Guaranteed delivery of semi-synchronous IP-based communication

Elroy P. Julius and William D. Tucker
Computer Science Department, University of the Western Cape, Private Bag X17, Bellville, 7535
Tel: (021) 959 2461, Fax: (021) 959 3006, E-mail: {2000266, btucker}@uwc.ac.za

Abstract— This paper presents an empirical solution for guaranteeing the delivery of synchronous and asynchronous messages within a semi-synchronous IP-based communication domain. The communication infrastructure that is needed between Deaf and hearing communities forms the application domain within which the research is situated. SoftBridging is a framework for multi-modal bridging as well as multi-user, multi-modal conversation sessions. An implementation of this concept called SoftBridge for Instant Messaging Bridging Application (SIMBA), is a communication platform that allows a hearing and Deaf person to communicate inside a single uniform space. The system makes use of various web services to do the actual data conversions such as voice to text and text to voice. Publish-subscribe systems are an emerging paradigm for building a range of distributed applications. The architecture of publish-subscribe systems make use of Message Oriented Middleware (MOM) to guarantee reliable delivery of messages within a communication domain. We have incorporated a publish-subscribe system within the overall architecture of SIMBA. We have thus modified the existing architecture of SIMBA to reliably transport semi-synchronous data over a synchronous established session. We have chosen to use the Session Initiation Protocol (SIP) for the establishment of a synchronous session between various users and SIMBA. The system is currently being used as a basis for developing a Deaf telephony application that guarantees the delivery of messages no matter the synchrony.

Index Terms – Asynchronous call handling, Soft Bridging, Synchronous call handling, Publish-subscribe systems, Deaf Telephony.

I. INTRODUCTION

This paper presents an empirical solution for guaranteeing the delivery of synchronous and asynchronous messages within a semi-synchronous IP-based communication domain. The communication domain constitutes two communities, the hearing and the Deaf [8]. The majority of the Deaf community in South Africa is exposed to a lack of Information and Communication Technology (ICT) training [2]. The advancement of ICT is a complete loss for the disadvantaged Deaf community as they are excluded from the equation which ultimately has great benefits for improving their quality of life. Technology in the field of telecommunications has far reached great advancement compared to a couple of decades ago. We are currently reaping the benefits of the 21st century's "information age". From simply making a landline telephone call to calling someone on a mobile telephone or sending a text message from the Internet to his/her mobile telephone device, we are all sharing this "information". Realizing the exclusion of the Deaf community from ICT, the research is geared towards positioning both hearing and Deaf communities into an "equal" communication and information space [8]. This positioning will help to narrow the digital divide between advantaged and disadvantaged communities. The Deaf and hearing communities are brought together into one communication space where interaction between these communities is that of a reliable semi-synchronous form of data exchange.

The remainder of the paper is organized as follows. Section 2 provides background information that help formulate the research question. Section 3 motivates the research while section 4 provides an overview of related work. Section 5 explains the process of answering the research question. In section 6 a technical solution, for the research question, is explained and in section 7 conclusion and future work is discussed.

II. BACKGROUND

SoftBridging is a framework for multi-modal bridging as well as multi-user, multi-modal conversation sessions [5]. The initial prototype of this concept called SoftBridge [10] was based on the Jabber protocol (*http://www.jabber.org)*, which allowed communication to occur asynchronously. [10] makes use of various web services to do the actual data conversions, such as text to voice and voice to text. The semi-synchronous interaction between end users is due to the media adaptations that the system performs according to the media capabilities of the end user. The system, for example, allows end users with different media capabilities, such as voice and text, to communicate with each other inside a uniform media space.

SoftBridge for Instant Messaging Bridging Application (SIMBA) is another system based on the SoftBridge concept [15]. This system focuses on providing life-cycle management, a carrier grade characteristic, for an IP-based application. SIMBA provides a Deaf Telephony service similar to that of [10]. The system itself is based on the Session Initiation Protocol (SIP) for session establishment. The initial SoftBridge prototype and SIMBA share a distinct commonality; they were not designed for a reliable form of data exchange.

III. MOTIVATION

The authors would like to acknowledge the Telkom/Cisco/THRIP Centre of Excellence at the University of the Western Cape for their support of this research.

