

Simulation and Performance Evaluation of a Distributed Haptic Virtual Environment supported by the CyberMed framework

Paulo Vinícius F. Paiva¹, Liliane S. Machado^{1,3} and Jauvane C. de Oliveira^{2,3}

¹LabTEVE – Federal University of Paraíba (UFPB)
paulovfpiox@gmail.com, liliane@di.ufpb.br

²A.C.I.M.A. – National Laboratory for Scientific Computing (LNCC)
jauvane@acm.org

³National Institute of Science and Technology – Medicine Assisted by Scientific Computing (INCT-MAC)

Abstract— Performance evaluation on Distributed Haptic Virtual Environments (DHVEs) became important to understand the new Internet requirements for supporting multisensorial and real-time collaborative applications. This paper presents the results of simulation and performance analysis of the CyberMed framework. The main goal of this experiment is to evaluate the real conditions of CyberMed when executed over a non-dedicated hybrid network, like the Internet, comparing its results with other similar works found in the literature.

Keywords- collaborative environments; networked virtual environments, network performance

I. INTRODUCTION

Virtual Environments (VE) are three dimensional spaces based on Virtual Reality (VR) which provides users with the sense of being interacting within the virtual world [12]. This property is one of the basic VR concepts also known as the immersion capability, which is the user's illusion of being inside of the VE. Such concept is caused mainly by the high level of realism that such systems can achieve. The immersion sensation can be provided through the use of special devices which explore the multi-sensorial human channels. One such example are the haptic devices which give haptic feedback (sense of touch) to the users [4]. In this context, applications which provide an interface between a haptic device and a VE are known as Haptic Virtual Environments (HVEs). In distributed HVEs (DHVEs) architectures, various users, with their haptic devices, can feel each other actions and interact at the same time through a network [10]. This idea of multiple users interacting in real time to accomplish a task together is part of another important concept inside the VR. Such task is known as collaboration and it is based on the Computer Supported Cooperative Work (CSCW) concepts [3]. Collaboration has many applications, such as collaborative design, training, teaching, software engineering and telepresence [8]. The medicine has also been benefited from collaborative and specialized VR based systems, particularly those designed for teaching and training purposes. Such systems are known

as medical simulators. In a medical training context, these kind of simulators bring some advantages, like the possibility of training new professionals with some cost reduction, the reduction of the need for "guinea pigs", as well as the possibility of remote surgeries and more broadly usage of interactive teaching techniques. Other advantage of the medical simulators is the use of new interaction devices such as haptic devices which enables the user to identify, through the sense of touch, textures and material properties (eg. hardness, softness, viscosity and elasticity) of the objects inside the VE [3,5]. Haptic devices are important, for instance, for VEs in which one can experience the simulation of procedures which involves medical tools such as needles, scalpel or even just the human touch for enabling more realistic diagnosis. Therefore it can be said that collaborative DHVEs allow remote tutoring and sharing of experiences among students who are learning medical procedures.

For this purpose, the CyberMed framework has been created aiming at facilitating VR based medical simulators development process [5]. A collaboration module named CybCollaboration has recently been introduced in the CyberMed framework. However, such module has not been validated considering various network architectures. One way to evaluate the performance of this collaboration module is the use of simulation which can shows the system's behavior based on the calculation of network impairments such as delay, packet loss, jitter, etc. These network impairments are known to degrade the performance of DHVEs and there's some considerations presented in [10] about the damages caused by them. This work [10] reveals that delay desynchronizes the different VE copies spread by network and it also makes the user's haptic device go through a virtual object before it is felt and packet loss can reduce the amount of force felt by the users and it can also cause abrupt force feedback. Thus, the main goal of our experiment is to evaluate the real conditions of CyberMed when executed over a non-dedicated hybrid network, like the Internet and some dedicated intranets. The rest of this work is divided in sections II through V, respectively covering an introduction on the collaborative

aspects of the CyberMed framework, details on the simulation we implemented, results and discussion and finally final discussion and future works, which is followed by acknowledgment and a list of references.

II. COLLABORATION ON CYBERMED FRAMEWORK

CyberMed is an open source framework designed to facilitate VR based medical application development. Like all frameworks, this framework allows programmers to build their systems in a faster and easier manner, also allowing them to extend it with new modules [10, 4]. This framework stands out mainly due to the large variety of resources offered by its modules, which include mono and stereoscopic visualization, interaction through conventional devices (mouse and keyboard) and non-conventional devices (haptic and tracking), interactive deformation of 3D models, collision detection, collaboration and online assessment support. The framework architecture is visible in Figure 1 extracted from [10].

