A Design and Analysis of a Hybrid Multicast Transport Protocol for the Haptic Virtual Reality Tracheotomy Tele-Surgery Application*

Azzedine Boukerche, Haifa Maamar and Abu Hossain
PARADISE Research Laboratory
SITE, University of Ottawa

Abstract

Nowadays, distributed collaborative virtual environments are used in many scenarios such as tele-surgery, gaming, and industrial training, However several challenging issues remain to be resolved before haptic virtual reality based class of applications become a common place. In this paper, we focus upon a tracheotomy tele-surgery application that is based on closely coupled and highly synchronized haptic tasks that require a high- level of coordination among the participants. We also propose a hybrid protocol that is able to satisfy all the collaborative and haptic virtual environment requirements in general and tracheotomy tele-surgery in particular. We discuss our C-HAVE tracheotomy tele-surgery framework and report on the performance results we have obtained to evaluate our protocol using an extensive set of experiments.

1. Introduction:

Nowadays, virtual environments are used in many scenarios such as tele-surgery, military training, gaming, industrial training, etc [8,9,17,20]. All these applications require that the users collaborate in closely coupled and highly synchronized tasks to manipulate shared objects. By adding the notion of "Haptic" to the CVE, the degree of synchronization became higher [17] and the design of a protocol that satisfies the CVE requirements and that ensure an efficient communication, became urgent. Several researches were completed, and many strategies were designed in order to avoid this problem. However, all of the proposed protocols failed to resolve the synchronization issue and satisfy all the CVE requirements: reliability, minimum delay, scalability and synchronization. In our work, we propose a hybrid protocol[a] that combines four protocols SCTP, SRM, RMTP and SRTP and take advantages of the features of everyone. This protocol has a multicast tree architecture to avoid the congestion and delay problems and uses three modes of transmission to ensure the transport of the data reliably. Our main contributions, added to the combination of the protocols, is to add a timer that allows the receivers to detect a loss or a delayed packet.

2. Previous Work

Several research efforts were done in order to design a protocol that solves the network problems such as the packet delay, loss and the jitter and that satisfies the requirements of CVE and C-HAVE applications: reliability, minimum delay, scalability, and synchronization. All these protocols and techniques were designed in order to provide good collaboration for CVE and C-HAVE applications. But no one was able to meet all the requirements of the CVE. In order to satisfy all these requirements, a hybrid solution is chosen. This new proposed protocol is the combination of four protocols: SRM, SRTP, RMTP and SCTP. The objective of our work is to solve the networking problems and satisfy the CVE requirements. The following; Section 3; gives a detailed description of the hybrid solution highlighting its advantages.

3. The proposed multicast transport protocol

As the applications that use the CVE are growing, the design of an efficient protocol became a necessity. In order to reach this goal, a hybrid solution is chosen. In this section, we shall present our new hybrid protocol, which is a combination of four protocols: SRM, SCTP, RMTP and SRTP. Our scheme takes advantage of the best characteristics of each of the four protocols.

3.1 Architecture of our proposed hybrid protocol

The architecture chosen for this protocol is the RMTP's architecture [7]: the virtual environment is going to be represented by a multicast tree.

[a] This work is partially supported by ORNEC, EAR Award, NSERC, Canada Research Chair program , and Ontario Distinguished Researcher Award.

Users that belong to the same region are grouped together, and one receiver is designed to as DR, the Designated Receiver. DRs are basically chosen by applying the same technique as RMTP, i.e, a DR is designed if it has the largest value of the Time to Live (TTL) when sending the SEND_ACK_TOME packet [7]. The receivers can be grouped into local regions based on their proximity, or their geographical positions.

The proposed protocol supports a multi-level hierarchy of local regions [7]. In this case, the sender receives ACK only from the DRs of the highest level. The DRs of a low level will be the receivers from DRs of high level and senders for the receivers of their own local region [7,11]. This architecture was chosen to minimize the number of messages sent and to avoid the problem of the ACK implosion that the other protocols, such as SCTP, encountered. It is also used as a congestion control plan.

RMTP is designed in a way that it does not have to know how many members a DR has in his local region [7]. This means that there is a good probability that the information would be missed by some receivers: they are either going to miss all the packets, or to miss some of them and to ask then for a re-transmission. In order to avoid this problem and to avoid the unnecessary re-transmissions, we are going to add this feature: Users can join a local group but they have to inform the DR or the sender before by sending an adhesion message. This approach is used by SRM [10]. In this case, we can ensure that the DR has a specific list of all the receivers and that it is going to send the message to all the members in the local region. New users can request for all the packets that they missed. In that case DR sends the packets by applying the immediate transmission request approach [7] that works as follows: when the user finds out that he joined the group in the middle of the transmission, he requests for all the packets that he missed. These packets will be sent using unicast.

