborative applications that involve two players. In this study, the performance of ALPHAN is examined for a collaborative application that involves three players.

The benchmark application consists of a simple game where three users attempt to lift a 3D triangular shape and place it in a triangular hole (Figure 1). The players' performance is evaluated according to the time it takes to perform the task. The application is especially useful for assessing the effect of network impairments on stability and transparency since the users will be continuously exerting forces on each other.

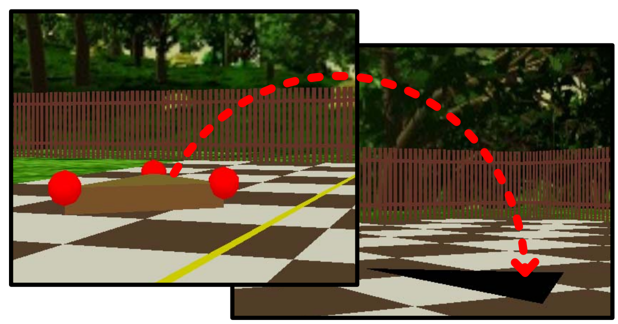

Figure 1 – Benchmark application snapshot

Based on ALPHAN's specifications presented in [19], we have developed a code library that implements the core functionality of the protocol. The library is developed in C++. Figure 2 shows a high level view of the library architecture. The Protocol Manager Module is at the basis of the architecture and most other modules communicate with it. The C-HAVE application itself makes use of the OpenGL and OpenHaptics code libraries. It interacts only with the Protocol Manager module for all its communication needs.

The 3D triangle has three attachment points. As soon as a player touches an attachment point, her/his proxy is affixed to it. The first player to attach to the triangle is considered the owner of the triangle. The triangle cannot be moved nor is any position data exchanged unless all three players have attached their proxies to it. The owner of the triangle continuously receives data from the other two players that denotes the respective position of their proxies. After gathering the proxy position data from the other players, the owner applies his own proxy position to the simulation algorithm in order to produce a common state of the triangle based on the positions of the three users. The resulting state of the triangle is sent back to the other players.

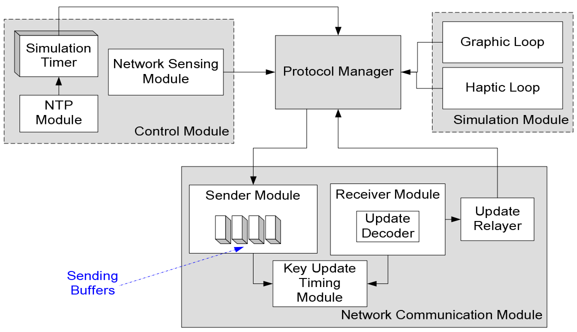

Figure 2 - High level view of the library architecture

The position data is used to calculate forces and therefore it should be sent at the haptic loop rate. Every haptic loop both the independent and dependent forces are calculated. The independent forces are responsible for simulating the interaction of the triangle with the environment (i.e. the triangle's weight, touching the ground, inserting the triangle into the hole). These forces are independently calculated at each workstation and do not require data transfer. The dependent forces simulate the forces of interaction exerted by the players on each other. They are calculated according to the positions of the participating players which are constantly exchanged at the haptic loop rate. Dependent forces are calculated as follows:

1. Calculate CT: the center of gravity of the triangle to be lifted
2. Calculate CP: the center of gravity of the triangle formed by the three proxies
3. Translate the triangle to be lifted so that CT=CP
4. On each participating workstation, calculate the distance vector *Di* between each proxy and its respective attachment point (Figure 3).
5. Calculate the dependent force using the following formula: *Fi*=*D**stiffness
6. Apply a low pass filter on the dependent forces being calculated in order to reduce instabilities.

