Evaluating QoS effects on a user centered CVE

Noureddine AGHBALOU YEDRI¹, ², Stéphane LOUIS DIT PICARD¹, Pascal LE MER¹, Jean-Christophe LAPAYRE²

¹France Télécom R&D, 2.av.Pierre Marzin, 22307, Lannion Cedex, France
noureddine.aghalouyedri@orange-ftgroup.com
stephane.louisditpicard@orange-ftgroup.com
pascal.lemer@orange-ftgroup.com

²Laboratoire LIFC (FRE CNRS 2661), 16 route de Gray - 25030 Besançon Cedex, France
jean-christophe.lapayre@univ-fcomte.fr
noureddine.aghbhalou_yedri@edu.univ-fcomte.fr

Abstract: Collaborative Virtual Environments (CVEs) allows geographically distributed team to work together around virtual object over the network. The deployment of CVE over the Internet must take into account that the Quality of Service (QoS) is not guaranteed. This paper presents an evaluation of the QoS effects on a CVE based on multi-channel network system. The aim of the study is to investigate the advantages of such a network system on a synchronous collaborative work. We first describe the network layer features of this CVE. Next we explain the method to assess the impacts of QoS on several interactivity indicators. And finally we discuss the results of an experiment carried out upon a collaborative application and propose a technical approach to evolve network communication layer of our CVE.

Key words: QoS, CVEs, collaborative work.

INTRODUCTION

Collaborative Virtual Environments (CVE) concept appeared in the nineties when the spread of 3D technologies [JOS04]. It allows geographically distributed team to work together around virtual objects over the network. Data shared in this kind of environment might come from different domains such as distance learning, CAD design, medicine, military training, etc. [LEI99]. One of the problems usually met in the deployment of this CVE is the adaptation to heterogeneous networks. Indeed, synchronization of participant interactions are subject to various network limitations, such as packet lost or delays. It is what we call QoS parameters.

In order to determine the effects of QoS on collaborative work, many studies have been performed. Kyoung and Robert [SH99a] assets on the effect of network latency and jitter on a cooperative tele-operation task in a CVE. They compare two commonly used networks, Ethernet and ISDN, to examine the various network QoS and interactivity in a CAVE based CVE system. Vagli and al [VAG99] study the effect of the network delay on task performance in a virtual ball game. The study was conducted on the case of two players and included how the players change their game strategy to take into account the increasing of the network delay. Chen [CHE05] studies the effect of latency and packet loss on coordination task in a Desktop CVS system. Researches have been done to focus on QoS in order to improve the collaborative work in CVEs. Pullen [PUL99] proposes the use of selective reliable protocols to drastically reduce network traffic. Boukerche and Shirmohammadi [BOU04] idea is the synchronous collaborative transport protocol based on the regular transfer of updates with a fixed time between the updated messages; that is the reliable transfer of part of the updates and the unreliable transfer of the remainder updates.

In our project context, Louis Dit Picard proposed a network layer based on a multi-channel communication system [LOU01]. This work is comparable to above approaches. But it is characterised by some distinctive features aimed at taking into account QoS. The communication layer is based on a strong-coherency channel (reliable channel) used for crucial updates and a weak-coherency channel (unreliable channel) used for the streaming data. The objective of this data classification is to improve the consistency between terminals and interactivity. This proposition has been implemented in a CVE named Spin-3D [DUM99]
The aim of this study is to evaluate such a multichannel communication layer against QoS changes. Thus, we studied the shared object manipulation time over the communication layer under the effect of various values of latency and packet loss. This paper is organized as follows: first we describe the Spin 3D user interface requirements and the network layer features. Then we explain the method to assess the impacts of latency and packet loss on several interactivity indicators such as the manipulation visualization time, the transfer time of manipulation updates, and the incoherence time of the system. Finally we discuss the results of an experiment carried out upon a collaborative application and future works to be able to run the Spin-3D on heterogeneous networks such as UMTS or satellite networks.