In an environment where one can not guarantee reliable IP-based network communication, most services are based on asynchronous call handling because of its store-and-forward nature of data exchange. Although asynchronous communication, to some extent, guarantees the delivery of messages, synchronous communication is a better form of data exchange because of its real time properties. There is thus a distinction between synchronous and asynchronous communication: Synchronous is best suited for real time communication and asynchronous for "reliable" communication. Synchronous communication does minimal error checking and needs a "clean" line of connection for data transfer. The motivation behind this research is related to the derivation of certain methods for relating asynchronous communication to synchronous communication. This relationship allows a SoftBridge system to guarantee exactly-once, reliable delivery of messages within a communication domain irrespective of the synchrony, with direct applicability to a Deaf Telephony domain. How does one allow a SoftBridge system to deal with most fault-related issues on one hand and guarantee message delivery on the other? The main research question is asked: "How does one guarantee exactly-once delivery of semi-synchronous messages within a communication domain?"

Most Deaf people in Cape Town use cellular telephones as a means of communication. This device does not guarantee the delivery of short messages sent but remains popular among the Deaf as it is provides a relatively cheap form of communication. Those that do not have access to cellular telephones must travel to the Deaf Community of Cape Town (DCCT) community center to access a public Personal Computer (PC). This is because the PCs are connected to an Asymmetric Digital Subscriber Line (ADSL) network that provides Internet connectivity 24 hours per day. The community members at DCCT make use of e-mail as there main source of communication over the Internet. At the moment the majority of Deaf people exchange e-mail with other Deaf people with the minority communicating with both hearing and Deaf people. The Deaf Community at DCCT is faced with delays on a macro level. This is due to the fact that in most cases Deaf people have to wait to access a PC that have a set duration period assigned to it. Another contributing factor to the macro delay is the problem of transport. The majority of Deaf women and men prefer to go to DCCT during the day as public transport, such as trains and taxis, are dangerous during the evenings. This is problematic as there are currently only five PCs available for public access. The Deaf community needs an application that reliably transports messages to destinations whether the receiving party is connected or not.

IV. RELATED WORK

A. Reliable synchronous communication

When traffic is sent and received within a bounded interval with the absence of timing faults we have a synchronous form of communication. An example of synchronous communication is a telephone conversation that provides a real time framework for communicating entities. There are traditionally one of three techniques used for error control in real time communication. These include: (1) automatic repeat request (ARQ) which is a combination of acknowledgements, time-outs and retransmissions, (2) forward error correction (FEC) and (3) error concealment (EC) at the receiver [12][3]. A major Quality of Service (QoS) requirement for real-time media, such as video and audio, is to deliver packets reliably to the end receiver(s) and in a timely fashion. Satisfying this requirement becomes more challenging as the distance between the end points increases. Retransmission schemes are very impractical for wide area multimedia communications due to the real-time constraints (e.g. the Real Time Protocol (RTP) [16]) and due to the small bit error rates of optical networks, and the ability of real-time media, especially video, to tolerate loss, they can be done away with. [9] describes a mechanism for integrating text and audio into one RTP packet for transmission over a synchronous established session such as Session Initiation Protocol (SIP). The paper reconciles the User Data Protocol (UDP) with the Transmission Control Protocol (TCP) by using the fundamentals learnt in TCP, such as error correction, and applying it to RTP which is based on UDP.

B. Reliable asynchronous communication

Asynchronous communication has an advantage over synchronous communication as synchronous communication has the limitation of requiring all infrastructure elements between distributed components be available at the time of the transaction. As an example consider a telephone conversation. When placing a synchronous call the receiving party needs to be on-line/available before the conversation can commence. Reliable asynchronous messaging allows a client, service or application to interact with other clients, services and/or applications using some form of persistent local message queue, regardless of whether the remote entity is actually available when the application initiates the interaction. The message is delivered in the form of a reliable-messaging provider which can be a persistent local queue or some form of message oriented middleware (MOM). The messaging infrastructure manages the communication between external resources, services and clients to achieve the highest possible system performance and reliability.

NaradaBrokering makes use of MOM to ensure the reliability of the system [13]. The system itself makes use of one or more brokers which provides delivery services for the messaging system. Message delivery relies upon a number of supporting components that handle connection services, message routing and delivery, persistence, security, and logging. In order for NaradaBrokering to perform message delivery, a broker must set up communication channels with clients, perform authentication and authorization, route messages appropriately, guarantee reliable delivery, and provide data for monitoring the system performance.

