An Adaptive Multiplexer for Multi-Modal Data Communication

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

ABSTRACT

Recent trends in multimedia applications strive to incorporate multi-modal media, such as audio, video, graphics, and particularly haptics to enhance the user's quality of experience. However each media type has a particular Quality of Service (QoS) requirement. The efficient use of network resources and the optimal distribution of these resources to all media streams remain an important challenge. This paper presents an adaptive and intelligent multiplexer for multiple input media streams based on the application requirements and the network conditions in a limited-resources network. The multiplexer adapts its multiplexing scheme and guarantees the allocation of minimum resources that are sufficient to make haptics interactions stable. The simulation results show that the proposed multiplexer provides the application immunity to dynamic changes in the network resources, and optimizes the communication of multimedia data based on their corresponding priorities.

KEYWORDS: Multi-modal communication, Tele-haptics, adaptive multiplexer, mathematical modeling.

1. INTRODUCTION

Traditionally, graphic images, 3D models, and audio and video media, define the contents used in a multimedia system. Recently, researchers have made significant progress in multi-modal multimedia systems by incorporating advanced media such as haptics and scent [1]. For instance, the incorporation of the sense of touch gives rise to far more exciting and appealing ways of supporting collaboration, co-presence, and togetherness in these multimedia systems by enabling users to feel each other's presence and the environments in which they are interacting [1].

Meanwhile there has been a trend to enable these multi-modal multimedia applications over a network, and particularly over the Internet. The aim is to allow users in geographically distant locations to collaborate, through the multiple media, to achieve a common goal. Examples of network enabled multi-modal applications include supermedia teleoperation systems [2], multi-modal graphical user interfaces (GUI's) [3], distributing training or Tele-mentoring [4], entertainment and gaming [5], computer-mediated social interaction [6], surgical simulations and rehabilitation [7-8], among others.

From the communication perspective, a multi-modal multimedia application is a combination of multiple input channels representing each media data that should be multiplexed and transmitted over the network. At the receiver side, a de-multiplexer forwards the received media to the appropriate destination channel.

The communication of multi-modal data over the Internet poses several challenges. First, with the limited and dynamically changing network resources, the application should optimally distribute the available resources and provide sufficient resources for each media. Second, not every media has the same contribution to the application quality. For example, haptic media communication is considered of a higher priority level than other media, since the degradation in the network quality (delays/jitters) may result in severe loss of quality and instability of the haptic device. Several haptics data communication researches have shown how even small delays can affect time completion time and how jitter has a greater impact on predicting other's actions [18].

On the other hand, there should be minimum resources that are allocated to each media channel otherwise the quality of that media will be unperceivable. Therefore, a fair distribution of resources, based on the priority level of the input media channel, should be utilized to optimize the usage of network resources.

Second, each media is characterized by varying and sometimes conflicting network requirements. For instance, haptic data is very short in size (usually position or force information) but should be transmitted at high rates (around 1 kHz for force feedback haptic interaction). On the other hand, video data is communicated at only 60 Hz rate whereas the size of the transmitted frame is very large (order of kilobytes). Therefore, each input media channel should be assigned particular QoS requirements based on the media contents.

Third, the network resources are dynamically changing based on the network conditions. Consequently, the application should adapt the allocated resources for each media channel based on the currently available resources. Therefore, an adaptive mechanism should be used to select the most critical data to be transmitted in case the limited resources are not sufficient to serve all the input channels (sending only what is necessary and sufficient). Therefore, the communication mechanism should be network conditions dependent.

978-1-4244-4218-8/09/$25.00

This paper proposes an adaptive statistical multiplexer that is capable of solving the up-mentioned issues by enabling three distinguishing features: First, the multiplexer assigns resources to each input channel based on the respective media via channel prioritization. Second, the application provides the desired QoS parameters using HAML notation [9]. This implies that the communication is setup based on the application requirements and network conditions.

The remainder of the paper is organized as follows: section 2 summarizes related and previous work to multi-modal multimedia data communication. Section 3 proposes the mathematical model of the multiplexer. In section 4, we present the simulation analysis and results and a discussion of interesting findings. Finally, section 5 summarizes the paper contents and provides insight into our future work.