1. The Spin-3D CVE

1.1. The Interface

Spin-3D [DUM99] is a three-dimensional user interface for synchronous collaborative work. Our interface is designed for small group meetings such as distant learning or co-design situations (Figure 1).

Figure 1. The 3D interface of Spin-3D.

The interface presents to users shared objects (objects that could manipulated by all users). In the interface, each user owns a 3D pointer to be able to designate and pick objects. Each pointer is represented by a telepointer in each remote interface.

1.2. Communication Platform

Observing that client/server architectures are not suitable to ensure a good interactivity because of the delay network, for usability reasons we choose a distributed architecture for Spin-3D [LOU01]. Each shared data is duplicated on each instance of Spin-3D present in the collaborative session. Each user interacts with the local copy, and then the platform ensures the coherency of all data. Figure 2 illustrates the general architecture of the Spin-3D communication layer.

Figure 2. The Spin-3D communication platform.

To support the co-operative activity, it is mandatory to ensure that all distributed copies are in the same state. The Spin-3D provides a strong-coherency channel (reliable channel) for group communication using a reliable transport protocol, at some cost. This channel is supported by a reliable CORBA Multicast [LOU01].

However, in some cases, the coherency can be relaxed: some intermediary positions a moving object goes through, or some frames of a broadcasted movie can be lost without harmful consequences. Therefore Spin-3D provides a second weak-coherency channel using an unreliable transport protocol (unreliable channel).

The 3D object designer has the choice to use any channel to transfer the data between all instances of the Spin-3D terminal.[LOU02].

1.3. The interaction mechanism in Spin-3D

The manipulation of a sharing object on the terminal Spin-3D passes mainly by three steps as shown in Figure 3(a):

1. The step one (Designation and selection): the user points the object which he wishes to manipulate; then after having clicked on this object, the object becomes then selected. Note that the selection is exclusive: it means that only one user should be able to select the designated object at once. It is ensured by a service which associates a token to each shared object. When the selection occurs, the service try to acquire the associated token.

2. The step two: when the object is selected, the user can manipulate it by using the isometric peripheral (step of manipulation).

3. Finally the step three: a click on the isotonic peripheral involves the deselection of the object.

The figure 3(b,c) presents the functioning of the interaction mechanism upon the network in various channels of Spin-3D.
2. Method

We propose an experiment to evaluate the effects of QoS in Spin-3D CVE. In this experience, we focus our study on the effect of the latency and packet loss in our CVE. Several definitions have been proposed for latency, we chose the definition of Delaney at all in [Del06]. “Network Latency is the time taken from the start of exchange of an application protocol data unit (APDU) at the application layer of one participating node to the end of exchange of the same APDU with the application layer of a second participating node”.

2.1. Scenario

Several applications can be developed with the Spin-3D where users are able to work around 3D objects.

To evaluate the influence of the QoS parameters in Spin-3D, we choose a learning situation for our study with the use of a 3D digital camera application, it aims to adjust the camera ring in a collaborative way and to take a photo. Two users, familiar with the Spin-3D interface, adjust one ring of the camera. The experiment proceeds in the same room, each user follows the adjustment manipulation on his terminal. The first user (S1) turns (with a fairly constant speed) the camera ring from the position 1 to the position 45 (Figure 4). The second user (S2) waits the end of the operation on his terminal to take a photo. To avoid the problems of the selection phase, we choose that the user (S1) has already the token associated with the shared action. Thus we consider only the notification of selection step, the manipulation and the deselection step.

The user (S1) selects the camera ring and turns it from the position 1 to 45, and then unselects it. We wait until the camera ring reaches the final position at both stations to end the experiment. Results associated to both stations will be collected by the Ethereal software during the experiment.

2.2. Definitions

The aim of this experiment is to study the effects of the latency and the packet loss on the manipulation time. So, we measure two parameters defined as follow:

- The time of visualization for the group: representing the time of manipulation of the shared object at the level of the working group (including all users).
- The time for transmitting over the network the updates resulting from the manipulation of shared object to remote participants.