Small networks are commonly seen as well-behaved networks where large-scale networks behave like the public Internet, exhibiting disruptive overloads and routing changes, periods of poor connectivity and throughput instability. As the numbers of participating components become larger, the failures exponentially increase. Some form of scalable perennial technology needs to be created to ride out forms of infrastructure instability, imposing loads that are either constant or growing very slowly, as a function of system size and network span. The Spinglass project employs gossip protocols at very high speeds. These protocols have an unusual style of probabilistic reliability guarantees [4]. The Spinglass project works towards overcoming scalability barriers as well as a methodology yielding applications that remain secure and robust even when failures occur or experiencing a denial-of-service attack. The project investigates reliable protocols under the influence of mundane transient problems, such as network or processor scheduling delays and brief periods of packet loss. [4] disproves the common acceptance that reliable protocols eliminate these problems. Particular attention is placed on the impact of a disruptive event as a function of scale (system and network size). The Spinglass project proves that reliable protocols degrade under mundane stress, a phenomenon attributable to low-probability events that become more likely and more costly as the scale of the system grows.

Publish-subscribe systems provide the capabilities of guaranteeing exactly-once delivery of messages between multiple end points. The system replaces the single destination in a point-to-point model with a content hierarchy, known as topics. Message distribution is handled by an underlying content based routing network [6]. These types of networks perform routing based on the data being transported in a message rather than on any specialized addressing and routing information attached to, or otherwise associated with, the message.

Proteus [7] is a multi-protocol library for integrating multiple message protocols, such as Simple Object Access Protocol (SOAP) and Java Message Service (JMS), within one system while supporting the dynamic addition of protocols. The publish-subscribe system makes use of MOM [14] [17] that acts as a broker, routing published messages for a topic to all subscribers for the topic. MOM is based on an asynchronous model. This allows for application and information dissemination to many users.

C. *Message Oriented Middleware*

MOM is based on an asynchronous model. This allows for application integration and information dissemination to many users. MOM provides a mechanism for integrating applications in a loosely coupled, flexible manner. It acts as an intermediary between end points allowing end points to interact with MOM instead of directly interacting with each other. The MOM infrastructure handles the network communications. If the communication network is temporally unavailable, MOM will store and forward the destined messages once the network connection is reestablished. Another interesting aspect related to the architecture of MOM is that it will only send a message to an actively executing receiving client. If the receiving client is not executing, MOM will hold that message until the application executes. MOM provides assured delivery of messages and makes all of the remote procedure calls (RPC) as well as taking care of network communication protocols.

A message flow graph may route a filtered subset of messages from one information space to another, merge messages from multiple sources or deliver a transformed version of a message from one information space to another. The main reason for the development of MOM is to glue together a large number of stand-alone applications. The MOM environment allows new applications to tap into information generated by existing applications without disturbing them. The fault model that is typically implemented in traditional group communication systems – that a failed or slow process is automatically removed from the group is inappropriate for MOM applications. In MOM the message flow graph is viewed as an abstraction reliable entity. Subscriptions are persistent, and messages may not be lost, permuted, or duplicated, nor must spurious messaged be generated. When a faulty subscriber reconnects, it must be possible to either deliver all the messages that it has missed, or else to compute a shorter set of messages which will re-create this state.

D. *Publish Subscribe*

Publish-subscribe systems provide the capabilities of guaranteeing exactly-once delivery of messages between multiple end points. The system replaces the single destination in a point-to-point model with a content hierarchy, known as topics. The publish–subscribe messaging system works with a “subscribe to topic” architecture, as show in Figure 1. In such a system, clients publish messages with highly structured content, and other clients make available a filter specifying the subscription (the content of the message to be received at that client). These types of networks route based on the data being transported in a message rather than on any specialized addressing and routing information attached to, or otherwise associated with, the message. The publish-subscribe system makes use of MOM that acts as a broker, routing published messages for a topic to all subscribers for the topic.

