Guaranteed delivery of multimodal semi-synchronous IP-based communication.

by

Elroy Julius

A thesis submitted in fulfillment of the requirements for the degree of Magister Scientiae in the Department of Computer Science, University of the Western Cape.

Supervisor: Mr. William D. Tucker

KEYWORDS

Asynchronous

Deaf Telephony

Forward Error Correction

Guaranteed Delivery

Internet Protocol

Message Oriented Middleware

Multimodal

Publish-Subscribe

Reliable Communication

Synchronous

ABSTRACT

Guaranteed delivery of multimodal semi-synchronous IP-based communication.

E.P. Julius

MSc Thesis, Department of Computer Science, University of the Western Cape.

A semi-synchronous environment consists of a combination of synchronous and asynchronous transport media used to transport messages from source to destination. This thesis explores an empirical solution for guaranteeing the delivery of messages in a multimodal semi-synchronous environment for a Deaf Telephony application. SoftBridge for Instant Messaging Bridging Application (SIMBA), is a communication platform that make use of a semi-synchronous framework to allow a hearing and Deaf person to communicate inside a single uniform space. SIMBA was modified to provide reliability for both synchronous and asynchronous transport media. In the process of modification, SIMBA was renamed to NaradaBrokering integrated in SIMBA (NIMBA). Within the literature various systems are analyzed and successes and failures are distilled to help formulate a good solution for the thesis question. To guarantee asynchronous messages sent, the Message Oriented Middleware (MOM) paradigm was used with Forward Error Correction (FEC) used to guarantee the delivery of synchronous messages sent. The work forms part of a social study conducted at the Deaf Community of Cape Town (DCCT) community centre. Ethnography was used to identify the requirements for a Deaf Telephony application. Thus, this thesis is based on a socio-technical environment where the system is developed in a laboratory and tested in an actual community. Results show that these solutions are acceptable for a semi-synchronous communication environment. However, interviews conducted with a select group of Deaf participants showed that cell phones are currently too popular to be replaced by NIMBA as a primary communication device and service. This is due to the immense popularity of Short Message Service (SMS) among the Deaf and cell phone devices mobile capabilities. The overall goal of the thesis is to guarantee delivery for a semi-

synchronous environment with broader implications of showing how multimodal semi-synchronous services like Deaf Telephony can be made attractive to service providers concerned with reliability in the new Internet Protocol (IP) world of telecommunications.

December 2005.

DECLARATION

I declare that *Guaranteed delivery of multimodal semi-synchronous IP-based communication* is my own work, that it has not been submitted for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged by complete references.

Full name…………………………………… Date……………………………………

Signed…………………………………….....

ACKNOWLEDGEMENTS

A plethoric amount of gratitude, love and appreciation I give to my two loving parents whose love and appreciation I enshrine more than anything in life. To Priest Galant I bestow a great deal of gratitude and appreciation for the wisdom and guidance you gave me throughout my life and academic career. To my sister and brother in law, I thank from the bottom of my heart for their wisdom, love and kindness. To a great supervisor and mentor, I thank not just for supervision but for super understanding. Bill, it was an honor and privilege to be guided by your wisdom and knowledge. To the Telkom/Cisco/Siemens Centre Of Excellence (COE), I thank for the financial sponsorship and computer equipment. To Phadlie Nordien, I thank for always taking the time off to talk nonsense. It was during these conversations brain storming occurred and if this thesis should ever be awarded a Nobel peace prize, you may collect it. To Xolisa Vuza, I thank for all the encouragement throughout my MSc. To Michael-John, I thank for always taking my mind off work and showing me a different side of life. To all my friends and family, too many to mention, I thank for filling that space in my life and always sharing some story or another. To my laboratory colleagues, I express great appreciation. A special thank you to Sun Tao for his friendliness and willingness to help. To the only “Afrikaans” speaking, Chinese looking comrade I know, I thank for always being such a comic. Yanhao, never change for anyone. To the staff of the Computer Science department, I thank for six years of education and a ticket to a better future. To Auntie Lilly, I bow down and thank for always taking good care of me and always walking that extra mile to make my day a bit easier. Sorry though for always leaving my desk so dirty. To anyone and everyone that believed in me, I say: “thank you very much!”

CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ACE	Australian Communication Exchange
ACK	positive acknowledgement
ADSL	Asymmetric Digital Subscriber Line
API	Application Programming Interface
APP	APPlication specific extension
ARQ	Automatic Repeat reQuest
ASR	Automatic Speech Recognition
CBSE	Component-Based Software Engineering
CC	CSRC Count
CSRC	Contributing SouRCe identifier
DCCT	Deaf Community of Cape Town
FCC	Federal Communications Commission
FEC	Forward Error Correction
GUI	Graphical User Interface
ICT	Information and Communication Technology
IESG	Internet Engineering Steering Group
IETF	Internet Engineering Task Force
IM	Instant Messaging
IP	Internet Protocol
IRC	Internet Relay Chat
IS	Information Systems
ISDN	Integrated Services Digital Network
IT	Information Technology
JMS	Java Message Service
JORAM	Java Open Reliable Asynchronous Messaging
Kbps	Kilobits per second
LAN	Local Area Network
LSB	Least Significant Bit

MAN	Metropolitan Area Network
MAS	Media Adapter Server
MECA	Multi-media End-to-end Communication Architecture
MMS	Multimedia Messaging Service
MOM	Message Oriented Middleware
ms	milliseconds
MSB	Most Significant Bit
MUD	Multi User Dungeons
NAK	Negative acknowledgement
NDIS	Network Driver Interface Specification
NGO	Non Governmental Organization
NIC	Network Interface Card
NIMBA	NaradaBrokering integrated in SIMBA
NPNT	Non-Persistent Non-Transactional
NRS	National Relay Service
PC	Personal Computer
PCM	Pulse Code Modulation
PFS	Persistent Filtering Subsystem
PGT	Persistent Global Transaction
PHB	Publishing Hosting Broker
PLT	Persistent Local Transaction
PNT	Persistent Non-Transactional
PSTN	Public Switched Telephone Network
PT	Payload Type
QoS	Quality of Service
RDS	Reliable Delivery Service
RPC	Remote Procedure Call
RR	Receiver Report
RTCP	Real Time Control Protocol
RTD	Round Trip Delay

RTP	Real Time transmission Protocol
SASL	South African Sign Language
SDES	Source DEScription
SHB	Subscriber Hosting Broker
SIMBA	SoftBridge for Instant Messaging Bridging Application
SIMPLE	SIP for Instant Messaging and Presence Leveraging Extensions
SIP	Session Initiation Protocol
SMS	Short Message Service
SOAP	Simple Object Access Protocol
SR	Sender Report
SSRC	Synchronization SouRCe identifier
TCP/IP	Transmission Control Protocol over Internet Protocol
TPDU	Transport Protocol Data Unit
TTS	Text To Speech
UCT	University of Cape Town
UDP	User Datagram Protocol
UWC	University of the Western Cape
VoIP	Voice over IP
VRS	Video Relay Service
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WWW	World Wide Web
XML	Extensible Markup Language

Chapter 1 INTRODUCTION

This thesis investigates an empirical solution for guaranteeing the delivery of semi-synchronous messages within the context of a Deaf Telephony application. The term semi-synchronous refers to a mixed interchange of synchronous and asynchronous forms of communication. Currently, Instant Messaging (IM) presents a good example of semi-synchronous communication as it appears to be synchronous when messages are delivered quickly enough, but is actually asynchronous (O'Neill and Martin 2003). However, in a semi-synchronous environment, messages traverse through both synchronous and asynchronous transfer protocols. A good example is Deaf Telephony. Deaf Telephony converts a Deaf person's text or video, received from an IM session for example, to audio across both Public Switched Telephone Network (PSTN) and Internet Protocol (IP) networks (Glaser and Tucker 2004). Therefore, Deaf Telephony is also multimodal (Figure 1.1).

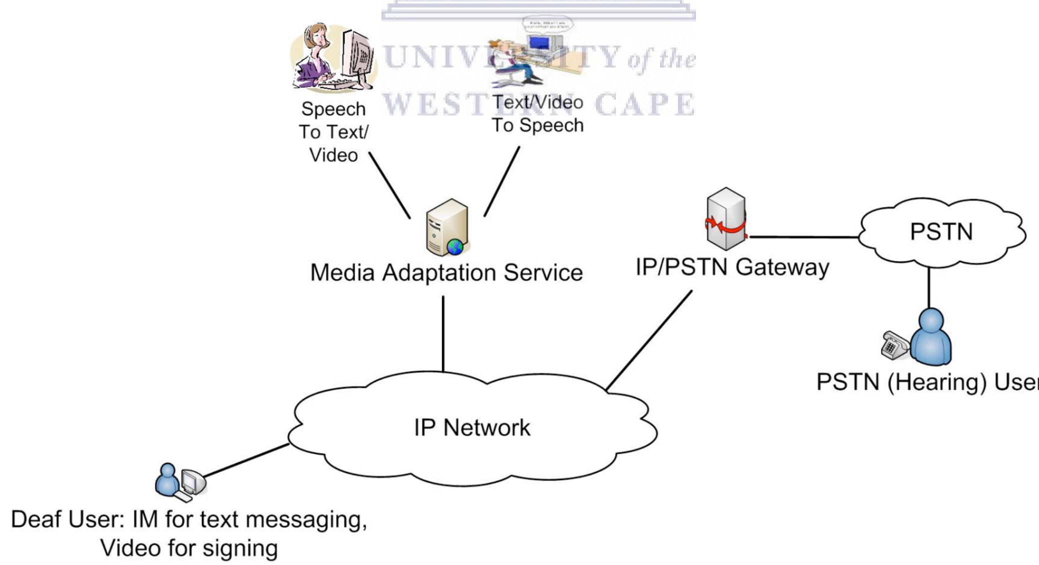

Figure 1.1 Deaf Telephony as an example of IP-based communication.