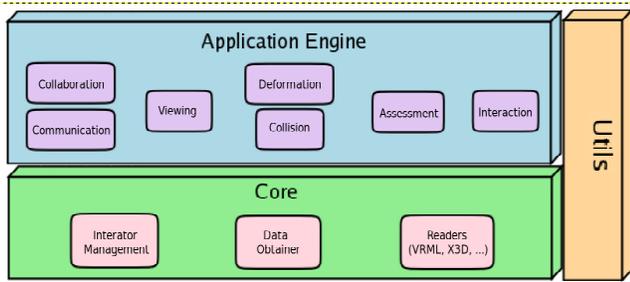


Figure 1. Software architecture of the framework CyberMed.

The network module, provides the network communication to the collaborative module and it simulates both reliable and non-reliable communications. In the reliable communication, the delivery of all packets is guaranteed through acknowledgement messages. The non-reliable communications are frequently used by applications where the speed has major importance compared to reliability of message delivery, like multimedia systems that supports audio and video data [13]. The network module supports both communication ways by two transport protocols: UDP (User Datagram Protocol) and TCP (Transmission Control Protocol). It can also handle the three basic structures of networking communication which are: unicasting, where the message is addressed to just one destination host; broadcasting, where the message is sent through the network to all connected hosts; and finally multicasting, where the messages are sent to different groups of destination hosts, giving the possibility of using flexible communication services, among other advantages. Considering the requirements for supporting collaboration with haptic feedback, some research has shown that DHVEs requires high levels of graphic and haptic update rates, which are at around 30 Hz and 1KHz respectively [6]. The collaboration provided nowadays by the collaboration module, named CybCollaboration, is based on the UDP transport protocol in association with a peer-to-peer

architecture, which provides a reasonable transmission rate for haptic data. The collaboration works without a server acting as a mediator to establish all clients' connections. Such setup avoids time waste. The client that creates the main VE's copy works as a facilitator, who is responsible for collecting and issuing the IP addresses and current position of all clients. The framework has a distributed architecture where copies of all clients scenes are distributed among all users connected to the application. The distribution of the main database copy is performed through a virtual objects distribution model known as active replication [2]. Once observed that just one VE version is executed by all collaboration participants at the same time, it is necessary to control the consistency, making sure that all the versions are in synch. In the active replication, this control is handled by the participants which are responsible for issuing any alteration in their local VE version. It is important to mention that the object manipulation management can be done in two ways named block and free states respectively. The free state allows a collaboration where everyone can modify the object in a sequential way, whereas in the blocked state, the object remains blocked and just the person who has the permission is allowed to change it. The blocked state is used in guided collaborations where just one user, acting as a guide, has the permission to manipulate the objects. Therefore, the CybCollaboration module enables the user to choose among different kinds of collaboration, considering a combination of interaction devices (haptic, mouse or tracking) as a way of manipulative object management (free or blocked). The most trivial way of collaboration allows the free manipulation object management associated with the mouse device. One other option is the guided collaboration where one host is designated as a leader and its haptic guides the remote users movements by their own devices.

III. SIMULATING CYBCOLLABORATION

In order to advance on research about DHVE, it is important to understand existing mechanisms, and create new ones, that enable high levels of Quality of Service (QoS) required by this kind of application. In order to analyze CybCollaboration's performance, it was necessary to simulate it running over various network scenarios (different architectures, bandwidths, etc). The simulation is an important resource to many areas due its possibility of artificial representation and real models analysis with low cost [11]. Network application simulation is a not simple process. It is necessary to pay special attention to input parameters and the methodology employed. For this work, it was performed a representation of our application according to a well-defined methodology which involved four sequential phases. Such phases were: the requirements analysis, the simulator selection, learning how the simulation tool works and finally the CybCollaboration's protocol implementation, simulation and analysis. In the first phase, the basic requirements chosen for implementing the simulation of this application were: transmission rates from the PHANToM Omni haptic device (Sensable Technologies) [15] used in our tests, the transmission protocol and its packet format (they have from 18 to 20

bytes of payload, not including the UDP and Ethernet headers). Also, it was necessary to choose a simulator that could allow the implementation of the network behavior of the CybCollaboration setup. Simulators available in research efforts (e.g. ns-3, OMNeT++, JiST) [12] as well as those frequently used in the industry were observed. After some evaluation of the simulation tools available we chose the OPNET Modeler (OM). OM is a simulator of discreet events [15] which allows the creation of models that properly simulate the network behavior of new architectures, network protocols and alike. The chosen simulation tool also enables the use of Probability Density Functions (PDFs) for producing and setting up different kinds of data traffic patterns such as those common in multimedia systems, as well as typical VR applications. In a DHVE context, the experiment presented in [6] shows one of the OM simulator capabilities by developing a PDF to characterize the data flows generated by a haptic device. In the CybCollaboration' simulation experiment, it was created a node responsible for generating haptic data traffic and for processing all packets according to CybCollaboration's protocol. The node model presented in Figure 2 contains four sub-units called modules and each module has its inner process model for defining its possible states using State Transition Diagrams (STDs).