In what follows, we will present the Message's format of the proposed protocol that is used in the form of an Interaction Stream, i.e., a sequence of update messages related to a sequence of interactions of user with a shared object is grouped into one Stream [9]. This approach is described above for **SCTP** [8,9]. The interaction stream is composed of two kinds of messages: key-update messages and update messages. The update messages represent the position or the motion of an object. The key update message represents the last update message sent, a final state of an object, or an update message that has to be delivered reliably.

The packet's header carries information that allows knowing if the packet is a key-update message or not [9]. The packet also allows identifying which shared object is being interacted with, the current interaction stream for this object, and the position of the specific update message in the current stream. The notion of Stream Interaction is really important in the CVE. It allows the users to know which object is shared at that moment and to have a stream of all the interactions done on that object.

In our work, the messages are going to be sent dependently of their type. This technique is used in SRTP [1,2,3,4]. Our protocol will use one of the 3 modes of transport defined by SRTP. The message sent is going to be either an "Update message" or a "Key-update message". Update messages are the data that change frequently (e.g., as position). They do not need to be transferred reliably and are going then to be always transferred using the mode 0 via a multicast architecture that operates as pure best-effort services. This kind of data can tolerate errors: in fact, even if a data is missed, the one that is delivered after is going to replace the missed one. This does not affect the performance of the protocol. For the key update message, they need to be transferred reliably. For these messages, we are going to use either the mode 1 or the mode 2. In fact, if the key-update message has to be sent reliably to a group of members, mode 1 is used and the reliability is ensured the IP multicasting technique. If the key-update message has to be delivered to only one member, in the case of a retransmission of a lost packet, then mode 2 is used and a reliable and timely transport is required and this by applying TCP.

In the case of the motion of an object, we are going to add the timer feature. In fact, as the update messages represent the data of an object which is constantly moving, we assume that these messages are sent periodically every a certain period of time. Then if a receiver did not get an update message within that period, it sends a NACK message requesting for the lost packet. Of course, if the object does not move anymore, than a key-update message should have been sent, and the receiver will not wait for another message and will keep its timer to 0. In order to ensure the collaboration, the synchronization and the minimum delay, an end-to-end delay is no more than 100 msec for update messages. This limit was proposed in [8, 9], and is going to be used for the proposed protocol.

3.2. Reliability, Detection and Retransmission of a lost packets:

DIS, CVE and C-HAVE applications require a reliable transport for some packets in certain cases. A reliable protocol is then obligatory for these applications. The reliability of our proposed protocol is applied by the NACK messages. This approach is used by SRM [6,10,14]. The detection of a lost packet is assured by a timer: in fact, the members that are waiting for a message and that do not receive it within the specified period of time, send a NACK message, to their DR, that contains the sequence number of the last packet received. Moreover, if the receiver gets an unexpected packet, he puts the packet received in its buffer and sends a NACK message. The use of the NACK-based approach was chosen in order to avoid the ACK implosion problem.

The re-transmission of the lost packet is the task of the Designated Receiver in the multicast tree. When a receiver finds out that he did not receive a packet, he sends a NACK to his DR to inform him about the lost packet. The DR receives the NACK message containing the sequence number of the missed packet, and he sends (unicast) the lost packet to the right receiver; if there is more than one receiver, than the DR has to multicast the lost packet to the receivers. Since lost packets are recovered by local retransmissions as opposed to retransmissions from the original sender, *end-to-end latency* is significantly reduced, and the *overall throughput* is improved as well.

3.3. Ordering and Scalability

Although some of the protocols, such as SRM, do not satisfy the ordering, the proposed protocol maintains the ordering by the use of the sequence numbers. In fact, every packet delivered contains a sequence number that helps the receiver to order the packet received and to detect a lost packet if there is a gap in the sequence number.

Our work is the combination of four protocols that ensure scalability. The proposed protocol is designed then to be scalable and fast. Every time a receiver wants to join the multicast group, he has to inform all the members by sending an adhesion message. When the members receive this message, they update their information and they send the message SEND_ACK_TOME, so the new member can elect his Designated Receiver.