A Deaf person's message, whether text or video, is converted via a media adaptation service for transport across the PSTN network. In reverse, a hearing person's audio message is converted into the correct medium for the Deaf person's end user service.

This thesis explores how hearing and Deaf users are brought together into one communication space where interaction between them is a semi-synchronous form of message exchange. The focus of this thesis is the means by which message delivery between two entities is guaranteed. The underlying communication system has the arduous task of guaranteeing message delivery between end points making it transparent to the end user. Amid the process to reconcile synchronous and asynchronous communication, an assistive communication service is built for a Deaf community. This communication service guarantees the delivery of both synchronous and asynchronous messages between hearing and Deaf users using the Deaf Telephony application. Recognizing the disadvantage of the Deaf[1] community with Information and Communication Technology (ICT), the thesis aims to position both communities into an "equal" communication and information space. This positioning helps to narrow the digital divide between advantaged and disadvantaged communities. Even though a community demarcation exists, the hearing and Deaf communities still need to communicate with each other reliably and in as real-time as possible.

1.1 Deaf background

The Deaf have both a community and a culture. Both have been in existence for a long time with the culture having characteristics identifiable to that of other subcultures or ethnic groups. The Deaf community has events ranging from athletics to community picnics and theatrical performances. Interestingly enough, the majority of Deaf children are born to hearing parents making the passing on or transmission of Deaf culture not familial but from the contact with other Deaf people in the community *(www.deafsa.co.za)*.

[1] The term "Deaf", spelled with a capital "D", refer to the members of a Deaf community that use a given sign language as their primary mode of communication.

1.1.1 The Deaf in South Africa

In South Africa the deaf as opposed to Deaf community constitutes 10 percent (4 million) of South Africa's population *(www.deafsa.co.za)*. This community varies from Deaf to hard of hearing, different levels of hearing loss, those that become Deaf at a later age and those that become Deaf as a result of noise pollution. It is estimated that around 1.5 million Deaf people in South Africa use South African Sign Language (SASL) as a first language. In South Africa it is estimated that 30 percent of all Deaf adults are illiterate as a result of low academic achievement (Asmal 2001; Glaser and Aarons 2002). Most Deaf children attend school at a late age (Asmal 2001; Kiyaga and Moores 2003) with education for Deaf children only made compulsory in 1996 (Aarons and Akach 2002). In order for us to fully understand the needs of the Deaf, an ethnographic study was conducted with a small group of Deaf people at the Deaf Community of Cape Town (DCCT) at the Bastion centre in Cape Town.

1.1.2 The Deaf Community of Cape Town

The profile of the Deaf community in the Western Cape is one of high levels of poverty, unemployment and poor education. In 1987, DCCT emerged as a nongovernmental welfare organization (NGO) whose perennial aim is to address the needs of Deaf people in the Western Cape (Lombard 2005). The organization is made up of 1090 active Deaf members who all help steer the organization. Currently, DCCT employs Deaf workers within a sewing project in which funds are generated from the sale of the goods produced. There are various other projects undertaken by DCCT aimed at the development and empowerment of Deaf people. Some of the projects include a welfare program with professional work services. These services include casework, groupwork and community development, burial fund, Deaf awareness program, job placement and training program and an HIV/AIDS program. The most notable project related to this thesis is the ICT training program. This program is an initiative undertaken in collaboration between the University of the Western Cape (UWC), the University of Cape Town (UCT) and DCCT (Glaser, Young and Porteous 2005). The aforementioned projects undertaken by DCCT are all initiated and developed for and by Deaf people. DCCT is recognized as a formidable organization which is strongly supported and backed by the Deaf community. This is one of the reasons why DCCT was selected as the facility to provide a test bed for a Deaf Telephony application that provides reliable communication.

1.2 Deaf Telephony

Deaf Telephony allows a Deaf person, with an asynchronous device and interface, to communicate with a hearing person with a synchronous device connected to the PSTN. Essentially, Deaf Telephony translation can be facilitated by a human relay operator or it can be fully automated. In South Africa, the provision of Deaf Telephony services is challenging (Tucker, Blake and Glaser 2003). This is due to the fact that most Deaf people in South Africa are undereducated, as mentioned earlier. At the moment the Teldem, the Short Message Service (SMS), IM, and e-mail are the only communication modalities available to the Deaf user in South Africa. From all these options, the Teldem is the only device that currently provides synchronous communication without the aid of the Internet (Glaser 2000).

