

Issues in Message Passing for a Tele-immersive Experience

Ravi Prakash
Department of Computer Science
University of Texas at Dallas
Richardson, Texas 75083-0688, U.S.A.
email: ravip@utdallas.edu

Abstract

Tele-immersion is one of the applications being developed in conjunction with the Internet-2 initiative. Most of the research in this area has focused on issues pertaining to image acquisition and display, tracking technologies, robotics, haptics, etc. In order to maintain consistency of a tele-immersive experience one also needs to develop a suit of solutions that will ensure proper coordination between the participants. This will require protocols for real-time causal delivery of messages while maintaining the privacy of conversations and preventing forgery of message contents. This paper discusses these issues while considering a tele-immersive environment as a distributed message passing system.

1. Introduction

The goal of this paper is to discuss a possible tele-immersive application. In the process we will outline the capabilities and interface provided to participating users. The idea is to take teleconferencing to the next level. Geographically distant participants should be able to interact with each other as if they were in the same room.

Tele-immersion is one of the applications being developed in conjunction with the Internet-2 initiative [9]. As stated in [10] this will require advances in the areas of:

- 3-D environment scanning,
- projective and display technologies,
- tracking technologies,
- audio technologies,
- robotics and haptics, and
- networking.

Most of the examples of tele-immersion tend to consider a very controlled environment. For example, a virtual classroom for distance education or a virtual meeting room

where a small number of individuals/groups interact with each other. The goal of such applications is to enable all parties to interact with each other, and to accurately deliver information to every participant.

As described in Section 2, this paper considers a much more flexible, and possibly chaotic, tele-immersive environment. It is our belief that an unconstrained tele-immersive environment will lead to a more spontaneous interaction between the participants.

Most of the work, so far, in the field of teleimmersion has concentrated on computer graphics, tracking technologies, etc. The author believes that in addition to these, coordination between participants and message passing protocols to accomplish the same need further study. So, this paper will be limited to a discussion of the distributed message passing issues of tele-immersion. Participants in a tele-immersive session, along with the network connecting them, will constitute a distributed system with no shared memory, nor a common clock, and all communication between them will be through message passing.¹ In Section 3 we will discuss the issues of:

1. group membership in tele-immersive applications,
2. determination of the subset of messages to be delivered to a process, based on its membership of various groups,
3. ordered message delivery and inter-process coordination in such a distributed environment, and
4. security and privacy of communication in such a network.

2. Application Description

Consider a virtual room of some dimension with a finite number of doors. The room may contain some whiteboards

¹One can assume the presence of tightly synchronized clocks as described later.

and some electronic tickers. Initially, the room is empty. Participants can enter and leave the room only through the doors. Once inside the room, participants can navigate through the room at a speed bounded from above by the participants' virtual running speed.² Once a participant, Alice, exits a room through the door she can re-enter through the same door without any delay. However, the minimum time for Alice to re-enter the room through another door is the virtual distance between the two doors divided by Alice's maximum virtual speed.

Once inside the room, Alice has no restrictions on movement except the restriction of maximum speed. A change in Alice's location alters the view presented to her in real time. Concurrently, the change in Alice's position is also visible to other participants in real time.

As Alice nears a subset, S , of other participants she can hear the conversation between members of S . As Alice moves away from S the voices heard from members of S fade. The sounds heard by Alice are the aggregation of the sounds generated by other participants, adjusted for their distance from Alice.

As participants are aware of their relative locations, they will be able to interact with each other just as humans do in a real setting. If a group of participants, S , sees Alice move towards them they can change the conversation if they do not wish to share some information with Alice. As soon as Alice is out of virtual earshot of S , members of S can resume their earlier conversation.

Participants can write onto and read from various whiteboards in their proximity. The contents of the whiteboards are readable until they are explicitly deleted. Optionally, there may be tickers on which transient information is visible.

In terms of the variety of tele-immersive experiences there could be a set of well known virtual rooms where participants interact within the parameters set by the creator/operator of the room. A set of participants could also get together and create a room for a certain duration. Some rooms could be public with unrestricted access while others could be private. In some rooms the actual participants would be humans with computers assisting them. In others a computer could be an active participant. The possibilities are endless. The idea is to take us from chat rooms of today to the holodeck of Star Trek.