2. RELATED WORK

The problem of communicating multi-modal multimedia data, particularly haptic media, has been the subject of research for the last decade. There have been three directions to solve this problem: (1) improve the control mechanisms to accommodate the unpredictable behavior of the Internet (such as delay compensation [10] and jitter smoothing [11] algorithms), (2) improve the quality of service of the Internet to make it as reliable as a dedicated communication channel, and (3) optimize the volume of data that should be transmitted, and transmit only what is necessary and sufficient to maintain the overall quality of perception for the end users. The research presented in this paper adopts the third approach where the multiplexer chooses which information to send at a time.

Several communication protocols/mechanisms such as Synchronous Collaboration Transport Protocol (SCTP), Smoothed SCTP, Light TCP [12], and STRON [13] classify the application data into categories based on the priority level. For instance, SCTP classifies the transmitted messages into “normal messages” that are sent unreliably and “key messages” that 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 and supports the notion of key and non-key updates. None of these protocols provide the application with the capability of dynamically defining and/or changing the communication based on the transmitted media contents and the available network resources.

The authors in [14] proposed a haptic data transport scheme that reduces the transmission rate by using adaptive aggregated packetization and a priority-based filtering. Another protocol, named ALPHAN [15], uses a similar approach using a multiple buffer scheme to prioritize and optimize media data transfer. Additionally, ALPHAN defines the application requirements using HAML and pass them on to the network protocol. However, the protocol as described in [16] is not adaptable to the network resources and conditions. The proposed multiplexer in this paper adapts resources allocation based on both the application requirements and the available network resources.

The research presented in [17] is probably the most related to what we propose here. It proposes a framework of QoS management for supermedia teleoperation systems, where latency sensitive supermedia streams are encoded using redundancy codecs and transmitted over multiple overlay paths. The overlay routes and encoding redundancy can be dynamically tuned to meet the QoS requirements of the supermedia streams to compensate for network performance degradation. The authors in [19] extended MPEG-4 BInary Format for Scene (BIFS) for haptic information (haptic nodes), where the DMF is responsible for establishing and terminating the communication session. However, the proposed architecture is not adaptive to the network condition and application requirements.

The purpose of this paper is to provide an adaptive multiplexing of multiple input media streams based on the particular application requirements and the network conditions. The concept is inspired from competitive learning neural networks where multiple input streams compete for resources and the ‘winner’ input sends its data through the network. A bias factor (called the conscience factor) is added to the competition model to enforce prioritization. Therefore, higher priority media, such as haptics, will have higher conscience factor values and eventually receives more network resources than for example video media.

3. THE MULTIPLEXING FRAMEWORK

This section introduces the communication framework where the multiplexing scheme will be used. The core part of this section is the mathematical model for the adaptive multiplexer. We present the model and provide insights to its comprising factors/coefficients.

3.1 COMMUNICATION FRAMEWORK

An overview of the proposed communication framework is depicted in Figure 1. Notice that the framework components are initiated and can be functionally modified using HAML descriptions that store the configuration information related to each component. The communication framework consists of the following building blocks:

- *Multiplexer/Demultiplexer:* As every multiplexer does, this component provides a way to interleave data from different input media channels into one serialized bitstream. The multiplexing/demultiplexing overhead and delay should be minimal to minimize the end-to-end delay of the application. Notice that the multiplexer can control the codecs to adapt to network conditions.
- *Network Interface:* The network interface provides the transport layer communication mechanism for multimedia communication (for instance UDP). In case UDP is chosen (for its simplicity and speed), some reliability schemes should be defined at the application layer, within the framework scope.