1.2.1 TTY

A TTY is a text telephone device that is used in conjunction with a telephone line to communicate with persons who are Deaf, hard of hearing, or have speech impairments, by typing and reading text. The South African telecommunications provider, Telkom, has produced such a TTY device for the Deaf community called a Teldem (Glaser 2000). To communicate by Teldem, a person types his or her conversation, which is read on a TTY display by the person who receives the call. When typing on the Teldem, each letter is transmitted by an electronic code called Baudot, which is sent from the Teldem on the sending end of the call through the telephone line in the form of tones to the Teldem on the receiving end of the call, the same way voiced communication occurs between two parties. The receiving Teldem transforms the tones back to letters on a small two line display screen. Communication between two persons using Teldems can only occur in half duplex.

Thus, both persons who are conversing cannot type to each other at the same time; they must take turns sending and receiving. The Teldem has not really reached any form of popularity among the Deaf community. This is due to 1) the absence of a relay service to the PSTN, 2) the device having half-duplex throughput capabilities which essentially provides a single-channel, one-way form of communication and 3) the tremendous popularity of SMS among the Deaf (Glaser and Tucker 2004).

1.2.2 SMS

Even though SMS cannot substitute for real-time synchronous communication, it is still the most widely used communication service among the Deaf (Nelson and Underschultz 2003). In November 2004 Telkom released a service for landlines to fuel the existing price warfare between Telkom and South Africa's three cellphone networks (Phasiwe 2004). The service allows Deaf people to access their text messages on both cellular and fixed-line telephones. The only drawback of this service is that the Deaf needs to purchase special handsets, and still pay for individual SMS's. For the Deaf, SMS communication between Deaf and Deaf is just as easy as communication between two ore more hearing individuals. The problem arises as soon as a Deaf person communicates with a hearing person via SMS. A Deaf person's grammar is different to that of a hearing person. In most cases a hearing person does not understand a Deaf person's SMS. This is because the Deaf uses a short hand description of messages that is widely understood among the Deaf. Even though SMS allows one to be mobile, it still can't make up for the fact that it's asynchronous in nature and provides no feedback whether the recipient actually read the message or not. Thus in situations where synchronous acknowledged communication is required, SMS fail to deliver.

1.2.3 Relay services

The ability to select from among many communication modes – voice, text, or video – enables users with disabilities who can perform some functions but not others to choose the telecommunication mode best suited to the needs and circumstances of their conversations. When Deaf users have the same communication device or service, such as video or text, no relay services are required. However, when Deaf users use different devices/services to communicate with each other or even hearing users, a relay service is required. This relay service allows different end user devices/services to interact with each other in a seamless fashion. Relay services involve a communications assistant that uses both a standard telephone and a TTY device, such as the Teldem, to type voice communication to a TTY user and read a TTY user's typed communication to a voice telephone user *(http://www.metrokc.gov/dias/ocre/relay.htm)*. Examples of relay services over the PSTN include the Royal National Institute for the Deaf's Typetalk at British Telecom *(www.typetalk.org)*, Talking text *(www.talkingtext.net)*, AT&T TTY Relay Service *(www.consumer.att.com/relay)*, Australian Communication Exchange's (ACE) National Relay Service (NRS), and Sprint Relay *(www.sprintrelayonline.com)*. In South Africa however, relay services to the PSTN does not exist since it requires provider subsidization.

Voice Relay

Voice relay services places a human relay operator equipped with both text and telephone capabilities between a Deaf user, using a TTY device, and a hearing user with a telephony handset. The relay operator then converts a Deaf persons text to speech, relays the speech to the hearing user, and vice versa (Tucker, Glaser and Lewis 2003). In order for voice relay services to work properly some degree of literacy is expected on the Deaf side of communication since the relay operator needs to read and process a Deaf person's text message before it is relayed to a hearing user.

Video Relay

IP-video is already revolutionizing communications for some people with disabilities. In March of 2000, the Federal Communications Commission (FCC) approved Video Relay Service (VRS), a service that enables Deaf people to use video links and sign language interpreters to communicate in sign language with hearing users on a telephone *(http://ftp.fcc.gov/cgb/consumerfacts/videorelay.html)*. These services make use of high speed Internet services, enabling people with hearing disabilities whose first language is sign language the opportunity to enjoy telephone services that are truly functionally equivalent to conventional voice telephone services. Broadband services coupled with IP video devices and software, permit point-to-point video communications among signing Deaf, hard of hearing, and hearing people which allows them to have natural telephone conversations. Video communication allows people who are hard of hearing and use speech and hearing for communication to see the lips and facial expressions of the speaker to better understand what is being said. Similarly, video communications are also useful to people who have speech disabilities but insufficient motor skills to type; their speech could naturally be augmented by visual cues such as facial expression, gesture, and the use of speech communication aids. Examples of video relay services include Visicast *(www.visicast.co.uk)* with Tessa (Cox *et. al* 2002). This thesis, however, does not address video relay services since the Deaf Telephony application utilized currently only semi-automated text relay.