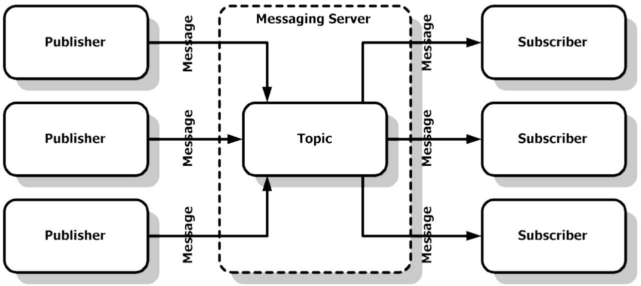

Figure 1 Publish-Subscribe Interaction

Figure 1 shows that each client within the communication domain can act as a publisher and/or subscriber. The client can publish or subscribe to a topic. Multiple clients can subscribe to a topic and multiple clients can publish to a topic.

V. RESEARCH METHODOLOGY

In this paper we describe our research methods as a means of acquiring and constructing knowledge within the boundaries of the research. The research is based on the fusion of three methodologies. The first methodology is based on the common approach for experimental computer science research [18]. This is an iterative approach where the researcher uses the lessons from one cycle to improve the performance in the next cycle. Because the re-development of a SoftBridge system involves the end user, it is appropriate to use this kind of methodological approach. The second methodology is based on the concept of Action Research [1]. This methodology is used to develop applications for targeted user communities and their needs. [5] used the concept of action research to build an automated Deaf Telephony bridging application with a SoftBridge platform. The last methodology is based on the principles of ethnographic research. This research approach gives the researcher the opportunity to conduct rigorous research that has direct practical relevance. The researcher obtains a deep understanding of the people, the organization, and the broader context within which they work [11].

We are currently involved with a literacy training program with Deaf participants at DCCT. The program is in its seventh month of operation. The Deaf participants, involved in the program, were exposed to a combination of literacy programs that consisted of "touch" typing, Internet and e-mail as well as Instant Messaging (IM). There are currently sixteen Deaf participants that went through the training program, since it commenced, and who are now familiar with the technology. We engaged these sixteen Deaf participants in experiments with SIMBA at DCCT. The majority of the experiments consisted of making internal calls to a software telephone (softphone) with the minority of the calls made to landline telephones. Prior to any experiment conducted, at DCCT, each Deaf participant signed a consent form. This form gave us permission to include them in our experiments without exposing their identity. With every experiment and training session conducted, we are realizing that 1) more and more Deaf people are finding the system useful 2) new Deaf individuals visit DCCT to gain knowledge about the system.

With the fusion of the three research methodologies and the experiments and training programs conducted, we are learning every day how we can build a system to actually benefit the lives of Deaf people in South Africa.

VI. TECHNICAL SOLUTION

Studying various communication systems revealed a precise, distinct, solution to the research question:

Move a SoftBridge system into a publish-subscribe domain that facilitates exactly-once reliable message delivery no matter the synchrony.

We have thus integrated a publish-subscribe solution into the existing architecture of SIMBA [15] to accommodate a reliable semi-synchronous messaging system based on a synchronous transport medium for a Deaf Telephony application. The messaging infrastructure can be seen as a network of messaging channels. These channels are responsible for the delivery of messages under any condition. The protocol used by the publish-subscribe system is tolerant to message drops, message reordering, node and link failures. Figure 2 shows the integration of SIMBA and NaradaBrokering. As can be seen by the figure, the inner cloud represents the IP network with the outer cloud representing the boundaries of the Reliable Deliver Service (RDS). The outer cloud thus guarantees the delivery of all IP-related messages.

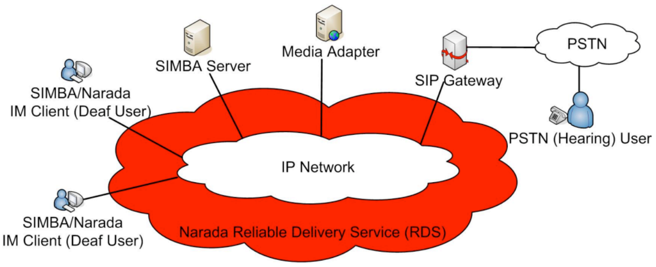
SIMBA Server
Media Adapter
SIP Gateway
PSTN
SIMBA/Narada IM Client (Deaf User)
PSTN (Hearing) User
IP Network
SIMBA/Narada IM Client (Deaf User)
Narada Reliable Delivery Service (RDS)

Figure 2 (Integration of SIMBA and NaradaBrokering)