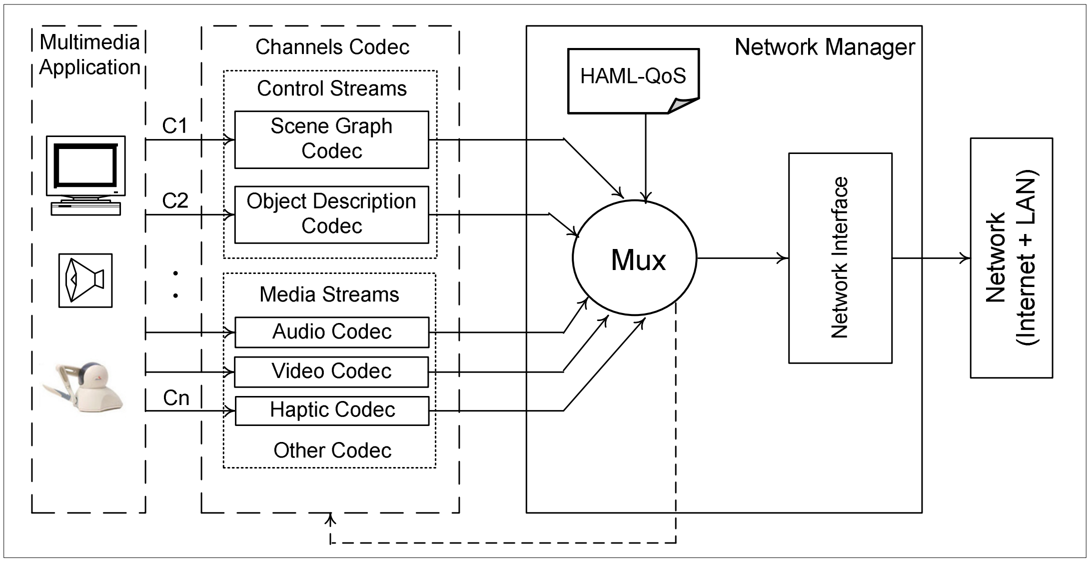

Figure 1 – Overview of the communication framework

- *Channel Codec:* This component defines two types of coders/decoders (codecs): control streams codecs and media streams codecs. Examples of control streams include the scene graph channel and the object description channel. As per the media channels, several audio/video codecs have been developed for real-time communication that can be incorporated in this component. On the other hand, few efforts have been made to develop codecs for haptic data.
- *HAML*-QOS: This is the HAML description schema that defines the quality of service parameters per each input channel (defined as an object). It also defines the default values for the coefficients used to (re)configure the multiplexer. The multiplexer parses this file at startup to know about the number of input channels and their respective network requirements.

3.2 MATHEMATICAL MODEL

3.2.1 GIVEN

Given N input channels represented by two input vectors: the input vector (X) that is defined by the application using HAML and representing the desired QoS requirements for each input channel (Equation 1), and the weight matrix vector (W) that represents the available resources (Equation 2). The objective is to make the distance between $w_i$and $x_i$as close to zero as possible.

$$X = [x_1, x_2 \ldots, x_N]^T \tag{1}$$

$$W = [w_1, w_2, \ldots, w_N]^T \tag{2}$$

Where:

$x_i = \sum_{j=1}^{M} \alpha_j * A_{p_j}$ and $w_i(0) = \frac{x_i \sum_{k=1}^{M} \beta_k * N_{p_k}}{\sum_{i=1}^{N} x_i}$

Where:

M is the number of quality parameters

N is the number of input channels

x_i is the desired probability of selection of the i^{th} channel.

w_i is the calculated probability of selection based on the network conditions (resources) and the history of winning.

α_j is the assigned weight of the j^{th} input QoS parameter

A_{p_j} is the desired value of the j^{th} input QoS parameter

β_k is the weighting factor for the k^{th} network parameter

N_{p_k} is the value assigned to the k^{th} network parameter

Additionally, the proposed model defines a minimum input vector ($X_{\{\min\}}$) that defines the minimum acceptable quality of service requirements per each input channel. When a resource is taken from a particular channel and given to another, the multiplexer checks if this results in a violation to any of the minimum requirements, and if so, it will cancel the resource re-allocation.

3.2.2 PROCEDURE

- *Step 1:* Calculate the intensity $I_j(t)$ for each input (which represents the differences between the available resources and the desired ones). The input channel with minimal intensity wins the competition and eventually a specific data block will be selected by the multiplexer. The equation used for calculating the intensity is shown in equation (3).

$$I_j(t) = D\left(w_j(t), x_j(t)\right) - C_j(t) \tag{3}$$

Where:

$D(w_j, x_j)$ is the Euclidean distance between w_j and x_j.

$C_j(t)$ is called the conscience term. It is used to enable fairness in selection among the N input channels.