1.3 The Digital Divide

Technology in the field of telecommunications has reached great advancement compared to a couple of decades ago (*http://www.aad.org/au/download/affordibility.pdf).* 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". The majority of the Deaf community suffers from a lack of exposure to ICT. The digital divide encompasses a lack of access to ICT in conjunction with existing challenges such as low literacy, lack of financial resources and a lack of trained technical staff within a third world country (Agboola and Lee, 2000). The divide is characterized by a shortage of computer hardware and software, access and training which leads to a lost opportunity for ICT experience. The digital divide exists because of a lack of knowledge and understanding with regards to the use and availability of ICT as well as the inaction of effective government policy. (Penton, Tucker and Glaser 2002) tried to narrow the gap between the Deaf and hearing communities by building a system called Telgo323. The system bridged between Deaf and hearing, voice and text, as well as the PSTN and the Internet.

1.4 The SoftBridge Approach

Deaf Telephony services need to interact with a wide spectrum of devices with varying networking capabilities. SoftBridge is an example of a synchronous mixed media space that allows end users with different media capabilities to communicate with each other in a uniform manner (Lewis, Tucker and Blake 2002; Lewis, Tucker and Blake 2003; Blake and Tucker 2005; Blake and Tucker 2004). The aim of SoftBridge is to provide a framework for multimodal bridging as well as multi-user, multimodal conversation sessions. The SoftBridge concept was first prototyped by John Lewis at the UCT (Lewis, Tucker and Blake 2002). An implementation called SoftBridge for Instant Messaging Bridging Application (SIMBA) was subsequently developed (Sun and Tucker 2004).

1.4.1 The first SoftBridge prototype

The first SoftBridge prototype was based on the Jabber protocol *(www.jabber.org)*, which allowed communication to occur asynchronously. Semi-synchronous interaction between Deaf and hearing end users could occur through media adaptations that SoftBridge performs according to the media capabilities of the end user. SoftBridge would, for example, allow end users with different media capabilities, such as voice and text, to communicate with each other within a single uniform media space (Figure 1.2).

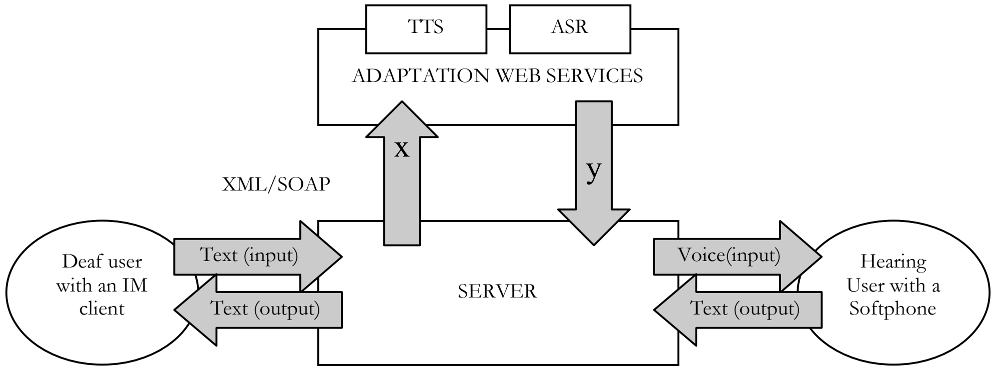

Figure 1.2 SoftBridge data flow

The first SoftBridge prototype allowed end-users to specify their media capabilities by selecting their input and output media type. The adaptation web service then chose the correct media conversion driver and converted the input stream to the output stream matched by the receiving user's output capability (Lewis et. al 2002). As an example, the Deaf user specified his/her input and output capabilities as text with the hearing user's input and output specified as voice. When a Deaf person sends a text message to a hearing user the media adaptation web server would convert it to an audio format for the hearing user to receive. The same happens in reverse. The hearing user sends a voice message and because the Deaf user specified a text input capability, the web server will convert the audio message to text.

All message transfers are based on web services using the Simple Object Access Protocol (SOAP), Extensible Markup Language (XML) and the Jabber IM protocol. The voice media is encapsulated in an XML packet. Within SoftBridge the adaptation services are provided by web services that provide Text To Speech (TTS) and Automatic Speech Recognition (ASR). These are interchangeable components due to the web service architecture.

1.4.2 SIMBA

A completely different SoftBridge prototype, SIMBA, was developed as a Deaf Telephony application that focused on providing life-cycle management, a carrier grade characteristic, for an IP-based application (Sun and Tucker 2004). The system was trialed between Deaf and hearing participants at DCCT during the period of March until October 2005. The system allows a Deaf and hearing person to communicate with each other inside a single communication space while bridging between the PSTN and IP network. Media adaptation, such as TTS conversion or text relaying is taken care of by a Media Adapter Server (MAS) which is a modified Tomcat *(http://tomcat.apache.org)* web server with tightly coupled web services that perform media adaptation. A Deaf person uses an IM client to send a text message to a hearing person. SIMBA forwards the text message to the MAS to convert the text message in audio with a TTS engine. From there, SIMBA sends the audio message to the hearing client. If situated on the PSTN, SIMBA uses a Session Initiation Protocol (SIP)-based gateway. The gateway acts as an intermediary between the IP and PSTN networks. If on an IP network, SIMBA simply uses SIP to establish a call to a SIP phone. All audio messages received from a hearing client, are forwarded to the MAS which in turn converts it to text with a human relay operator to improve speech recognition. The relay operator listens for incoming audio, converts it to text and uses an IM client to send the converted text to the Deaf person. Even though SIMBA provides a semi-synchronous environment for the exchange of messages between Deaf and hearing persons, it still has some shortcomings.