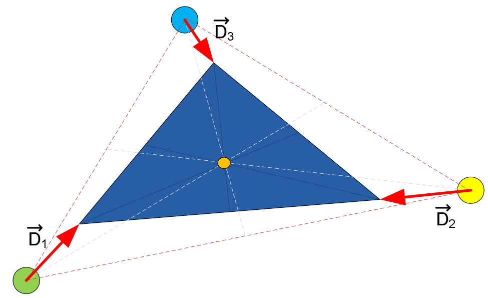

Figure 3 - Dependent Force Calculation

4.1 FRAMING

For this application, two types of updates are used: "normal update" and "key updates". Normal updates are sent unreliably while key updates are sent reliably. In order to conserve bandwidth, several normal updates are buffered and sent in one packet. This method also reduces the overhead created by the addition of a header for each update. Buffering the correct amount of updates is a critical issue. On the one hand, buffering too many updates can severely decrease the responsiveness of the application which can result in instability. Also, it is highly discouraged to produce packets larger than the Maximum Transmission Unit (MTU) since this would result in fragmentation on the Network Layer. In the case of fragmentation, if one fragment is lost, the whole packet is discarded. On the other hand, inadequate buffering can result in excessively consuming more bandwidth unnecessarily. Key updates are not buffered. They are often chosen using heuristic methods that depend on the nature of the application. In this application, the key updates are chosen at points, on the trajectory curve of the haptic proxy, where the slope evaluates to either zero or infinity, since this would indicate a major change in movement and thus must be carefully conveyed to the other users. Figure 4 shows an example of a possible trajectory for a haptic device attached to one end of the board. Key updates are also used to communicate other important events when a user attaches to a triangle or when the target is reached.

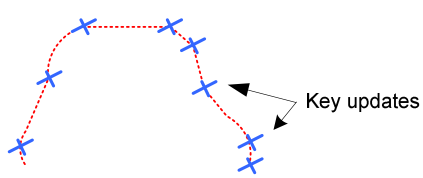

Figure 4 - Possible haptic device trajectory

4.2 MULTIPLE BUFFERING SCHEME

The concept of multiple buffering introduced in [19] is used in this simulation. It stipulates that each object in the environment is attributed a sending buffer [19]. Consequently, it is possible to attribute a priority for each buffer. This priority corresponds to the importance of the

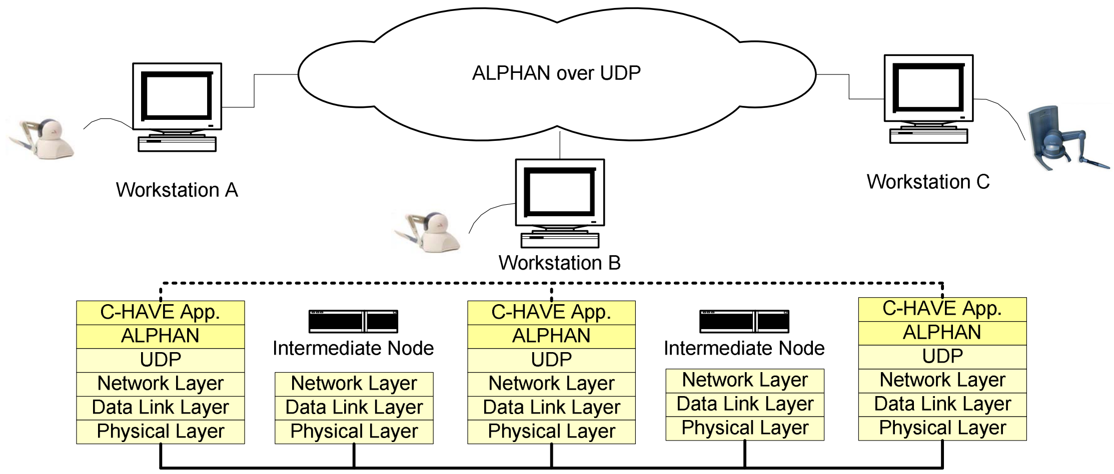

Figure 5- Experimental setup

object in the environment. Buffers with higher priorities have precedence when it comes to sending data over the network. For the purpose of this simulation, each user has two sending buffers. In the case of the owner, the first sending buffer is used to hold the data that describes the state of the triangle. This buffer is given the highest priority. The second buffer holds general event data (attachment, target reached, collision with ground, etc) which are sent at a much lower rate and have a lower priority. The other two users have a buffer to hold data representing their haptic proxies' locations while the second buffer holds event data.