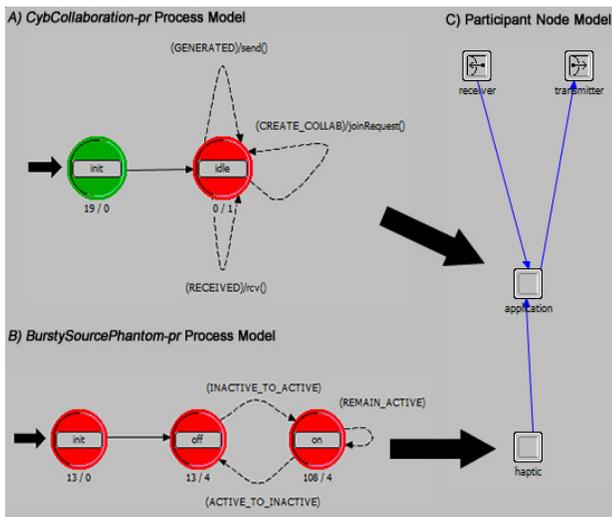


Figure 2. The Node Model and its Process Model's STDs (State Transition Diagrams) used in CybCollaboration' simulation: (a) STD of the collaboration protocol; (b) STD which emulates the behavior of the haptic device; (c) Node Model and its four modules.

The node model, that simulates the behavior of all nodes, has four modules, namely:

- haptic – This module simulates the behavior of the PHANToM Omni device in generating its haptic data. For its modules, it was necessary to define both inactive (off) and active (on) states in which the process may remain. To emulate this function, the BurstySourcePhantom-pr Process Model was developed.

- application – Simulates the collaboration protocol used by CybCollaboration that is responsible for establishing all client's connections, sending, receiving and processing the haptic data packets generated by the haptic module. This module uses the CybCollaboration-pr Process Model for defining the state of the collaboration environments.
- transmitter – Module that represents a network interface for transmitting packets.
- receiver – Module that represents a network interface for receiving the incoming packets.

A. BurstySourcePhantom-pr: Process Model of the haptic module

The BurstySourcePhantom-pr process model is composed by a list of attributes that defines the ideal parameters and its PDFs for the traffic generation. For example: the ON/OFF State Time attribute specifies the amount of time which both "on" and "off" states will remain valid. In this case, the Packet Interarrival Time attributes specifies the PDF arguments that will be used for generating random outcomes for times between successive packet generations in the "on" state [15]. Finally, there is some packet generation attributes that must be defined. Such attributeds include the Packet Format and Packet Size, respectively determining their format and length. Using these parameters, it is possible to simulate the bursty characteristic of outgoing packets from a given haptic device. The distribution types used for transmitting and receiving packets were the Exponential and Poisson, respectively as recommended in the literature [16]. These PDFs are commonly used for emulating communication systems, like network communications, once packet interarrival times (at the receiving nodes) are often random, independent of each other and exponentially distributed [15]. The three main state transitions used on this process are executed when the on/off period times expires, satisfying some conditional pre-defined macros. The algorithm described by this process model can be seen in Figure 3.

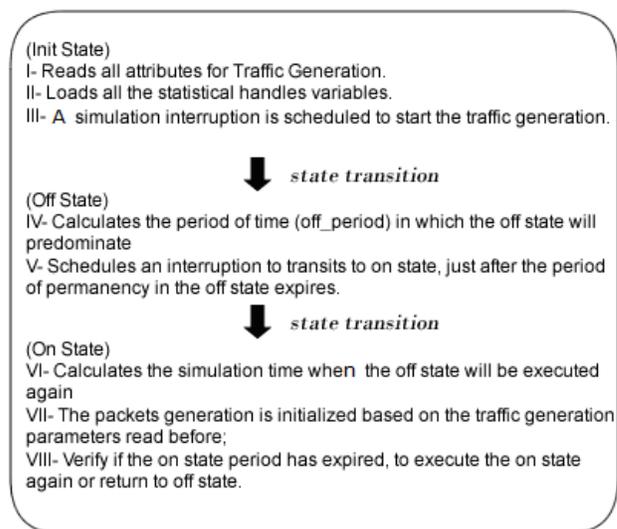


Figure 3. Algorithm used in traffic generation for simulating the haptic device's behavior.

B. CybCollaboration-pr: Process Model of the application module

The CybCollaboration-pr process model is responsible for representing the main states and events used in a collaborative application developed with the CyberMed framework. Therefore, it has basically two main states which are the init and idle state. The first one is used to initialize some parameters such as temporary and statistical variables. It also reads the "Node Address" and "Leader" attributes which are used to identify all hosts on the network and the

host that will act as a leader of the collaboration. The idle state remains in a passive behavior while it waits for interruptions on the simulation's kernel which can be caused by the detection of packets generated by the haptic module or by the arrival of packets in the receiver module. The conditionals macros GENERATED, RECEIVED and CREATE_COLLAB were defined to determine the necessary conditions that must be satisfied for enabling the state's transitions and the execution of the functions which are responsible for sending, receiving and processing all packets. The first macro indicates when packets generated by the haptic module arrive at the application module.