4. Implementation of the Hybrid Multicast Transport Protocol (HMTP) in Tele-Surgery applications:

The Hybrid Multicast Transport Protocol can be used in many applications based on Collaborative Virtual Environments and using Haptic devices. In this paper, we present a tele-surgery application that has been carried out in the DISCOVER Research Laboratory at the University of Ottawa: The Tracheotomy Tele-Surgery application. The architecture of the Tele-Surgery application is illustrated in Figure 1. Tele-Surgery applications require that the users collaborate in closely coupled and highly synchronized tasks to manipulate a shared object, the patient. By adding Haptic devices to the application, the degree of synchronization becomes higher and the use of an efficient protocol, Hybrid Multicast Transport Protocol HMTP, that ensures a good communication is important. This section is devoted to give a detailed description of the Tele-Surgery application and the results illustrating the failures that occurred when using the HMTP and changing the Network conditions.

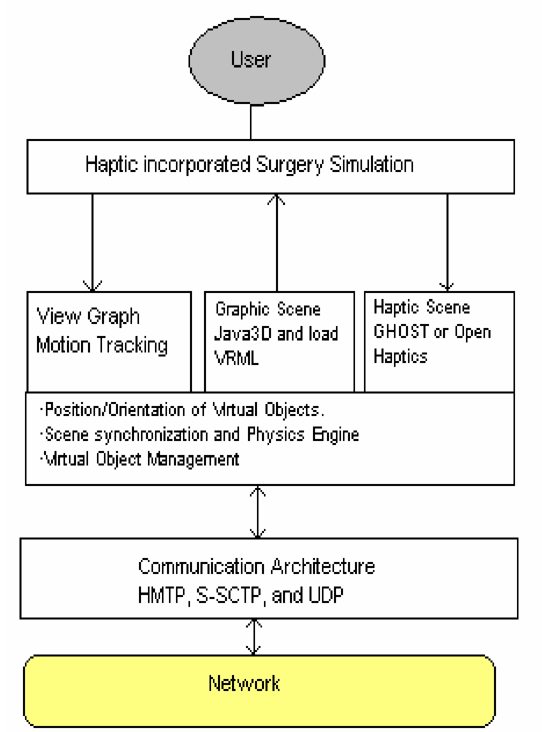


***Figure1.** Haptic Incorporated virtual environment Application Architecture.*

4.1 The Haptic Virtual Reality tracheotomy Tele-Surgery application

The main application being written in Java, the Java Native Interface (JNI) which has been used to communicate with the C++ program using General Haptic Open Software Toolkit (GHOST). The JNI provides "native" methods to call the Microsoft Dynamic Loadable Library (DLL) from Java program. The description of a Java3D universe is the description of a simple universe, used in our implementation to provide a complex viewing architecture, separated from the scene graph. VRML descriptions are loaded in Java3D and handle avatar behaviors.

The Haptic Interface is implemented by JNI and a C++ DLL to the GHOST SDK. It loads a VRML description of the throat in the haptic interface and allows a method that returns the PHANToM position and orientation. This VRML description provides a feeling of touch to the user.

The tracheotomy is an application that aims to simulate a surgical act commonly performed in emergency medicine. This surgery requires very tightly coupled collaboration between members of the surgery team. In fact, the lack of collaboration can lead to fatal errors. In this scenario, we presented a simple tele-surgery application where two surgeons, or trainees, must share tools such as a scalpel, surgical hooks, and a piece of gauze, to cut the skin of a virtual patient's throat, spread it open, and cut inside the underlying muscle layer.

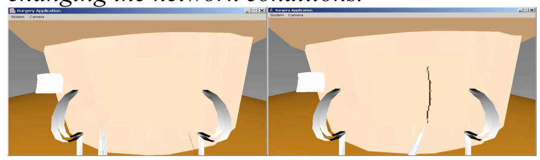
During each simulation session, one surgeon grabs the scalpel and performs a vertical cut while the other surgeon grabs the gauze to remove spilled blood. The first surgeon successively takes both hooks to pull the skin, while the second surgeon performs a horizontal cut on the muscle. Once it has been cut, blood spurts from the muscle. The successive operation is described in figure 2.

The above tele-surgery application was tested between two users, one is acting as the first surgeon and the other acting as the second surgeon; their respective duties were explained. The success of the surgery depended on close collaboration between the remote surgeons. Collaboration failure happens in two ways:

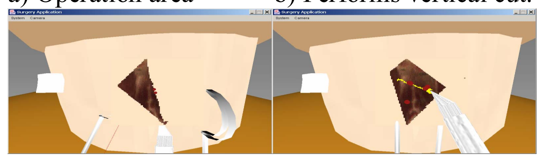
1- The first surgeon cuts the wrong place because he or she does not correctly perceive the position of the surgery, resulting in a great deal of bleeding and the skin being unable to be pulled.
2- The second surgeon cuts the wrong place of the muscle and there is bleeding.