The main entities of RDS are acknowledgements database(s), entities, profiles and event templates. An event consists of headers, content descriptors as well as the payload (actual message) encapsulating the content. The content descriptors describe information pertaining to the encapsulated content. The content synopsis is comprised of both content descriptors and the values that these content descriptors have. A collection of headers and content descriptors constitute the template of an event. When two or more events contain identical sets of headers and content descriptors they conform to the same template. Entities signify interests in events that conform to a certain template. Each entity associated with RDS has a unique identifier associated with it. Every entity subscribes to this identifier to ensure that messages target to it are routed and delivered by RDS. The RDS component makes use of both positive (ACK) and negative (NAK) acknowledgements for the delivery of messages between entities. RDS provides a reliable delivery service for messages that conforms to one of its managed templates. All published events are archived by RDS under the condition that it conforms to one of its managed templates. For every managed template, RDS also maintains a list of entities for which it facilitates reliable delivery. RDS provides a facility to calculate a valid destination for a give template event. It also keeps track of both the entities that are supposed to receive a given template event, as well as the entities that have not explicitly acknowledged receipt of these events. All event templates are given a unique identifier (templateID) that is used by RDS to advertise its archival capabilities for a specific event template.

For every archived event RDS assigns monotonically increasing sequence numbers that play a crucial role in error detection and correction. When a template event is archived, RDS issues an archival notification which allows subscribing entities to keep track of the template events it has received while facilitating error detection and correction.

The invoice events encapsulate the exchange of information between subscribing and publishing entity as well as RDS. The invoice events correspond to the set of template events received and can also requests to retransmit missed template events. When a subscribing entity receives an archival notification, it checks to see if it has received the corresponding template event. If the template event was received the subscribing entity issues an ACK invoice event. It is possible that the ACK invoice events may get lost in transit to RDS. To prevent this from occurring the entity continues to maintain information about the archival sequence it has received. If the information gets lost, RDS routes those evens which were not explicitly acknowledged using invoice events. When RDS receives the ACK invoice event from the entity, it updates records in the dissemination table associated with the sequence outline in the ACK invoice event to reflect the fact that the entity received template events corresponding to the archival sequences. If RDS detects missing sequences it issues an ACK-response invoice. This invoice contains information related to the entity's sync advancement as well as the sequencing information corresponding to the missed template events. When the entity receives the ACK response invoice event, the entity gains information regarding the archival sequences that it missed. The entity then issues a NAK invoice event requesting the missed events. When RDS receives the NAK invoice event it retrieves the inventory event corresponding to the sequence number and creates a recovery event. It is interesting to note that the NaradaBrokering system is "attached" to the interaction of SIMBA. Looking at Figure 3 it can be seen that RDS is seen as an external entity that monitors all messages sent and received over an IP-based network making use of the UDP protocol for the transport of data.

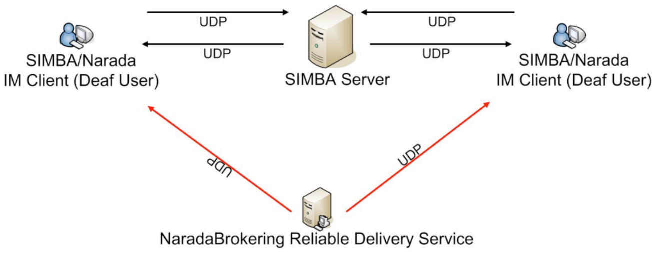
UDP
UDP
UDP
UDP
SIMBA/Narada IM Client (Deaf User)
SIMBA Server
SIMBA/Narada IM Client (Deaf User)
UDP
UDP
NaradaBrokering Reliable Delivery Service

Figure 3 (The NaradaBrokering monitoring process)

The system thus guarantees the delivery of messages over a SIP established session with UDP as its transport protocol. It is quite ironic to think that reliability is brought with the use of an unreliable transport protocol. The entity providing reliability is transparent to the end-user. The user connects to the system as normal via the SIP negotiating stack. At the same time a transparent connection is made to the NaradaBrokering system. After a successful connection, all messages are routed to the RDS component which monitors its process throughout the "life" of the message. The RDS component is responsible for reliably delivering the message to its end destination.

VII. CONCLUSION AND FUTURE WORK

A reliable semi-synchronous Deaf Telephony SoftBridge system that guarantees the delivery of semi-synchronous messages has been built. We are currently in the debugging process with on-going experiments still pending. The Deaf Telephony application forms the test bed to test the overall feasibility of the system. The experiments will be conducted at DCCT involving actual Deaf participants. All experiments will be conducted with the assistance of an interpreter. We will measure the delay metrics that occurs when integrating NaradaBrokering into SIMBA. From a users perspective we expect that interaction with the system will be no different than that of the current version of SIMBA. From an underlying network perspective we expect the system to reliably deliver message under any network and/or user related problems.