1.4.3 Delay Issues and SIMBA

Unreliable communication is a detractor when providing real time communication. Network delays are one of the primary concerns related to data transfer within an IP-based network. The delays that occurred during the preliminary Deaf Telephony trails came from media conversion overhead. SoftBridge attempts to deal with the delay by using presence cues because macro delays affect the target Deaf telephony end users on a larger scale than micro delays (Tucker, Blake and Marsden 2004). Most of the Deaf or hearing impaired people doesn't have access to personal computers. They have to physically travel to DCCT to access public computers. In most cases, there are set times for using these computers. With the possibility of the receiving party not being online or the communication "line" being busy as well as a fix time allocation per computer, communication delay can easily increase from a couple of milliseconds to a couple of days. Thus, the total macro delay is measured in terms of propagation delay, media conversion as a result of TTS processing and text relaying, the time taken to type a message by both Deaf user and relay operator as well as the time taken to travel to DCCT. What the Deaf community needs is a communication infrastructure that will reliably deliver messages whether the destination party is online or not. This is especially important when macro delay is no longer measured in terms of milliseconds but in terms of days of even weeks.

1.5 Motivation

The motivation behind this thesis, then, is the need to derive methods for guaranteeing combined asynchronous and synchronous communication. These methods will allow SIMBA to guarantee exactly-once, reliable delivery of messages within a semi-synchronous environment. The semi-synchronous environment provides a real time and store-and-forward based communication framework. A Deaf person's IM messages traverse through an asynchronous medium, are converted to audio by the MAS and finally sent via a synchronous medium to a hearing user. Thus vulnerability for data loss is created for both synchronous and asynchronous communication. Besides the Deaf, the hearing user is also affected by the loss of data. This thesis, however, only looks at the Deaf and their perception of message loss and delivery. The Deaf community needs a communication platform that guarantees the delivery of messages between source and destination. Our targeted groups of Deaf or hearing impaired individuals make use of the DCCT community center to gain access to a computer for a limited amount of time. In many instances they find that the communication party is not available due to him/her being off-line. The Deaf or hearing impaired individual should not be challenged with the difficulty of delivering messages to its recipient. In fact the delivery of messages should entirely rely on the underlying network and its transport medium. The Deaf community needs a communication platform that guarantees the delivery of messages no matter the synchrony.

1.6 Research question

The main research question is: "How does one guarantee exactly-once delivery of mixed synchronous and asynchronous messages within a Deaf Telephony application?" Thus guarantees should be provided for both asynchronous text messages and synchronous audio communication.

1.7 Research aim and significance

Exploring the mixed synchrony for communication interactions revealed several key areas of enhancement related to the architecture of SIMBA. The aim of the thesis is to find a relationship between asynchronous and synchronous call handling. In particular, the thesis wishes to address a mixture of SIP (Rosenberg *et. al* 2002) and Sip for Instant Messaging and Presence Leveraging Extensions (SIMPLE) (Campbell *et al.* 2002). The SIP established session will be used to transfer synchronous as well as asynchronous data between end users of the system.

Even though IM is perceived to be an asynchronous form of communication, it is interesting to note that at times, during communication, it appears synchronous. IM has the ability to appear synchronous when communication between two parties occurs quickly. The current architecture of SIMBA facilitates different modes of communication, combining IM and Voice over IP (VoIP). These modes include audio and text. The asynchronous/synchronous relationship allows SIMBA to switch between asynchronous call handling and synchronous call handling according to the end-users device capabilities. The thesis thus focuses on finding ways of guaranteeing reliable message delivery in a semi-synchronous environment.

1.8 Thesis outline

The remainder of this dissertation is organized as follows.

Chapter 2 - Literature review: This chapter presents related work in the field of reliable synchronous and asynchronous communication. For error control in synchronous communication, three main techniques are looked at and discussed. To guarantee the delivery of asynchronous messages, various systems that incorporate positive and negative acknowledgements are examined. These approaches are joined together and a proposed solution is suggested to handle the mix of synchronous and asynchronous communication found in a Deaf telephony application.

Chapter 3 – Research methodology: In this chapter two methodologies for the software implementation process are listed and discussed. These are exploratory prototyping and Component Based Software Engineering (CBSE). Furthermore, to situate the process, ethnography is employed. Finally, a technique for both synchronous and asynchronous communication is evaluated by the use of automated software agents.