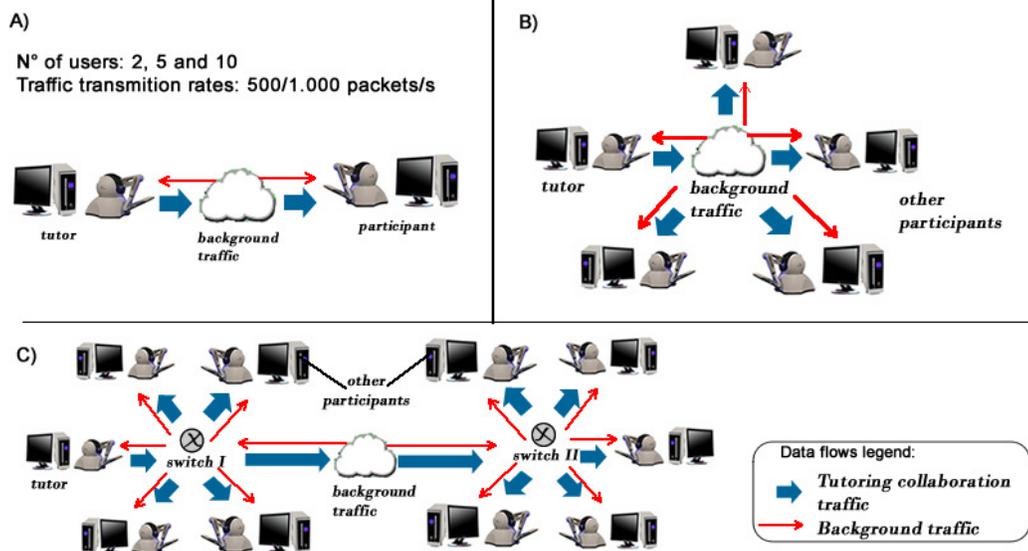


Figure 4. First set of scenarios for simulating tutoring collaborations with 2, 5 and 10 users (A, B and C quadrants respectively).

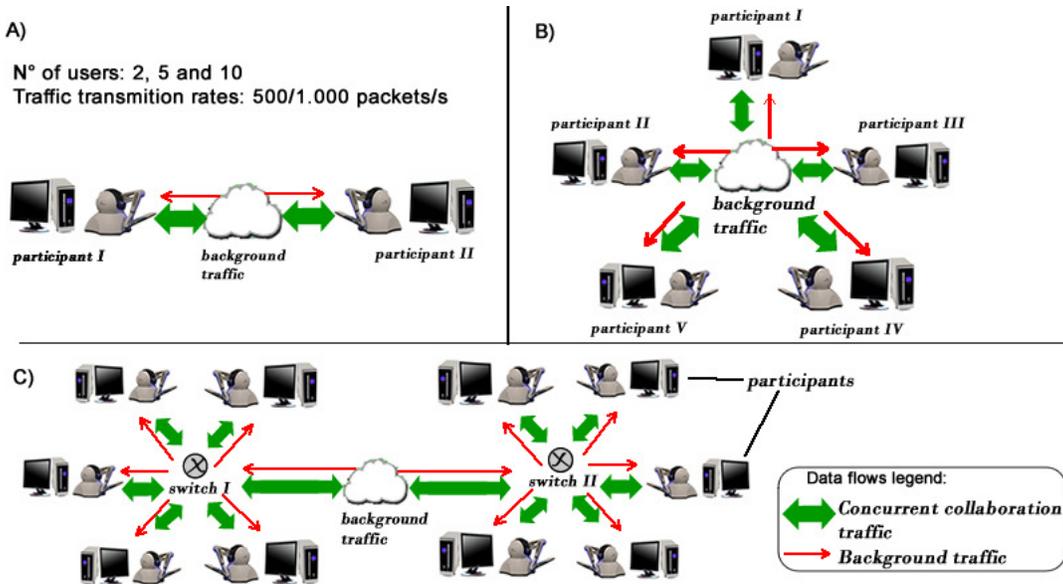


Figure 5. Second set of scenarios for simulating concurrent collaborations with 2, 5 and 10 users (A, B and C quadrants respectively).

After receiving a packet, the application module performs the `send()` function that is responsible for sending it through the network soon after setting values for its fields, which in this case contains the x, y and z positions of the haptic device interator, as well as, the message type field that can vary according to the payload content. The total load of the CybCollaboration's packet is 66 Bytes counting with the UDP, IP and Ethernet headers. The second macro detects when a packet arrives in the receiver module of the node and then it executes the `receive()` function, where the collaboration protocol are defined with Proto-C and C/C++ functions [15] which are responsible for processing the incoming packets. Finally, the last macro detects an interruption scheduled by the init state and executes the `joinRequest()` function that represents a client's solicitation to participate of the collaboration.