3. Issues in Message Passing

To realize the application described in Section 2 various technological advancements are needed in the areas of graphics, projection, movement tracking, audio and video reproduction, robotics, sensory inputs, etc. Moreover, a participant may need to go to a special room equipped with

²The maximum speed is determined in advance.

multiple cameras, microphones, 3-D projectors, surfaces on which images can be projected, and speakers that provide surround-sound. Such a set-up will be in place at every participant's site. These instruments will communicate with each other, at the same site and across sites, over a high bandwidth network. In this section we will concentrate on the distributed message passing issues that need to be addressed for the application.

3.1. Lifetime of Objects

Every entity in the system will be treated as an object. All participants (including humans and computers) will be persistent objects. Such objects have lifetimes that can span the duration of the tele-immersive experience. They have well defined interfaces for interaction with the environment. In the case of computer participants the public and private methods are defined by the way they are programmed. For human participants their behavior depends entirely on them. A whiteboard is a persistent object that provides two public methods: writing and reading,³ and the contents of the board represent its state. Thus, the value returned by the read operation is determined by the preceding sequence of write operations. However, the read and write operations are constrained by the logical distance between the whiteboard and the participant. The writer has to be no more than a certain virtual distance from the board to be able to write on it. The farther a reader is from the board the smaller the image.

In contrast to the participants and whiteboards, messages are transient objects. Each message object is created by a participant and is consumed by another. If multicasting is accomplished using many one-to-one communications, multicast messages correspond to multiple instances of a message object. On the other hand, if all the recipients of a multicast message receive exactly the same message content and a one-to-many communication protocol is used, a multicast message can be treated as a single object. Later, in Section 3.3, we state how sometimes we have to treat a multicast message as multiple one-to-one communications if a continuum of message degradations is required for proximity-based content delivery.

In some cases a participant may receive a message, modify it, and forward it to one or more participants. In such a case the possible modifications are limited by the public methods provided by the message object. As described in later sections, such a restriction is important to ensure privacy of communications and to prevent forgery of messages. Also, a participant should be able to authenticate the message source and message contents before executing its methods locally.

³We consider erasing something on the board to be a write operation, too.

3.2. Group Membership

As participant Alice moves through the virtual room she interacts with different groups of participants. With each group the interaction is unique. To facilitate communication, membership services may run on top of the network and maintain dynamically changing communication groups [2]. Alice's application program, running on top of the membership service, only has to specify the name of the group to which a message has to be delivered [8, 17]. The membership service maintains information about the identities of participants that belong to the group and delivers the message to them. Thus, the application program is shielded from changes in groups and is not responsible for managing them. This promotes modular design of distributed applications [8].

It has traditionally been assumed that, in the context of a given application, a process belongs to a single group at any instant of time [6]. Additions to a group take place on the recovery of previously failed processes. Deletions from a group take place on the failure of previously operational processes. However, in the context of the tele-immersive application described earlier, group membership is affected not only by the state of participants (operational or crashed) and links (connected or disconnected), but also by the movement and relative locations of the participants. Moreover, Alice may simultaneously belong to multiple groups. After-all, it is quite common for human beings to simultaneously participate in multiple conversations in a variety of social settings.

Besides the virtual motion of the participants, group membership should also be affected by the strength of the transmitted signal. For example, if Alice speaks softly the group of participants that can hear her is limited to those in close range, and if Alice shouts her message should be delivered even to participants that are at a great distance.

3.3. Proximity-based Content Delivery

As described earlier, information received by Alice is a function of her distance from the source of the information. Speech signal is supposed to attenuate as it travels away from the source. Thus, for a natural feel to the experience the goal is to actually deliver a degraded form of the message to the receivers. This is quite contrary to the goal of most of preceding research in message communication, which is accurate delivery of messages. How to deliver degraded messages is a research issue in the context of privacy of conversations. This will be described later.