5. EXPERIMENT

5.1 SETUP

Three Pentium 4 PCs with 2 GB of RAM and 100 Mbs Ethernet cards were used for the experiment. The haptic devices consisted of two Phantom Omnis and one Phantom Desktop, all developed and marketed by SensAble Devices, Inc. A snapshot of the experimental setup is shown in Figure 5. Notice that the ALPHAN protocol is implemented at the application layer of the network protocol stack. The application resides on top of ALPHAN.

The experiment was conducted on an Ethernet Local Area Network. Network disturbances such as delay and jitter were simulated using a software tool we developed for this experiment. The software tool intercepts incoming packets, buffers them and forwards them to the application after T ms; T is calculated using equation 1 where R is a random number between 0 and 1.

$$T = \text{Delay} + (\text{Jitter} \times R) \qquad (1)$$

In order to make use of the jitter smoothing algorithm (local lag), the clocks of both workstations were synchronized. For this purpose, a Network Time Protocol (NTP) server was used. Both workstations maintained a connection with the NTP server in order to synchronize their clocks. The clock synchronization precision obtained for this experiment was comfortably within one millisecond.

Fifteen subjects (forming five teams of three members each) took part in the experiment; all of them were undergraduate students from the School of Information Technology and Engineering, University of Ottawa. No particular reward was given to them for their collaboration. Prior to the start of recording the experiment sessions, we asked the teams to perform "rehearsal" sessions just to eliminate any inconsistencies in the performance due to a lack of experience in using haptic devices. During the simulation, the users were neither allowed to communicate verbally nor to see each other.

5.2 EVALUATION CRITERIA

The users' performance is calculated according to the time it takes them to complete the simulation. The following metrics were collected during and after the simulation:

- Time to complete: the amount of time it takes to perform the task.
- Transparency: the extent to which a user feels the other users' presence.
- Stability: the user's perception of the stability and realism of the experience.

Both the transparency and stability metrics are subjective. They are collected with the help of a survey questionnaire given to the users at the end of the simulation.

5.3 Results

The first experiment consisted of conducting the simulation under different delay conditions without injecting any jitter into the network. Each time the experiment was performed by a team, the time to complete metric was logged. The experiment was conducted twice by each team; first, with multiple buffering enabled and then with multiple buffering disabled. The results collected from all teams were averaged to produce a single time to complete value for each delay condition (Figure 6). As it can also be deduced from the results, enabling multiple buffering does not result in a dramatic increase in the users' performance. The multiple-buffering scheme has been proven to increase performance especially in environments that include a multitude of objects with varying priorities; each object is usually allocated an independent sending buffer. Our environment only includes one object (the shared triangle).

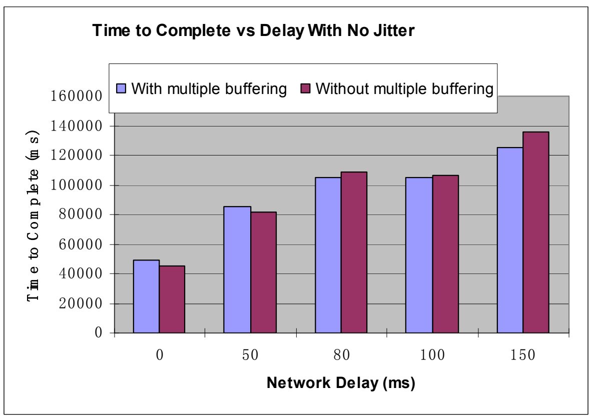

Figure 6 – Time to complete task vs. Network delay

Furthermore, the users were handed a questionnaire to assess the stability and transparency of the simulation on a scale from 0 to 10. Both parameters were well explained to the users. These were also averaged to produce one stability and transparency value for each delay condition (shown in Figure 7). As expected, the performance deteriorates with the injection of more delay into the simulation. This is a direct result of the decrease in the stability and transparency of the simulation. Figure 7 also shows that the stability of the simulation, as perceived by the users was well maintained as the delay increased, as opposed to the transparency that dropped significantly. Some users reported almost not feeling the presence of the other users for delays equal to or larger than 150 ms.

The second experiment concentrated on the evaluation of the effect of jitter on multi-user collaborative haptic applications. The delay was set to a constant value of 50 ms (a typical delay in non-dedicated networks), while the jitter varied. Figure 8 shows these results. Enabling the local lag algorithm has a dramatic effect on the performance of the users as the jitter increase. The local lag algorithm has eliminated the jitter effect. This decreases any instability caused by jitter.