C. Collaborative Scenario Set

The types of DHVE explored by the created scenarios utilizes both unidirectional and bidirectional tele-haptics architectures. Therefore, in the unidirectional tele-haptics, the data flows are always transmitted from just one source node and received by the remote clients, and the bidirectional tele-haptics enables that all nodes transmits and receives the data flows in a concurrent manner. The unidirectional tele-haptics are used, for instance, in medical simulators in which there's a professional of a medicine area acting like a guide to train remote students in a medical procedure, such as blood collection applications which needs the use of haptic devices. The bi-directional telehaptics are used in collaborative environments where all hosts are able to perform modifications on it. To investigate how CybCollaboration's protocol reacts with many participants collaborating over a hybrid network (with different bandwidth and architectures), it was defined to run the simulations with two, five and ten users interacting in real-time with both unidirectional and bidirectional DHVE architectures. Thus, it was defined two different sets of scenarios. The first set (Figure 4) contains three sub-scenarios in a tutoring collaboration approach. The second set of scenarios (Figure 5) contains the same number of users and network topologies, but in a concurrent collaboration manner. Before running the simulation, it's important to choose what network metrics and parameters would serve to analyze the performance of the simulated model. Those metrics and parameters are describe in [11] and it respects to the evaluation of all traffic that is generated, transmitted or received via network. Commonly the metrics are represented in average rates like the latency, bandwidth, throughput, delay, jitter or in percentage rates like the packet loss metric [11]. The metrics that were chosen for evaluating our framework performance were the end-to-end delay, the traffic sent and received and the packets loss. The end-to-end delay measures the difference between the packet generation time and the moment of the simulation where the packet is received at the destination node, and it's usual to utilize this

metric to evaluate networked multimedia application's performance. The metrics of traffic sent and received count the average number of packets sent and received by the hosts in a determined period of time and finally the packet loss is the percentage of packets which for some reason weren't received at the destination nodes [13, 8]. For simulating the backbone bottleneck for collaborations with ten users it was used an Ethernet (10 Mbps) link and for the subnetworks it was used a Fast Ethernet (100 Mbps) link. To evaluate the performance of the CybCollaboration model it was fundamental to represent some background traffic to characterize the web applications which runs over the Internet. To simulate this behavior, it was used a background traffic generator node for transmitting random outcomes of packets through the network with variables sizes, modifying the calculation's results of network impairments such as end-to-end delay, traffic sent/received and throughput. To characterize both 500 and 1,000 packet/s transmission rates, used by all scenarios, it was necessary to use the ON/OFF State Time in together with the Packet Interarrival Time attributes of the haptic module. During a simulation with a test bed scenario, it were collected the graphics displaced on Figure 6 which shows that the average throughput of transmitted packets by the tutor node increases to 600 and 2,000 packets/s at the very beginning of the simulation what is expected because of the establishing of all client's connections and the extra packets which transits on the network in this moment, but its stabilizes in 500 and 1,000 packets/s throughput rates.



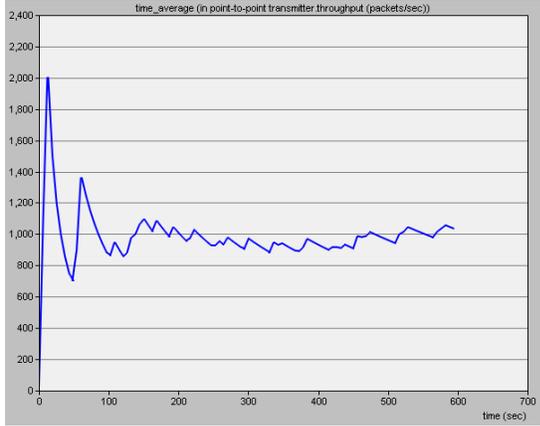


Figure 6. Throughput of 500 and 1.000 packets/s which simulates the transmission rates of the haptic device, used in CyberMed simulation.

IV. RESULTS AND DISCUSSION

The analysis of the results collected on this work simulation was compared to some research found in the literature that explores the acceptable values for some network impairments in DHVEs. One of these works conclusions reveals that for the user perception experiment, a good sense of touch requires a delay of less than 5 ms [6]. In [7], the authors performed several experiments to investigate the requirements for DHVEs which runs under best-effort services. Therefore, it was used a network emulator to analyze the impact of network impairments such as packet loss, delay and jitter on the QoS of force feedback. Those experiments have shown that haptic data should have a different treatment on its transmission and should be placed in separated queues. Results of those work presented acceptable haptic QoS requirements values which are: delay < 10 ms, jitter < 3 ms and 1 – 5% for packet loss rate. Other results presented in [5] explored the medical simulators context and investigates the network parameters required for supporting medical applications that uses tele-haptics. Therefore, it presents other group of QoS requirements for haptic communications which includes: packet loss < 10% and delay < 20 ms. Table I presents a comparative table which presents the necessary QoS requirements for supporting haptic traffic and other kinds of traffic such as video and voice. These haptic traffic parameters were chosen to serve as basis to the CybCollaboration's performance evaluation, and the simulation results will be discussed in comparison with these parameters. The first scenario set was simulated for 10 minutes and the second one for 5 minutes and those results are presented in Table II.