Ideally, there should be a continuum of message degradations with distance. Achieving such a goal would be computationally expensive. Alternatively, the message can be degraded into a small number of levels: from almost noise free at close range to entirely unintelligible at greater dis-

tances. The resultant loss of authenticity of the experience may not be perceived by the human participants. For example, for human voice we could define the levels based on proximity to the speaker. Participants far away from the speaker may only hear *comfort noise*.

Regardless of whether we provide a continuum or discrete levels of degraded signals, we will need a *proximity sub-layer* in the protocol stack below the application and group membership layers. The proximity sub-layer will get its input from the instruments that track movements of all the participants in the room. Based on this input it determines the participants in the *d-proximity* of Alice, where *d* is a parameter provided by the group membership layer. The group membership layer can use the *d-proximity* information to form groups.

3.4. Privacy of Conversations

For voice the message played out at the receiver should be a function of the receiver's distance from the sender. One simple approach could be for all messages to be broadcast to every participant in the room. At the receiver each received message could be attenuated based on distance from the source. All such messages could then be combined and played out. However, this option is not acceptable as it cannot ensure privacy of conversations. A participant that is well beyond hearing range of a conversation still receives the original message and could amplify and/or record it to snoop on distant conversations.

So, the other option is to degrade the signal prior to reaching the destination in such a fashion that a distant receiver cannot extract the original signal from the received signal through signal transformations. This will help reduce the possibility of an unintended participant hearing a private conversation.

The issue at hand is how to implement the privacy requirement described above. Some of the possibilities are:

1. **Centralized server:** All participants could send their messages to a trusted server. The server, based on its knowledge of the relative locations of all participants in the room, could modify and combine the signals on a per receiver basis in real time. Then, the server could send a personalized combination message to every participant. If there are n participants then there are $O(n)$ message streams flowing through the network. Here, a message stream is defined as a stream of packets sent by one participant to another.

In such an implementation the central server is a single point of failure and could become a bottleneck. So, this approach will not scale to large numbers of participants. Moreover, if a set of participants spontaneously decide to start a new tele-immersive room who would become the central server? If one of the

participants becomes the central server it will have to shoulder greater responsibility than others. Also, this participant is aware of all the conversations.

- 2. Distributed approach with preprocessing at source:** In this approach the source uses its knowledge of the location of other participants to suitably modify the message for each participant. Then the source sends personalized messages to each receiver. All that the receivers do is combine all the received messages in real time and play out the result. For an n participant system this results in $O(n^2)$ message streams.
- 3. Tree-based approach:** In this approach all the participants in the room are arranged as a logical tree with the source of the message being the root. Participants closer to the source, that are expected to receive the message with least distortion, are closer to the root of the tree. When a message is generated by the source's application a small number of modified copies of the messages may be sent by the source, one to each child. At each internal node the received message is played out as well as further modified (degraded) for transmission to children. This approach may result in better distribution of responsibility among all the participants.

As each one of the n participants can generate messages we have n message propagation trees in the system, one per source, for a total of $O(n^2)$ message streams. Greater the number of levels in the tree, greater the latency in communication. So, this approach is feasible for shallow trees only.

Once messages are delivered to a participant, the participant can choose to record the view, take snapshots, or enhance certain parts of the image in real-time. Such an action should be taken with the knowledge of all other participants. This is because human behavior is sometimes influenced by the knowledge that it is being recorded.

3.5. Authenticity of Real-Time Clocks or Sequence Numbers

In tele-immersive applications messages have to be delivered in real-time. In such instances the timestamp of the messages will be used to determine the order of message delivery. However, timestamps derived from logical clocks based on Lamport's *happened before* relationship [12] may not be sufficient for this purpose. Messages will have to carry the timestamps obtained from the physical clocks at individual participants. All the physical clocks will have to be synchronized for this purpose [11, 18]. Allowances will have to be made for the fact that it is not possible to perfectly synchronize all clocks in the system.