- *Step 2:* Update the calculated probability of selection for the winner input. The more a channel wins the competition, the more it becomes likely to win the competition the next time. This is to accelerate the rate at which the w_j value converges to the desired $x_j$value (Equation (4)).

$$w_j(t+1) = w_j(t) + a_j\big(x_j - w_j(t)\big)Z_j + \Delta Q \frac{w_j}{\sum_{k=1}^{N} w_k} \quad (4)$$

Where:

$$Z_j = \begin{cases} 1, & for\ the\ winner \\ -\frac{1}{N-1}, & otherwise \end{cases}$$

a_j is the rate at which w_j converges to the $x_j$value.
ΔQ is the change in the network conditions

- *Step 3:* For each run for the competition, the conscience term (C_j) should be computed as defined in Equation (5).

$$C_j = g(\tfrac{1}{N} + f_j(k) + q_{l_j}.\varphi_j) \quad (5)$$

Where:

g is the fairness gain constant, empirically set to 10.
$f_j(k)$ is the contribution of the j^{th} input channel. The contribution factor is defined as a function of the number of winnings (K_d) and the exponent prioritization coefficient (q_{e_j}), as shown in Equation (6).

$$f_j(k) = 1 - (1 - \tfrac{1}{N})^{R_j(k_j - 1)} \quad (6)$$

$$R_j = \frac{1}{K_{d_j} q_{e_i}} \quad (7)$$

$$\varphi_j = \rho_j P_{size_j} + \sigma_j P_{rate_j} \quad (8)$$

Where:

k_j is the size of data that will be multiplexed from the winner channel to the output stream.
φ_j is the state of the transmit buffer of the j^{th} channel. It is computed, according to Equation (8), as the summation of the number of data units in the buffer (P_{size_j}) multiplied by the weight coefficient (ρ_j) and the rate of data units flow (P_{rate_j}) multiplied by its respective weight coefficient (σ_j).
K_{d_j} is the number of wins for the j^{th} channel.
q_{l_j} is the linear prioritization coefficient.
q_{e_j} is the exponential prioritization coefficient.

3.2.3 MODEL ANALYSIS

The proposed model enables adaptive multiplexing based on the application requirements and network conditions. The channel with minimum intensity wins the competition and thus the corresponding w_j will increase whereas other w_i do not change. This gives that input a higher chance to win the competition the next time. This continues until another channel has the minimum intensity, which makes it win the competition and have its w value increasing, and so on. The selection of an input is made to minimize the differences between X and W.

The conscience factor enforces 'fairness' in the competition among all the input channels. Therefore, as C_j increases, the intensity of the input decreases which gives that channel a better chance to win the competition. The two coefficients q_{l_j} and q_{e_j} can be adjusted at run-time to dynamically change the prioritization of the input media. This is very useful in case a media channel becomes more important than the others at particular events/conditions. For instance, the haptic device position becomes highly important in case the user's avatar is colliding with the virtual world or grabbing an object. If the user is exploring the free space, the haptic data become less important than, for example, audio data. In case $q_{l_j} = q_{e_j} = 1$, no priority control is enforced.

The calibration of the several coefficients that are used in the multiplexer model has a direct effect on the performance of the multiplexer. Therefore, we performed empirical studies and found that the input vectors X and W should be normalized (between 0 and 1), g = 10, a_j =0.2, $q_{e_j} = 0.5$, $k_j = 4$, $\rho_j = 10^{-4}$, $\sigma_j = 5 \times 10^{-3}$. These values have resulted in the intended performance of the multiplexer. The desired performance is demonstrated when the w_j value for all channels converts to the desired x_j almost at the same time and at rates proportional to their respective prioritization values.

One of the model features is that it is statistical multiplexer. Statistical multiplexing results in higher bandwidth utilization and minimum time complexity [20]. However, the major limitation here is that no minimum QoS requirements can be absolutely guaranteed. This becomes an issue particularly for stable haptic interaction. Therefore, this model should be complimented with delay compensation [21] and jitter something [22] algorithms to guarantee stable haptic interaction.

4. MULTIPLEXER SIMULATION AND RESULTS

Three features of the multiplexer are examined: (1) the adaptability of multiplexing to changes in network resources, (2) the time complexity of the multiplexing process, and (3) the scalability of the number of input channels. The simulation is conducted with a Pentium 4 1.83 GHz PCs with 1 Gb RAM

The multiplexer is simulated with four input channels (ordered according to their respective