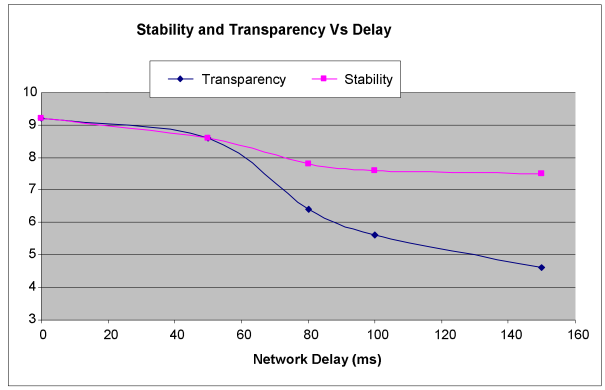

Figure 7- Time to complete task vs. Transparency and Stability

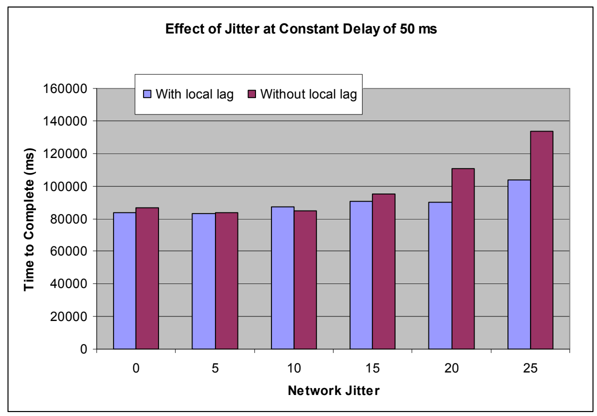

Figure 8- Time to complete task vs. Network jitter

The third experiment consisted of performing the first experiment again, only this time, the application was modified to support only two users. Figure 9 shows a comparison between the "time to complete" values obtained when the experiment was performed with the participation of two and three users respectively. The simulation with two users shows an improved performance. This shows that the effect of network impairments is amplified with the addition of more users. Nonetheless, the improvement in performance can not only be attributed to network factors. In fact, it is much easier to conduct the simulation with two participants rather that three; the users have to collaborate with only another users, which results in less contradicting decisions when it comes to choosing the exact path to be taken to reach the goal. This is obvious since the simulation with two participants produced better results (smaller "time to complete") even when the network delay was set to 0 ms.

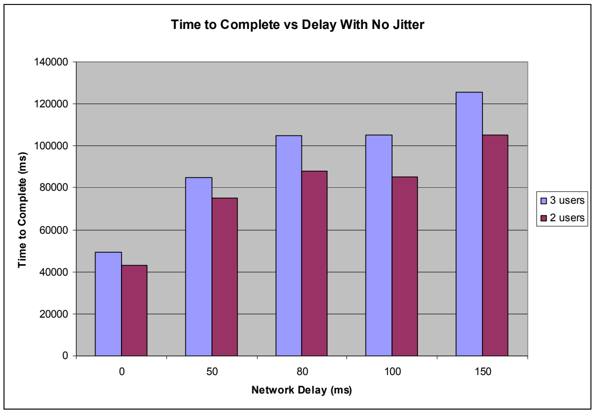

Figure 9- Time to complete task vs. Network delay

6. CONCLUSION

The current paper presented the evaluation of ALPHAN protocol as an application layer protocol for haptic collaboration in the case of three users. A simple game application where three users attempt to lift a 3D triangular shape and place it in a triangular hole has been developed. It is shown that when the delay exceeds 150 ms, the users were not even able to feel the existence of each other. Furthermore, the three users' game imposed significant network load compared to that for two users. This is because all the three users should feel each other and this causes the update messages to increase significantly.

In our future work, we plan to consider more complex scenarios involving more than three users. Also, environments with more than one object will be considered to show the power of the multiple-buffering scheme. Finally, applications from different fields (other than gaming, such as medical and/or Tele-mentoring scenarios) will be considered for ALPHAN performance. Such studies will experimentally prove the ability of ALPHAN to adapt to various haptic applications with different networking requirements

REFERENCES

[1] B. Chebbi, D. Lazaroff, F. Bogsany, P. X. Liu, N. Liya, and M. Rossi, "Design and implementation of a collaborative virtual haptic surgical training system", IEEE Int. Conf. on Mechatronics and Automation. V. 1, Page(s): 315 – 320, 2005.