As the order of delivery of messages is influenced by their timestamps, there may be incentives for participants to forge the timestamps of messages they generate. Sometimes, participant Alice has to forward a message generated by another participant Bob and Alice may have an inclination to tamper with the timestamp of such a message. Forging timestamps may provide little or no advantage in the presence of a central controller. The controller has global knowledge of system behavior and may be able to detect and/or neutralize acts of timestamp forgery.

However, such forgery could be exploited when the control is distributed because no participant may have the latest global information. Let Alice receive a message from Bob with a certain timestamp. Based on the contents of this message Alice generates a message of her own with a forged timestamp that is earlier than the timestamp of Bob's message. A third participant that receives both these messages and uses the timestamps to determine the order of message delivery will not realize that Bob's message is the predecessor of Alice's message.

It is possible to make it difficult for Alice to forge timestamps placed by Bob on messages. For this purpose public key encryption [7, 15] may be employed. When injecting a message into the system, Bob may encrypt the timestamp using his private key. Alice and everybody else can obtain the actual timestamp using Bob's public key. However, as Alice does not have Bob's private key she cannot directly forge a timestamp so that appears to originate at Bob. However, Alice can exploit the fact that time is monotonically increasing to undo this solution. Let Alice receive messages m_1 followed by m_2 from Bob. Let the timestamp of the two messages be t_1 and t_2 , respectively. Any subsequent message generated by Alice is in the causal future of both m_1 and m_2 . However, Alice may only forward information about its dependence on events at Bob at or before m_1 (using t_1 encrypted with Bob's private key).

Sophisticated protocols need to be developed to prevent such forgery of timestamps. However, a solution that requires one or more rounds of message passing between participants to detect forgery of timestamps may not be acceptable. Such a solution may fail to satisfy the real-time constraints of the application.

In the following discussion we will assume that all timestamps are authentic.

3.6. Preserving Causal Relationship between Events

Preserving the causal ordering of message communication is extremely important. Causally ordered message delivery requires that if a message and its causal predecessor have a common destination then at that destination the predecessor should always be delivered first.

However, video and audio streams are usually sent using an unreliable transport protocol like UDP. So, sometimes a

message may reach a destination while its causal predecessor may be lost in transit. For a real-time system, like the one described here, it is not possible to indefinitely delay message delivery. Every message has a deadline. A message may be delivered to a destination after all its causal predecessor messages meant for that destination have either been delivered or their deadlines have expired. This is the basic idea of the Δ -causal ordering protocol proposed by Baldoni, Mostefaoui and Raynal [3].

However, [3] and all general purpose causal ordering algorithms prior to that, including [1, 5, 14, 16], require each message to carry an $n \times n$ matrix that contains information about all causal predecessors of the message.

In the tele-immersive application described earlier most of the time a message will be sent to only a small subset of participants that are within a short distance of the source. All other participants will receive comfort noise. Maintaining causal order between successive messages that carry comfort noise is not necessary. Hence, sending an $n \times n$ matrix with every message may be wasteful. Also, the set of destinations could change with every message.⁴ For such a scenario the Δ -causal ordering solution can be combined with the causal ordering algorithm proposed by Prakash, Raynal and Singhal in [13]. In the causal ordering algorithm described in [13] every message carries information only about its direct causal predecessors. By eliminating other kinds of control information from the messages the communication overheads can be significantly reduced. Also, if a small subset of participants interact with each other for a period of time, excluding others from the conversation, the overheads are proportional to the square of the number of participants in that conversation, and not the square of the total number of participants in the tele-immersive experience. As most of the conversations do take place in small groups this does result in significant reduction of actual communication overheads, even though the quadratic complexity of the overhead still holds. The algorithm is also capable of handling dynamically changing multicast communication groups.

A combination of Δ -causal ordering concept of [3] and the direct causal predecessor concept of [13] is described in [4].