TABLE I. QoS REQUIREMENTS FOR DIFFERENT TRAFFIC TYPES [8]

Traffic	QoS requirements
Haptic	Delay < ~50ms Throughput ~500 kbps – 1 Mbps Jitter < ~2 ms Packet loss < ~10%

Voice	Delay < ~150 ms Throughput ~22 kbps – 2000 kbps Jitter < ~30 ms Packet loss < ~1%
Video	Delay < ~400 ms Throughput ~2.5 kbps – 5 kbps Jitter < ~30 ms Packet loss < ~1%

A. Results of the Tutoring Collaboration Scenario Set

The expected results for the Traffic Sent and Traffic Received metrics, in the tutoring collaboration scenarios (with over than two participating users) can be calculated using the following formula: $T \cdot (n - 1)$; where T is the total number of packets sent and n is the number of participating users. In that way, the expected results for tutoring collaboration scenarios with five users (visible at B quadrant in Figure 7), is that the total number of packets sent should be approximately four times higher than the number of received packets in each participant node, once there's just one node replicating packets to four receiving nodes. The same can be said for collaborations with ten users in which the traffic sent should be nine times higher than traffic received. This expected behavior of CybCollaboration was successfully reached as the graphic displayed on Figure 7. Tutoring collaborations with two users should have the same receiving and transmission rates, and the CybCollaboration module attends to this requirement with approximately 380.000/500.000 transmitted/received packets for both 500 and 1.000 packets/s transmission rates scenarios respectively. For collaboration with five users, CybCollaboration remains with the expectable behavior which is that the packets sent number should be four times higher than the packets received number. For example, in a collaboration with five users and a 500 packets/s sending rate, 340,000 packets

TABLE II. RESULTS OF THE SIMULATED SCENARIOS.

Tutoring Collaboration Scenarios Set			
Transmission rates (packets/s)	Number of users	End-to-end delay (ms)	Packet loss (%)
500	2	0,5	0,0
1,000		5	0,0
500	5	0,9	1,1
1,000		1,1	1,7
500	10	11,2	0,9
1,000		17,5	1,2
Concurrent Collaboration Scenarios Set			
Transmission rates (packets/s)	Number of users	End-to-end delay (ms)	Packet loss (%)
500	2	5,0	0,2
1,000		6,0	0,2
500	5	5,5	0,05
1,000		7,7	0,071
500	10	24,2	10,2
1,000		26,0	10,5

were transmitted by the tutor node while 84,000 packets are received in each node. From these numbers and using the formula previously given it is possible to calculate the expected number of packets received just dividing the total number of packets sent, which in this case is 340,000, by the number of receiving nodes that is 4 giving 85,000 expected packets. Using this number it is possible to calculate the percentage of packet loss by subtracting the number of received packets (84,000) from the expected number calculated before (85,000) resulting in a loss of 1,000 packets (1,17%) as result. As it can be seen from Figure 7, in all tutored collaboration scenarios with 1,000 packets/s of throughput, the total number of packets which were sent and received has almost duplicated when compared with the 500 packets/s sending rate scenarios, what is a normal behavior of this kind of application once more packets are transmitted from tutor node. The end-to-end delay measures the average time in which a packet takes to be transferred from a sender to a receiver node. The average end-to-end delay increases considerably in the beginning of the simulation while

network traffic increases in the same order. In almost all second set of scenarios, the end-to-end delay will be stabilized around 50 s of the simulation time, indicating the end of the establishing of all client's connections. After this moment the end-to-end delay stabilizes in constant values of 0.5, 5, 9, 0.9, 1.1, 11.2 and 17.5 milliseconds as it can be seen on the results Table II. The maximum value of end-to-end delay was 0.0175 seconds (17.5 milliseconds) of retardation for tutoring collaboration, what is totally acceptable according to delay's requirements for supporting telehaptics applications that should be less than 50 milliseconds according to the parameters basis presented on Table I. Thus, for collaborations of tutoring with up to ten users, it can be said that CybCollaboration runs with almost no environment degradations, over a hybrid network, once the maximum packet loss percentage was 1,2 % and the end-to-end delay was 17,5 ms, which is totally acceptable by the QoS parameters of the telehaptic applications presented by the literature previously mentioned [5, 6, 7].