[2] G. Nikolakis, I. Tsampoulatidis, D. Tzovaras, and M. G. Strintzis, "Haptic Browser: A Haptic Environment to Access HTML Pages", Informatics and Telematics Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, 2004.

[3] K. Hikichi, Y. Yasuda, A. Fukuda, and K. Sezaki, "The effect of network delay on remote calligraphic teaching with haptic interfaces", In proceedings of the 5th Workshop on Network & System Support for Games (NETGAMES), 2006.

[4] L. Durbeck, N. Macias, D. Weinstein, C. Johnson, and J. Hollerbach, "SCIRun/Haptic Display for Flow Fields", In Proceedings of the Third PHANTOM Users Group Workshop, No. 624, pp. 71-75, 1998.

[5] D. Morris and N. Joshi, "Haptic battle pong", In Experimental Gameplay Workshop, Game Developers Conference, 2004.

[6] L. Bonanni, C. Vaucelle, and J. Lieberman, "TapTap: a haptic wearable for asynchronous distributed touch therapy", CHI 2006, 2006.

[7] M. Bergamasco, A. Frisoli, and F. Barbagli, "Haptics Technologies and Cultural Heritage Applications", Proceedings of the Computer Animation (CA), 25-32, 2002.

[8] C. Basdogan, S. De, J. Kim, M. Manivannan, H. Kim, and M.A. Srinivasan, "Haptics in minimally invasive surgical simulation and training", IEEE Computer Graphics and Application, Volume 24, Issue 2, Page(s):56 – 64, 2004.

[9] U. Mali, and M. Munih, "HIFE-Haptic Interface for Finger Exercise", IEEE/ASME Transactions on Mechatronics, Vol. 11, pp. 93-102, 2006.

[10] R.S. Allison, J.E. Zacher, D. Wand and J. SHU, "Effects of network delay on a collaborative motor task with telehaptic and televisual feedback", In ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry, pp. 375–381, 2004.

[11] R.T. Souayed, D. Gaiti, W. Yu, G. Dodds, and A. Marshall, "Experimental Study of Haptic Interaction in Distributed Virtual Environments", Proceedings of EuroHaptics, Springer-Verlag, Munich, pp. 260-266, 2004.

[12] C. Gunn, "Collaborative virtual sculpting with haptic feedback", Virtual Reality, Springer-Verlag Ed., Vol. 10, Issue 2, pp. 73-83, 2006.

[13] C. Gunn, M. Hutchins, and M. Adcock, "Combating latency in haptic collaborative virtual environments", Presence: Teleoperator and Virtual Environments 14(3):313–328, 2005.

[14] K.S. Park and R.V. Kenyon, "Effects of Network Characteristics on Human Performance in a Collaborative Virtual Environment", IEEE International Conference on Virtual Reality, Houston, Texas, 1999.

[15] C. Gutwin, "The effects of network delays on group work in real-time groupware", In European Conference on CSCW, pp. 299–318, 2001.

[16] D. Wang, K. Tuer, M. Rossi, L. Ni, and J. Shu, "The effect of time delays on tele-haptics," 2nd IEEE International Workshop on Haptic, Audio and Visual Environments and their Applications - HAVE, pp. 7-12, 2003.

[17] MPB Technologies: http://www.mpb-technologies.ca.

[18] S. Dodeller, "Transport Layer Protocols for Telehaptics Update Messages", Proceedings of the 22nd Biennial Symposium on Communications, 2004.

[19] H. Al Osman, M. Eid, R. Iglesias, and A. El Saddik, "ALPHAN: Application Layer Protocol for HAptic Networking", Proc. IEEE Workshop on Haptic Audio Visual Environments and their Applications, Ottawa, Canada, 2007.

[20] H. Al Osman, M. Eid, and A. El Saddik, "Evaluating ALPHAN: A Communication Protocol for Haptic Interaction", In Proc. of the 16th Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems, Reno, Nevada, USA, 2008.