3.7. Synchronization of Streams

Δ -causal ordering of messages is necessary to ensure a logical consistency among messages belonging to one media stream. In addition, contents from different media streams have to be synchronized with each other for inter-media consistency. However, different media streams impose different requirements that make it difficult to synchronize them. For example, audio and video streams can with-

⁴Something spoken loudly should carry farther and therefore has a greater number of destinations than something spoken softly.

stand occasional loss of packets and can be sent using unreliable protocols like RTP over UDP. On the other hand, text and figures to be displayed on whiteboards have to be sent in a reliable fashion using TCP. But, TCP provides no real-time guarantees and may choke the flow of packets during periods of network congestion.

So, it will be a challenge to synchronize audio and video streams from participants with their writing on the whiteboard.

4. Conclusion

Some of the challenges in designing and implementing tele-immersive applications have been presented. It has been shown that issues of participant authentication, communication privacy, group membership, causally and totally ordered message delivery with real-time constraints, and media synchronization will be very important for such applications.

Substantial work has already been performed in these areas, though not directly in the context of tele-immersive applications. The sheer complexity and variety of tele-immersive interactions among participants will require us to revisit some of these solutions. This is a very fertile area for future research.

References

- [1] S. Alagar and S. Venkatesan. Causally Ordered Message Delivery in Mobile Systems. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 169–174, Santa Cruz, December 1994.
- [2] R. Baldoni, R. Friedman, and R. van Renesse. The Hierarchical Daisy Chain Architecture for Causal Delivery. In *Proceedings of the 17th IEEE International Conference on Distributed Computing Systems*, pages 570–577. IEEE Press, 1997.
- [3] R. Baldoni, A. Mostefaoui, and M. Raynal. Delivery of Messages with Real-Time Data in Unreliable Networks. *Journal of Real-Time Systems*, 10(3):245–262, 1996.
- [4] R. Baldoni, R. Prakash, M. Raynal, and M. Singhal. Efficient Δ -Causal Broadcasting. In *Proceedings of the 23rd EUROMICRO Conference*, pages 617–626, Prague, September 1996. IEEE Press. ISBN 0-8186-7487-3.
- [5] K. Birman, A. Schiper, and P. Stephenson. Lightweight Causal and Atomic Broadcast. *ACM Transactions on Computer Systems*, 9(3):272–314, 1991.

- [6] F. Cristian. Synchronous and Asynchronous Group Communication. *Communications of the ACM*, 39(4):88–97, April 1996.
- [7] W. Diffie and M.E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22:644–654, 1976.
- [8] R. Friedman and R. van Renesse. Strong and Weak Virtual Synchrony in Horus. Technical Report TR95-1537, Department of Computer Science, Cornell University, August 1995.
- [9] Internet 2 Applications Working Group. Internet 2. <http://www.internet2.edu>.
- [10] Internet 2 Applications Working Group. Tele-Immersion. <http://www.internet2.edu/html/tele-immersion.html>, January 2000.
- [11] H. Kopetz and W. Ochsenreiter. Clock Synchronization in Distributed Real-Time Systems. *IEEE Transactions on Computers*, 8:933–940, 1987.
- [12] L. Lamport. Time, Clocks and the Ordering of Events in a Distributed System. *Communications of the ACM*, 21(7):558–565, July 1978.
- [13] R. Prakash, M. Raynal, and M. Singhal. An Adaptive Causal Ordering Algorithm Suited to Mobile Computing Environments. *Journal of Parallel and Distributed Computing*, 41:190–204, 1997.
- [14] M. Raynal, A. Schiper, and S. Toueg. The Causal Ordering Abstraction and a Simple Way to Implement it. *Information Processing Letters*, 39:343–350, 1991.
- [15] R.L. Rivest, A. Shamir, and L.M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
- [16] A. Schiper, J. Egli, and A. Sandoz. A New Algorithm To Implement Causal Ordering. In *Proceedings of the 3rd International Workshop on Distributed Algorithms*, LNCS-392, pages 219–232, Berlin, 1989. Springer.
- [17] R. van Renesse, K.P. Birman, and S. Maffei. Horus, a flexible Group Communication System. *Communications of the ACM*, April 1996.
- [18] P. Verissimo. Ordering and Timeliness Requirements of Dependable Real-Time Systems. *Journal of Real-Time Systems (Kluwer Eds.)*, 7:105–128, 1994.