We propose the following definitions so that to understand the course of this experience:

- **The time of local visualization** ($T_{VL}$)

  The Time of local visualisation is the time of course of the manipulation (from the selection to the deselection) on the terminal of the user who manipulates the object. The $T_{VL}$ is independent of the channel of communication and QoS on the network. It depends only on the speed of manipulation. In this experiment, $T_{VL}$ is equal to the time of visualization of S1, we fixed it at 6.9 (s) (the way of the time calculated for 50 operations of rotating the camera ring from the position 1 to the position 45 with a fairly constant speed (latency = 0ms and packet loss = 0%).

- **The time of visualization for the group** ($T_V$)

  The Visualization group time ($T_V = T_S - T_1$) as shown in Figure 3 (b,c) is the time of the manipulation (including the selection and deselection) for all users: it starts when the user S1 selects the camera ring and ends when the visual manipulation of the camera ring finishes in both terminals (the camera ring reaches the position 45 and becomes deselected in all interfaces of participants).

- **The time of data transfer over the network** ($T_N$)
The time of data transfer over the network (\(T_N = T_6 - T_1\)) as shown in Figure 3(b,c) starts when the user S1 selects the camera ring positioned to 1 and ends when the data transfer between S1 and S2 is finished (the camera ring reaches the position 45, becomes deselected in all interfaces of participants and S1 receives the ACK of S2). In the case of packet loss 0\%, \(T_N = T_V + L\), where \(L\) is the network latency between S1 and S2.

- **The incoherence time of the system (\(T_I\))**

The incoherence time of the system (\(T_I\)) is the time when the system does not assure the coherence of the collaborative work and of the duplicated data. The incoherence time of the system is equal to the time of local visualisation minus the time of data transfer over the network (\(T_I = T_N - T_{VL}\)) as shown in figure 3(b,c).

### 2.3. The test bed

As shown in Figure 5, we set up an experimental test bed composed of two windows stations running the Spin-3D software (holding the same properties) connected to a Linux station that plays the role of a multicast router. The Linux router embeds a network emulator (NetEtem) [NET06] which allows us to set different QoS parameters (such as the network latency, the packet loss, the jitter and the bandwidth management).

![Figure 5. The test bed.](image)

### 3. Results

We use previous test bed to evaluate the effect of QoS in Spin-3D. The evaluation of the interaction mechanism against different settings of QoS (the latency and the packet loss) is enough to evaluate our CVE. For both channels implemented in the Spin-3D and for different settings of the network latency and the packet loss, we test the interaction mechanism by measuring the times (\(T_V\), \(T_N\) and \(T_I\)) defined in the previous section.

#### 3.1. The case of the reliable channel

The Figure 6 presents the interaction mechanism implemented in Spin-3D running over the reliable channel. All messages are sent successively: a new message is sent only after receiving the ACK for the previous message. If the transmitter does not receive all the ACK for a given packet, the system retransmits again the same packet after a "timeout" (in this experiment, we are fixed the timeout at 500ms). All the next packets stay blocked in a queue until it receives the ACK for the current packet.

![Figure 6. The interaction mechanism implemented in Spin-3D running over the reliable channel:](image)

- a) Network without loss, and a low latency.
- b) Network without loss, and a high latency.
- c) Network with loss, and a low latency.

The Figure 7 show the variations of the times \(T_V\), \(T_N\), \(T_I\) in different setting of latency (0ms to 400ms), with only the latency (packet loss =0\%), the times \(T_V\), \(T_N\) and \(T_I\) increase "exponentially". The increases are mainly due to synchronous transfer used with the reliable channel. Each packet is subjected to the latency; we cannot transmit a new packet before the reception of the ACK corresponding to the packet previously sent.

![Figure 7. The times \(T_V\), \(T_N\), \(T_I\) and \(T_{VL}\) with different settings of the latency.](image)
loss, after this value, these times increase more rapidly. The increase is mainly due to the fact that with the reliable channel all the packets are sent in a synchronous manner. In the case of a loss, S1 retransmit again the same packet only after a timeout, so for each packet lost, the times $T_V$, $T_N$ and $T_I$ increase with a timeout.