REFERENCES

[1] D. E. Avison, F. Lau, M D. Myers & P. A. Nielsen, "Action Research." *Communications of the ACM,* vol. 42 no. 1, pp. 94-97, 1999.

[2] I.O. Agboola & A.C. Lee, "Computer and information technology access for deaf individuals in developed and developing countries". *Journal of Deaf Studies and Deaf Education,* 2000, pp. 286 – 289.

[3] E.W. Biersack, C.J. Cotton, D.C. Feldmeier, A.J. McAuley & D. Sincoskie, "Gigabit networking research at bellcore", *IEEE Network Conference*, vol. 6, no. 2, pp 42-48, 1992.

[4] K.P. Birman, R. van Renesse & W. Vogels, "Spinglass: Secure and Scalable Communication Tools for Mission-Critical Computing", *International Survivability Conference and Exposition. DARPA DISCEX*, 2001.

[5] E. Blake & W. D. Tucker, "Bridging Communications Across the Digital Divide", *Proc. of Social Intelligence Design (SID)*, University of Twente, Enschede, the Netherlands, pp. 29-38, 2004.

[6] A. Carzaniga & A.L. Wolf, "Content-Based Addressing and Routing: A General Model and its application", *Technical Report CU-CS-902-00*, Computer Science Department, University of Colorado, 2000.

[7] K. Chiu, M. Govindaraju & D. Gannon, "The Proteus multiprotocol message library", *Proc. of the ACM/IEEE conference on Supercomputing*, pp. 1-9, 2002.

[8] M. Glaser & W. D. Tucker, "Telecommunications Bridging between Deaf and Hearing Users in South Africa", *Proc. of the Conference and Workshop on Assistive Technologies for Vision and Hearing Impairment (CVHI)*, Grenada, Spain, CD-ROM publication, 2004.

[9] G. Hellstrom & P. Jones, Network Working Group Internet Draft, "RTP Payload for Text Conversation", 2004.

[10] J. Lewis, W. Tucker & E. Blake, "SoftBridge: A Multi-modal Instant Messaging Bridging System", *Proc. SATNAC*, George, South Africa, CD-ROM publication, 2003.

[11] M. D. Myers, "Investigating information systems with ethnographic research", *Communication of the AIS,* vol. 2 no. 23, pp. 1 – 20, 1999.

[12] S. Pejhan & M. Schwartz, "Error control using retransmission schemes in multicast transport protocols for real-time media". *IEEE/ACM Transaction on Networking*, vol. 4 no. 3, pp 413 – 427, 1996.

[13] S. Pallickara & G. Fox, "NaradaBrokering: A Distributed Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids". *Proceedings of the ACM/IFIP/USENIX International Middleware Conference*, Pp. 41 – 61, 2003.

[14] V. Quema, "Application-Driven Customization of Message-Oriented Middleware for Consumer Devices." In *Workshop on Software Infrastructure for Component-Based Applications on Consumer Devices,* 2002.

[15] T. Sun & W. D. Tucker, "A SoftBridge with Carrier Grade Reliability Using JAIN SLEE", *Proc. SATNAC*, Stellenbosch, South Africa, pp. 251-252, 2004.

[16] H. Schulsrinee & S. Casner, R. Frederick, V. Jacobson, "RTP: a transport protocol for real-time applications". *Working Group Draft, IETF AVT*, 1994.

[17] P. Tran & P. GreenField, "Behavior and performance of Message-Oriented Middleware systems", *Proc. of the 22nd IEEE international conference on distributed computing systems workshops (ICDCSW'02)*, 2002.

[18] National Academy of Sciences, "Academic Careers for Experimental Computer Scientists and Engineers", 1994.

Elroy Julius is a Telkom Masters Student in the Department of Computer Science at the University of the Western Cape. His research focuses on the guaranteed delivery of semi-synchronous IP-based communication.

William Tucker is a Senior Lecturer in Computer Science at the University of the Western Cape. He is finishing a PhD at the University of Cape Town on delay and multi-modal semi-synchronous IP communications.