Chapter 4 – System design: This chapter gives a detailed discussion of how synchronous and asynchronous reliability was implemented. For asynchronous communication, acknowledgements within a Message Oriented Middleware (MOM) framework are used between a Deaf IM client and SIMBA to ensure that messages are reliably delivered and persistently logged. For synchronous messages a Forward Error Correction (FEC) algorithm is used, based on Vandermonde matrices, to ensure real-time audio is delivered between hearing and Deaf individuals.

Chapter 5 – Experimental setup: This chapter presents a detailed discussion of the experimental design used to evaluate the overall performance of the developed system/software. The discussion includes laboratory evaluation procedures for reliability in terms of synchronous and asynchronous forms of communication. The overall system is also briefly evaluated within an end-user environment in an actual Deaf community.

Chapter 6 – Data collection and results: This chapter presents results obtained from experiments and interviews conducted in a laboratory and a Deaf community respectively. The discussion of results includes measurements obtained for the implemented software solution as well as ethnographic findings obtained from social interaction with Deaf people under study.

Chapter 7 – Conclusion and future work: This chapter succinctly concludes and summarizes the most important points throughout this thesis. The overall contribution of this thesis is presented and a bigger “picture” is presented in which to situate the research. Suggestions for future work conclude the thesis.

Chapter 2 LITERATURE REVIEW

In the previous chapter we identified the problem of reliable communication and its impact on the Deaf community. This chapter examines ways of guaranteeing message delivery over an IP-based communication network. The focus of this chapter is primarily the goal of achieving reliable data exchange that guarantees the delivery of synchronous and asynchronous messages. The desired output is to learn how other work provides both synchronous and asynchronous guarantees in order to apply these techniques to a semi-synchronous environment like a Deaf Telephony.

This chapter examines reliable communication by looking at the overall contributing factors such as synchrony and middleware. This provides a survey of related work in the field of reliable synchronous and asynchronous communication. This chapter, then looks at various systems and a focus is placed on problems, shortcomings and successes identified by these systems. These techniques will then be used to inform the approach to answer the research question regarding guarantees in semi-synchronous environment.

2.1 Reliable Synchronous communication

When traffic is sent and received within a demarcated interval with the absence of timing faults we have a synchronous form of communication. An example of synchronous communication is a telephone conversation over the PSTN which provides a real time medium for communicating entities. Synchronous communication handles data more efficiently and with less overhead than asynchronous communication. As an example, Integrated Services Digital Network (ISDN) lines use a synchronous technique for data transfer opposed to the modem's asynchronous technique. This is because asynchronous communication sends small blocks of data with lots of control bits for error correction, while synchronous communication uses big blocks of data with control bits only at the start and the end of the entire transmission. This is one of the reasons why synchronous communication is much faster than asynchronous communication. Because synchronous communication does minimal error checking, it needs a "clean" line for data transfer. A disturbance on the line of connection can throw the data on the line out of sync. In the Internet, packets carrying real-time data may be dropped or arrive too late to be useful because the Internet is a packet-switched, best-effort delivery service, with no guarantee on the Quality of Service (QoS). Most audio and video compression algorithms are not robust to transmission errors as their sole objective is to maximize coding gain, assuming error-free channels (Wah, Su and Lin 2000). The same can be said for low bit-rate speech systems that incorporate recursive filters to remove as much redundancy as possible without considering error resilience. The speech frame's loss results in the degradation of the lost frame as well as subsequent frames (Shah *et al*. 1996).

2.1.1 Internet Telephony

In the last decade, the market for and the use of VoIP and other streaming media applications in the Internet have grown exponentially (Lockhart and Lara-Barron 1991). As these services are deployed within the context of a best-effort IP network, it is necessary to provide these media streams with protection from the worst effects of packet loss. The Internet Engineering Task Force (IETF) has defined a number of playout formats that extend the Real Time Transmission Protocol (RTP) (Schulzrinne *et. al* 1994) to provide a degree of error control (Perkins *et. al* 1997). Bolot and Garcia (1996) proposed a rate control mechanism that attempts to minimize the number of packets lost by making sure that the rate at which audio packets are sent over a connection matches the capacity of the connection. This, however, does not prevent loss altogether. A loss recovery or error control mechanism is required when the number of lost audio packets is higher than that tolerated by the listener at the destination.

2.1.2 Error control in real time communication

A good basis for analyzing the performance of error control in synchronous communication includes the following parameters (Pejhan and Schwartz 1996):

1. *The probability of dropping a packet.* Even though packets may be retransmitted as many times as required in order to deliver them correctly, the probability of dropping a packet needs to be considered.
2. *The average time required to deliver a packet correctly*. This is an important constraint for synchronous communication.
3. *The average number of transmissions*. This parameter is concerned with the average number of times that a packet has to be retransmitted.