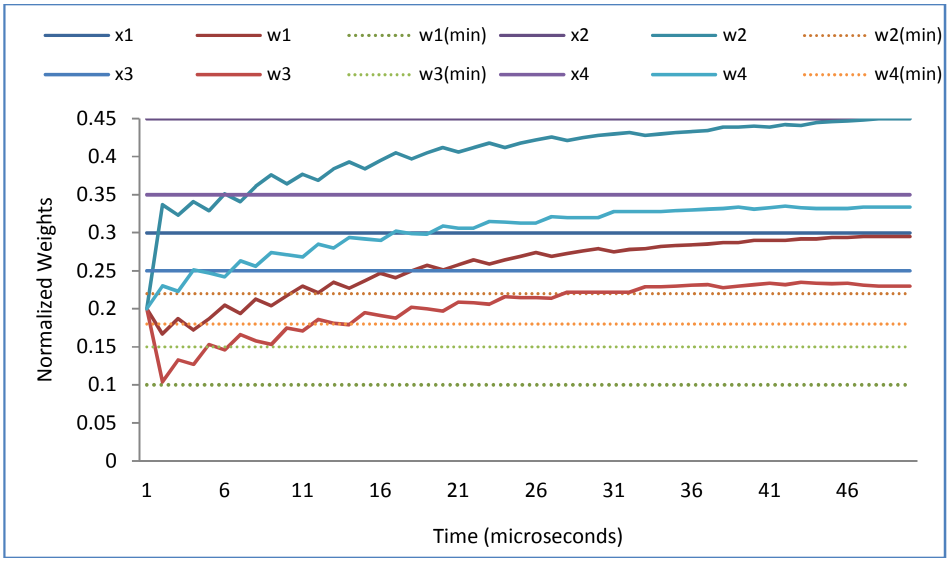

Figure 2 – Simulation with 4 channels (adaptability of the multiplexer)

priorities as used in the simulation): haptic, audio, graphics, and video. We assumed initially equal values for the weighting factors (as an example, we set $w_j(0) = 0.2$).

4.1 ADAPTABILITY

The objective of this test is to evaluate the ability of the multiplexer to adapt to changes in the network conditions (resources). Figure 2 shows how the weighting factors (w_j) are converging to the desired values (x_j) over time. Notice that the optimal performance is achieved when the weighting factors match the desired values, but due to limited network resources, this cannot be achieved for some media, in some cases. Therefore resources are redistributed based on the priority level of the respective media and network resources. Furthermore, the minimum thresholds ($w_j\{min\}$) were never crossed through all the multiplexer operation.

Figure 3 shows the variations in the weighting and desired factors as function of changes in the network conditions (delay changes in this example). When the network delay increases from 30 ms to 100 ms, the recomputed w_j values decrease. However, the multiplexer starts selecting this channel more often and thus increasing its w_j value to adapt to network delay changes.

4.2 TIME COMPLEXITY ANALYSIS

One of the critical factors to examine is the computation time of the multiplexer. This is of particular importance for haptic data communication where computation delays cannot exceed 1 millisecond (to enable a 1 kHz haptic rendering loop). We measured the computation time for per multiplexing cycle and found that the average computation time is 1.12 µs. This result is comfortably within the 1 ms interval necessary for haptic rendering. Therefore, the multiplexer does not cause significant overhead on the overall computation delay.

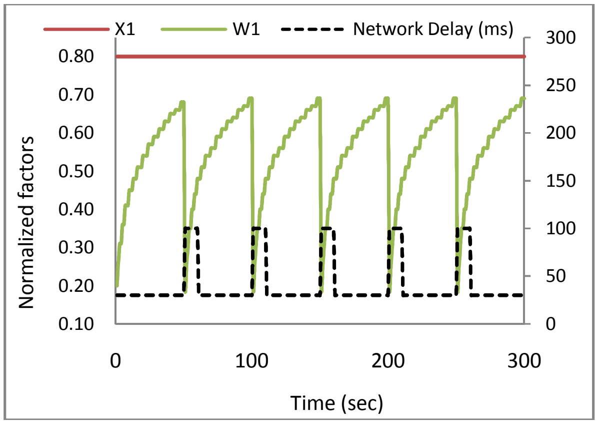

Figure 3 – Adaptability to changes in network delay

5. CONCLUSION

The current paper presents the mathematical modeling and simulation results of a multi-modal multiplexer that dynamically adapts to both the application needs and the network conditions. The simulation results showed that the multiplexer performs with computation time of 1.2 µs, which is comfortably within the 1 ms interval for haptic rendering. Furthermore, the multiplexing scheme is adaptable to the network conditions (Figure 3). Finally the multiplexer is scalable as the number of input channels increase (Figure 4).

In our future work, we plan to add the multiplexer to the ALPHAN protocol, develop a multi-modal application, and test it with the Internet network. Of particular interest, we will test the stability of the haptic communication with different network conditions. Finally, we will conduct a usability testing to quantify the effects of different network conditions on the quality of experience of the end users.