The unsuccessful surgery is depicted in figure 3. Failure of collaboration occurs because of lost updates, jitter, and network delay.

Before commencing performance tests in the presence of packet loss, the application was first tested with no packet loss; no failure was observed over fifty trials. This is a necessary test to ensure that failures, which do occur, are not due to the nature of the application but instead to network delay, jitter, and packet loss. The following section gives the results of the tests performed when changing the network conditions.

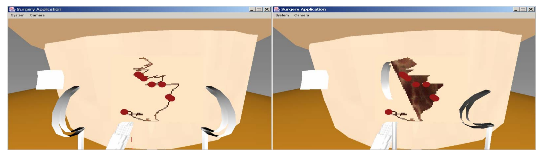


a) Operation area b) Performs vertical cut.

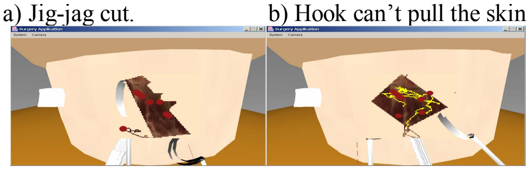


c) Pull the skin. d) Cut on the muscle

Figure 2: Successful collaboration in tele-surgery application.



a) Jig-jag cut. b) Hook can't pull the skin



c) Cut in wrong place d) Jig-jag cut the muscle.

***Figure 3**: Failure of collaboration in tele-surgery application.*

4.2 Results of the Haptic Virtual Reality tracheotomy Tele-Surgery application

A group of four participants were used to conduct this experiment. In case of execution time consideration, there were 5 trials, each involving normal reverse order experiments to avoid the "training effect". For communication, Uniform Datagram Protocol (UDP), which is an unreliable protocol, Smooth SCTP and Hybrid Transport protocol were used. The tests were performed in scenarios in which there was no lag, as well as scenarios in which there were loss rates of 20%, 30%, 50% and 10 to 50%. We performed the same tests again by adding 50 ms delay, with and without 10 ms jitter. These tests were done using three protocols: UDP, S-SCTP, and HMTP.

Collaboration errors are defining in early Section 4.1. 50 trials were processed using UDP, S-SCTP and HMTP to test the tele-surgery application. The figure 4 is showing the test results for the failures that occurred when using these 3 protocols.

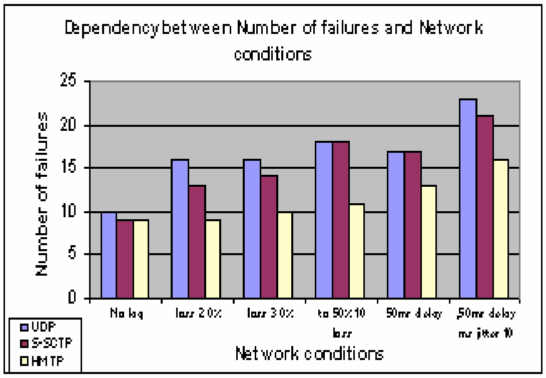


Figure4. *Results of the Tele-Surgery application based on number of failure during trials.*

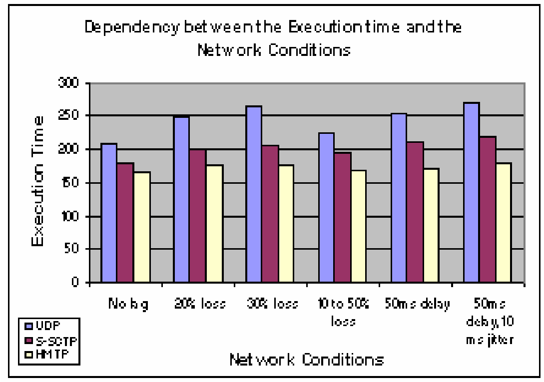


Figure5*. Results of the Tele-Surgery application based on execution time to complete the application.*

The results show that the occurrence of failures is reduced in Hybrid Multicast Transport Protocol-based tele-surgery application as opposed to scenarios with UDP and Smooth-SCTP based tele-surgery model.