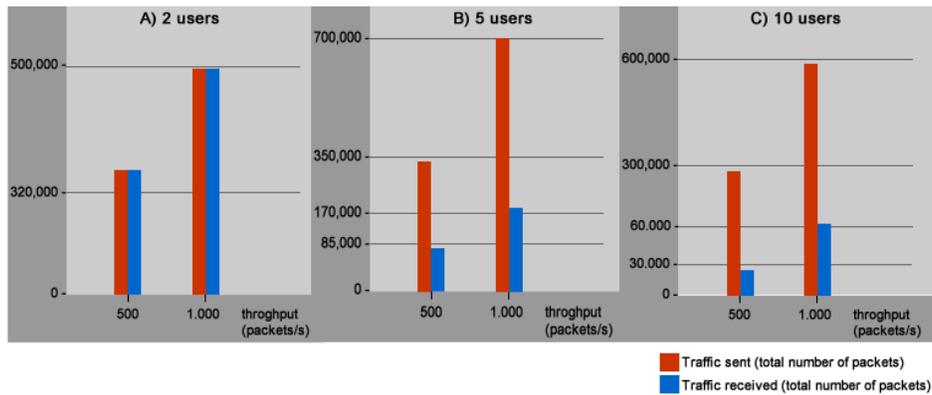


Figure 7. Traffic sent and received measurements results of the Tutoring Collaboration scenarios set.

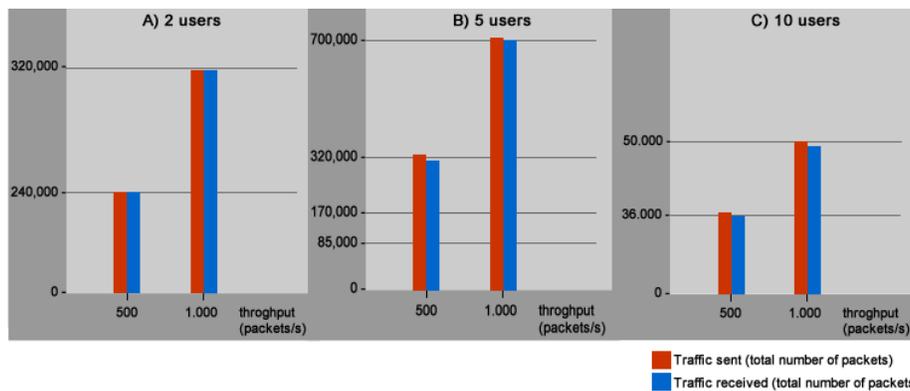


Figure 8. Traffic sent and received measurements results of the Concurrent Collaboration scenarios set.

B. Results of the Concurrent Collaboration Scenario Set

Concurrent collaborations by bidirectional telehaptics approach have a quite different behavior when compared with the tutoring collaboration type. Thus, the data flows runs in both transmitting and receiving directions

increasing considerably the network traffic. Therefore, it is expected for this kind of applications that values of the traffic sent and received should be the same and CybCollaboration satisfies this requirement, as it can be seen at result's Table II. For concurrent collaborations with two users, CybCollaboration has the expectable

behavior with almost no packet loss. In the scenario with five users and a 1,000 packets/s sending rate, for instance, an average of 700,500 packets were transmitted by all nodes, while 700,000 packets were received in each node. The percentage of packet loss is calculated by subtracting the number of received packets (700,000) from the number of packets sent (700,500) giving a loss of 500 packets (0,071%) as result. The maximum packet loss percentages are 10,4% and 10,5 % for the ten users scenario and according to the QoS requirements basis used for this experiment presented earlier in Table II, it is notable that the packet loss exceeded a little the maximum value allowed of 10%, despite that the exceeded percentage is probably insignificant to degrade the VE. The end-to-end delay values of the second set of scenarios are visibly higher than the measurements of the first one. It occurs basically because of higher traffic which transits on the network, once all nodes are sending packets. However, despite of the growth in the network traffic, the top values of 24,2 and 26,0 milliseconds for the ten users scenario with 500 and 1,000 packets sending rates respectively, shows that the VE stays stable and the concurrent collaboration of CybCollaboration runs in a stable form. It is important to highlight that the sending and receiving traffic rates aren't higher on Figure 8 in comparison with the values presented on Figure 7 because the second set of scenarios run just for five minutes, collecting a smaller number of samples values.

V. FINAL DISCUSSION AND FUTURE WORKS

Because of the little or absence of attempts in quantifying or qualifying the QoS requirements for remote haptic collaboration [8], these types of requirements still aren't well defined and can change according to specific experiments and studies. Thus, based on the results obtained and comparing them with the network requirements for haptic transmission found in the literature it was observed that the CybCollaboration's protocol can satisfactorily support both unidirectional as bidirectional collaborations with up to ten users interacting in real-time. Assuming that packets delays should be under 20 ms to maintain the medical telehaptics simulators integrity, like the studies of P. Dev et al. [5] reveals, it can be said that 17.5 ms is an acceptable value for tutoring collaborations with ten users and they are well supported by CybCollaboration with no degradation. Using the parameters proposed by A. Marshal et al. [8] which served as analysis basis for our work, the same conclusions were reached. On the other hand, when comparing our delay maximum results (17,5 and 26,0 ms) with the K. M. Yap et al. [6] and R. T. Souayed et al. [7] studies, which reports that the packet delays over 5 and 10 ms are unacceptable for haptic data transmission, it is observed that CybCollaboration probably starts to be unstable for tutoring collaborations with ten users. Nevertheless, the QoS parameters for supporting haptic data transmission aren't yet well defined and it is important to take in account simulated network conditions, which includes the use of a backbone bottleneck of 10 Mbps and local area