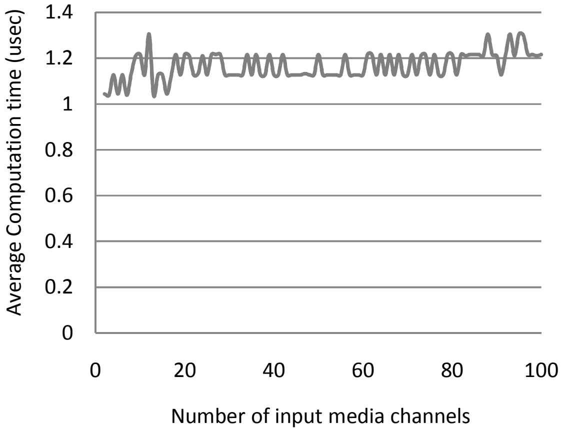

Figure 4 – Scalability of input media channels

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] W.K. Fung, N. Xi, W. Lo, B. Song, Y. Sun, and Y.H. Liu, "Task Driven Dynamic QoS based Bandwidth Allocation for Real-time Teleoperation via the Internet," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2003.

[3] Y. Masunaga, and C. Watanabe, "Design and Implementation of a Multi-modal User Interface of the Virtual World Database System (VWDB)", Proceedings of the 7th int. Conf. on Database Systems For Advanced Applications, 2001.

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

[5] K. PARK, B. BAE, T. KOO, "A haptic device for PC video game application", Elsevier Science, Oxford , vol. 14, no2, pp. 227-235, 2004.

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

[7] 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.

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

[9] M. Eid, A. Alamri, and El Saddik, "MPEG-7 Description of Haptic Applications Using HAML," HAVE'2006, 2006.

[10] S. Lee and J. Kim, "Transparency analysis and delay compensation scheme for haptic-based networked virtual environments", Comp. Comm. Vol. 32, Issue 5, 2009.

[11] O. Wongwirat, S. Ohara, N. Chotikakamthorn, "An Adaptive Buffer Control using Moving Average Smoothing Technique for Haptic Media Synchronization", IEEE Int. Conf. on Systems, Man and Cybernetics, Volume 2, Issue, 12-12, Page(s):1334 – 1340, 2005.

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

[13] Z. Cen, M.W. Mutka, D. Zhu, and N. Xi, "Supermedia Transport for Teleoperations over Overlay Networks", IFIP-TC6 Networking Conference, 2005.

[14] S. Lee, S. Moon, and J. Kim, "A network-adaptive transport scheme for haptic-based collaborative virtual environments", Proceedings of 5th ACM SIGCOMM Workshop on Network and System Support for Games, 2006.

[15] H. Al Osman, M. Eid, R. Iglesias, and A. El Saddik, "ALPHAN: Application Layer Protocol for HAptic Networking", HAVE'2007, Page(s): 96 – 101, 2007.

[16] H. Al Osman, M. Eid, and A. El Saddik, "Evaluating ALPHAN: A Communication Protocol for Haptic Interaction", Symposium on Haptic interfaces for virtual environment and teleoperator systems, Page(s):361 – 366, 2008.

[17] Z. C. Mutka, M.Y. Liu, A. Goradia, and N. Xi, "QoS management of supermedia enhanced teleoperation via overlay networks", IEEE/RSJ International Conference on Intelligent Robots and Systems, page(s): 1630- 1635, 2005.

[18] 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", ACM SIGGRAPH, pp. 375–381, 2004.

[19] J. Zhou, X. Shen, and N.D. Georganas, "Haptic tele-surgery simulation", HAVE'2004, page(s): 99- 104, 2004.

[20] K. Chandra, "Statistical Multiplexing", Wiley Encyclopedia of Telecommunications, 2003.

[21] S. Lee and J. Kim, "Transparency analysis and delay compensation scheme for haptic-based networked virtual environments", Comp. Comm. Vol. 32, Issue 5, 2009.

[22] O. Wongwirat, S. Ohara, N. Chotikakamthorn, "An Adaptive Buffer Control using Moving Average Smoothing Technique for Haptic Media Synchronization", IEEE Int. Conf. on Systems, Man and Cybernetics, Volume 2, Issue, 12-12, Page(s):1334 – 1340, 2005.