![Figure 8: The evolution of times $T_V$, $T_N$, $T_I$ and $T_{VL}$ with different setting of the packet loss.](image)

In summary, we have for the reliable channel: in the case where the packets are subjected to both the latency and the packet loss, we have the following equations:

$$T_N = T_V + L + n \cdot \text{timeout}, \ n \geq 0 \quad (1)$$

Where $n$ is the number of the lost packets for the deselection step (notification packet and the associated ACK). The time $T_V$ already includes of the packets transfer of packets until the object becomes deselected in the interface of S2.

The incoherence time of the system is:

$$T_I = T_N - T_{VL} = (T_V - T_{VL}) + L + n \cdot \text{timeout} \quad (2)$$

These results show that the increase of the times $T_V$, $T_N$ and $T_I$ relies on the three steps of the interaction mechanism. To minimize the effects of the packet loss, we have to choose a little value for the timeout thus increasing the reactivity of the system against the packet loss. But choosing such a value for the timeout will increase the traffic sent over the network.

### 3.2. The case of the unreliable channel

The Figure 9 presents the interaction mechanism implemented in Spin-3D running over the unreliable channel. Note that the selection and deselection steps always run over the reliable channel. With the unreliable channel, the manipulation starts without waiting the ACK of the notification of selection.

![Figure 9: The interaction mechanism implemented in Spin-3D running over the unreliable channel:
   a) Network without loss, and a low latency.
   b) Network without loss, and a high latency.
   c) Network with loss, and a low latency.](image)

The Figure 10 show the variations of the times $T_V$, $T_N$, $T_I$ in different setting of latency (0ms to 800ms), with only network latency (packet loss = 0%), the times $T_V$, $T_N$ and $T_I$ increase successively with the latency. The time $T_V$ raises approximately linearly with the latency in comparison to reference ($T_{VL}$), the will, the times $T_N$ and $T_I$ increase approximately of $2*L$. We validate the following equations:

$$T_V = T_{VL} + L \quad (3)$$

$$T_N = T_V + L \sim T_N = T_{VL} + 2*L \quad (4)$$

And for the incoherence time of system:

$$T_I = T_N - T_{VL} \sim T_I \approx 2*L \quad (5)$$

![Figure 10. The evolution of times $T_V$, $T_N$, $T_I$ and $T_{VL}$ with different settings of the latency.](image)

The Figure 11 show the variations of the times $T_V$, $T_N$, $T_I$ in different setting of packet loss (0% to 20%).
with only the packet loss (latency = 0ms), the times $T_V$, $T_N$ and $T_I$ remain nearly constant for a packet loss less than 5%. For packet loss greater than 5%, these times increase successively.

As shown in Figure 9, for the unreliable channel, only the packets of the selection step (notification) and of the deselection step will have to be retransmitted in case of loss. Only the loss of the packets of the deselection step impact the times $T_V$, $T_N$ and $T_I$. Therefore, the increase of the time $T_V$ is only due to the loss of packets corresponding to the deselection notification. The times $T_N$ and $T_I$ remain greater than $T_V$ as they depend of the loss of more packets than $T_V$ (the notification of deselection and the associated ACK). Nevertheless the increase of the time $T_V$, $T_N$ and $T_I$ is limited as the number of manipulation updates sent remains important comparing to the number of notification packets.

![Figure 11. The evolution of times $T_V$, $T_N$, $T_I$ and $T_{VL}$ with different settings of the packet loss.](image)

In summary, we have for the unreliable channel: in the case where the packets are subjected to both the latency and the packet loss, we have the following equations:

$$T_N = T_{VL} + 2*L + n*timeout, \ n \geq 0$$  \hspace{1cm} (6)

where $n$ is the number of the lost packets for the deselection step (notification packet and the associated ACK).