A second type of tests was done in order to illustrate which one has the smallest total execution time. The total execution time is defined in our tracheotomy application by the difference between the completion times, i.e. the gauze finishes cleaning the blood spilled after the second cut, and the beginning time i.e. the time at which the scalpel or the gauze starts moving. All the times are considered in seconds. The results are shown in figure 5. As illustrated in figure 5, the execution time is reduced using the Hybrid Multicast Transport Protocol in comparison with UDP and S-SCTP.

Another type of tests was done in order to show which protocol has the least number of failures by changing the network conditions and starting from the worst case (50 ms delay and 10 ms jitter) and going to the best condition which is no lag. We know that this type of test is important to show the efficiency of the protocol. In fact, the results may differ if we start from state Failure instead of a state Success. The results of this type of test are shown in the figure 6.

The results of this type of tests shows that the HMTP has the least number of failures as opposed to UDP and Smooth-SCTP when applied in the tele-surgery model and by starting from the worst network condition.

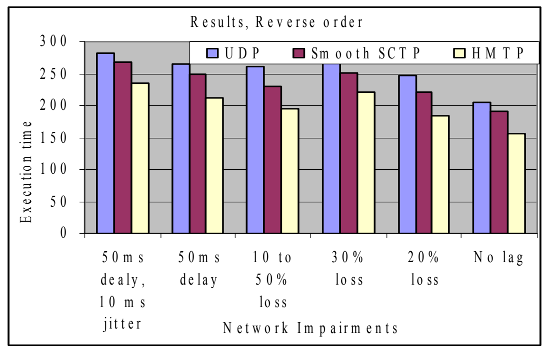


Figure6*. Results of the Tele-Surgery application based on number of failure during trials (Reverse).*

5. Conclusion and Future work

In this paper, we have presented a hybrid solution protocol combining SRTP, SRM, RMTP and SCTP protocols and the advantages of their characteristics. The Hybrid Multicast transport protocol was developed, implemented and tested in The Tracheotomy Tele-Surgery application.

Results of the tests and their comparison with two transport protocols UDP and Smooth-SCTP, show that the occurrence of failures and the execution time are reduced when using HMTP as opposed to UDP and Smooth SCTP. Further research is required to test our proposed protocol in different CVE and C-HAVE applications in order to improve its performance.

References

[1] *J. M. Pullen and V. P. Laviano* .Adding Congestion Control to the Selectively Reliable Transmission Protocol for Large-Scale Distributed Simulation. *Fall 1997 Simulation Interoperability Workshop paper 97F-SIW-018,*

[2] *J. M. Pullen and N. Kakarlamudi*. Performance Issues for the Light-Weight RTI.

IEEE Simulation Interoperability Workshop, Orlando, FL, September 1998

[3] *J. M. Pullen and V. P. Laviano* .A Selectively Reliable Transport Protocol for Distributed interactive Simulation. *Proc. of the 13th Workshop on Standards for the Interoperability of Distributed Simulations*, 1995, *pp95-102*

[4] *D. M. Moen and J. M. Pullen* .A Performance Measurement Approach for the Selectively Reliable Multicast Protocol for Distributed Simulation *Proc. of the Fifth IEEE Workshop on Distributed Simulation and Real-Time Applications,* 2001, pp.30-34.

[7] *S. Paul, K. K. Sabnani, J. C. Lin and S. Bhattacharyya.* Reliable Multicast Transport Protocol. *IEEE Journal on Selected Areas in Communications, April 1997,* pp 407-421

[8] *S. Shirmohammadi and N. D. Georganas* .An End-to-End Communication Architecture for Collaborative Virtual Environments. *Computer Networks Journal, Vol. 35, No. 2, pp. 351-367, February 2001*

[9] *S. Shirmohammadi and N. D. Georganas* Collaborating in 3D Virtual Environments: A Synchronous Architecture. *Proc. IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Knowledge Media Networking Workshop, National Institute of Standards and Technology (NIST), U.S.A., June 2000, pp. 35-42.*

[10] *S. Floyd, V. Jacobson, C. G. Liu, S. McCanne, and Lixia Zhang* .A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. *IEEE/ACM Transactions on Networking, Dec 1997. pp784-803*

[14] *P. Parnes*. The mStar Environment. Scalable Distributed Teamwork using IP Multicast. *Master Thesis, September 1997*

[17] *A. Boukerche, S. Shirmohammadi, A. Hossain*. Prediction Based Decorators for Distributed Collaborative Haptic Virtual Environments. *J. Computer Applications in Technology, Special Issue in Collaborative Multimedia Applications in Technology 2006*