networks running at 100 Mbps with ten users collaborating. Certainly the end-to-end decreases while using higher connections for the backbone, like the 1 Gigabit Ethernet or higher connections, and probably these network impairments should be softened and collaborations may proceed without difficulties. As future works, it will be studied the benefits of implementing the multicast communication architecture on CyberMed's architectures and its results will be compared with the current unicasting architecture of this framework. Further simulations with a large amount of users is also expected, as such results may help in fine tuning the data distribution protocol aiming at enabling successful collaboration with a larger audience.

ACKNOWLEDGMENTS

This project is supported by CNPq through the National Institute of Science and Technology – Medicine Assisted by Scientific Computing INCT - MACC (CNPq Process Number 181813/2010-6 and FAPERJ E-26/170.030/2008) as well as CNPq Process Numbers 312375/2009-3, 308857/2008-9 and FAPERJ # E-26/103.106/2008).

REFERENCES

- [1] Souayed, R. et al. "Haptic virtual environment performance over IP networks: A case study", Proceedings of the Seventh IEEE DS-RT'03, 2003.
- [2] Singhal, S. and Zyda, M., "Networked Virtual Environments: Design and Implementation", ACM Press/Addison-Wesley Publishing Co, New York, USA, 1999.
- [3] Sung, M. Y; Yonghee Yoo; Kyungkoo Jun; Nam-Joong Kim and Jinseok Chae, "Experiments for a collaborative haptic virtual reality", Proceedings of IEEE ICAT'06, Hangzhou, pp. 174-179, 2006.
- [4] Sales, B.R.A. and Machado, L.S. "Interactive collaboration for virtual reality systems related to medical education and training". 6th Int. Conf. on Tech. and Medical Sciences. Porto. 2010.
- [5] P. Dev, D. Harris, D. Gutierrez, A. Shah, and S. Senger, "End-to-end performance measurement of internet based medical applications," in Proceedings of the Annual Symposium of the American Medical Informatics Association, pp. 205–209, San Antonio, Tex, USA, November 2002.
- [6] K. M. Yap, A. Marshall, W. Yu, G. Dodds, Q. Gu, and R. T. Souayed, "Characterising distributed haptic virtual environment network traffic flows," in Proceedings of the 4th IFIP Conference on Network Control and Engineering for QoS, Security and Mobility, pp. 297–310, Lannion, France, November 2007.
- [7] R. T. Souayed, D. Gaiti, W. Yu, G. Dodds, and A. Marshall, "Experimental study of haptic interaction in distributed virtual environments," in Proceedings of Eurohaptics, pp. 260–266, Springer, Munich, Germany, June 2004.
- [8] Alan Marshall, Kian Meng Yap, and Wai Yu, "Providing QoS for networked peers in distributed haptic virtual environments," *Advances in Multimedia*, vol. 2008, Article ID 841590, 14 pages, 2008.
- [9] Basdogan C., Ho, C., Slater, M. and Srinivasan, A., "An experimental study on the role of touch in shared virtual

environments,” *ACM Transactions on Computer-Human Interaction*, Vol. 7, pp. 443-460, 2000.

- [10] Paiva, P.V.F. and Machado, L.S. “Um estudo sobre manipulação cooperativa em ambientes virtuais colaborativos,” In: *VII Workshop de Realidade Virtual e Aumentada*, São Paulo, Brazil, p. 185-190, 2010.
- [11] Carlos Kamienski; Djamel Sadok; Cavalcanti, D. A. T.; Dênio Mariz and Kelvin L. Dias . “Simulando a Internet: aplicações na pesquisa e no ensino,” *Anais do XXII Congresso da SBC (JAI)*. :, 2002, v. 2, p. 33-86.
- [12] Weingartner, E., vom Lehn, H. and Wehrle, K., “A performance comparison of recent network simulators,” *Proceeding of the IEEE ICC 2009*.
- [13] A. S. Tanenbaum, “*Redes de Computadores*”. Editora Campus, 4 rd ed., 2003.
- [14] SensAble Technologies. Online: <http://www.sensable.com/>. Accessed in 02/2011.
- [15] OPNET Technologies Inc. website. Online: http://www.opnet.com/solutions/network_rd/modeler.html. Accessed in 03/2011.
- [16] Willian Stallings, “*Data and Computer Communications*”. Prentice Hall, 8 th ed, 2006.