The incoherence time of the system is:

$$T_I = T_N - T_{VL} \sim T_I = 2*L + n*timeout$$  \hspace{1cm} (7)

The results show that the increase of times $T_V$, $T_N$ and $T_I$ relies only of the deselection step of the interaction mechanism. To minimize the effects of the packet loss, we have to choose a little value for the timeout thus increasing the reactivity of the system against the packet loss.

4. Discussion

The coherence of the collaborative work in Spin-3D is obviously affected by the QoS. The more the values of the latency and of the packet loss great are, the more their effects are important. Nevertheless the effects of these QoS parameters remain various from a communication channel to another. For example, in the case of Spin-3D, the effects of latency and packet loss on the unreliable channel remain acceptable for the latency values going from 0ms to 400ms and for the packet loss values lower than 10%. On the contrary, with the reliable channel, the effects of the latency and packet loss are more important: the range of satisfactory values for the latency and the packet loss is reduced, (0ms to 10ms) for the latency and a packet loss value lower than 5%.

The unreliable communication (unreliable channel) minimize the effects of latency and packet loss on the manipulation time, but this effect remains important on the coordination work in the CVEs [SH99a] [VAG99] [CHE05]. This solution is mainly used to support audio and video streaming.

The reliable communication (reliable channel) does not answer to the real time constraint. Moreover, such a type of communication generates an important traffic on the network. In the case of the reliable channel, choosing the timeout value remains difficult: the more the timeout high is, the more the values of $T_V$, $T_N$ and $T_I$ increase, and the more the timeout low is, the more the system is reactive to detect packet loss. So the time decreases slightly, but the network traffic grows with the re-emission of packets and ACKs (which can be sometimes useless).

The effects of the QoS parameters (latency and packet loss) on the manipulation time of a sharing object are generally translated by:

The visual incoherence between the interfaces of S1 and S2 results in misunderstandings during the session of collaborative work. The incoherence is mainly due to the fact that $T_V > T_{VL}$. For example, S1 selects (deselects) the camera ring; however due to the delay, it stays deselected (selected) on S2 until it receives the selection (deselection) notification. Or on both interfaces, the camera ring could be located in different positions at the same time. Gergle and colleagues study in [GER06] the effect of the visual delay in the performance on the collaborative work. They determine that the effect of the visual delay increases with the latency value and causes the difficulty of the work.

The incoherence of the system is mainly due to the fact $T_N > T_{VL}$. The incoherence time $T_I$ reflects the performance of the system: the more the time $T_I$ increases, the more the performance of the system to ensure the coherence of the collaborative work decreases. During this time, the system is not stable; for example, S1 and S2 can share different information for a shared object (producing the visual incoherence). Moreover, the system can not guarantee the transfer of the message, S1 can re-select the shared object (acquisition of the same local token) but the notification of remote users could be blocked at S1 (the notification message is posted to the queue of the
reliable channel and will be handled later). Moreover in the case of the unreliable channel, S1 can transfer the manipulation updates to S2. However, in the case of the reliable channel, the transfer of packets will be blocked until all previous packets already pushed in the queue will be successfully sent.

5. Conclusion and Future Works

In order to evaluate the QoS effects on the Spin-3D CVE, we have carried out an experiment using a test bed. This test bed connected Spin-3D terminals through a multicast router to vary some factors of QoS between the various terminals. We observed that the manipulation time evolves greatly on the visual and the network level. The results show that the communication layer on Spin-3D based in a multi-channel communication is limited. We think that the reliable channel must be limited to the transfer of sensitive data; And that the unreliable channel must be used to transfer data update.

Current works aim to upgrade Spin-3D with the introduction of a semi reliable and reconfigurable communication channel to adapt Spin-3D to the QoS. We think that the supervision of network to inform the system with values of the QoS factors (latency, lost, jitter...) to adapt the transmission allow to prevent problems. Such a channel will be based on the reliable/unreliable transmission of the updates. The aim of this channel is to take into account all QoS factors in order to define a strategy of transfer which allowing us to have a better manipulation and guarantee the coordination and coherence during the collaborative session. Moreover, informing also the users with the factors of QoS by the visual index allow reducing the incoherence problems [GUT04].

REFERENCES