A major QoS requirement for real-time media, such as video and audio, is to deliver packets reliably to the end receiver(s) in a timely fashion. Satisfying this requirement becomes more challenging as the distance between the end points increases. Packet loss recovery can occur at both receiving and sending sides of transmission (Perkins, Hodson and Hardman 1998).

2.1.3 Sender-based packet loss recovery

Sender-based packet loss recover techniques require the participation of the sender of an audio stream to recovery from packet loss. These techniques are split into two classes: active retransmission and passive channel coding (Figure 2.1).

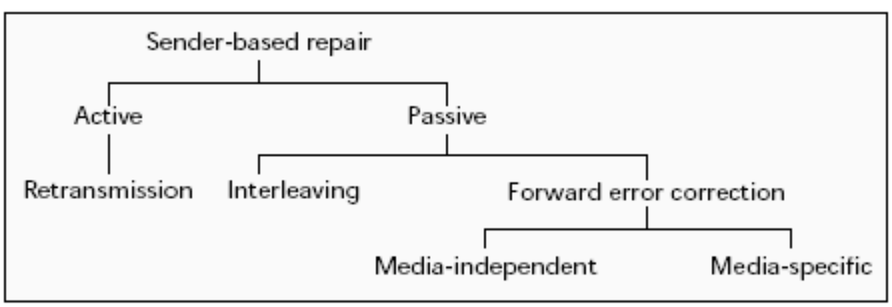

Figure 2.1 Taxonomy of sender based repair techniques

The two major classes of sender-based repair schemes: active retransmissions and passive channel coding. The channel coding techniques are further subdivided into FEC and traditional interleaving-based schemes (Perkins, Hodson and Hardman 1998).

.

The channel coding techniques are further subdivided into Forward Error Correction (FEC) and interleaving-based schemes. FEC data can either be media independent, which is based upon exclusive-or operations, or it can be media specific, that is based on the properties of an audio signal (Perkins, Hodson and Hardman 1998).

2.1.3.1 Forward Error Correction

In FEC, there are two classes of repair data that may be added to an audio stream: data that are independent of the contents of the stream, and data that uses knowledge of the stream within the repair process.

Media Independent FEC

This scheme is media independent because the operation of FEC does not depend on the contents of the packets. The repair process replaces an exact copy of the lost packet regardless of the media. The computation required for the derivation of the error correction packets is relatively small, and implementation is simple compared to other error correction schemes. The disadvantages of media independent FEC are the additional delay imposed, and the requirement of an increase in available bandwidth due to the amount of redundant information for each packet in subsequent packets. Figure 2.2 graphically depicts the repair process within the media independent FEC scheme. The scheme uses algebraic codes to produce additional packets for transmission that aid in the process of packet loss recovery. Each code uses k data packets to generate n-k additional packets (used for error checking) for the transmission of n packets over the network.

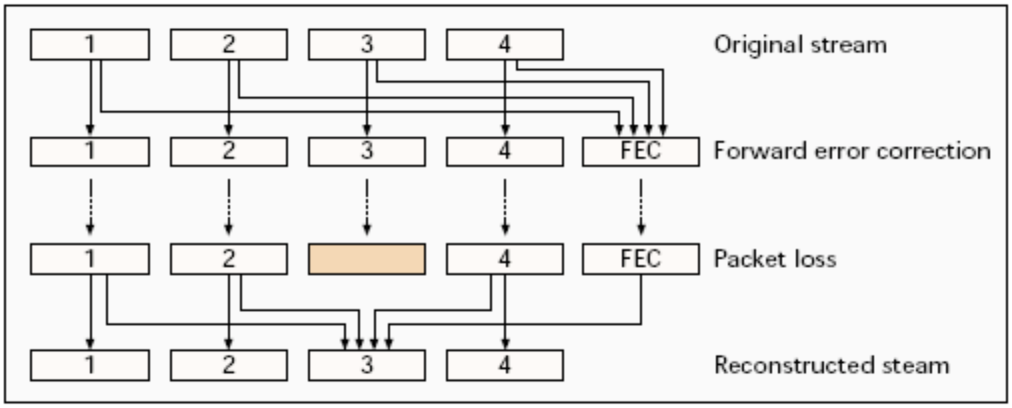

Figure 2.2 Media independent FEC repair process

In parity coding techniques, the exclusive-or (XOR) operation is applied across groups of packets to generate corresponding parity packets. This scheme is successful provided that there is just one loss for a certain amount of packets transmitted (Perkins, Hodson and Hardman 1998).

Media Specific FEC

Media specific FEC schemes transmit each unit[2] of audio in multiple packets. When packet loss occurs, another packet containing the same unit is used to recover from the loss (Figure 2.3). The first copy of the audio data that is transmitted is referred to as the primary encoding and subsequent transmissions as secondary encodings. Media specific FEC has the advantage of low-latency as a result of only a single-packet delay that is added. The media specific FEC scheme has been implemented and an extensive simulation environment has been adopted by Podolsky, Romer and McCanne (1998).

[2] A unit is an interval of audio data which is stored internally in an audio tool.

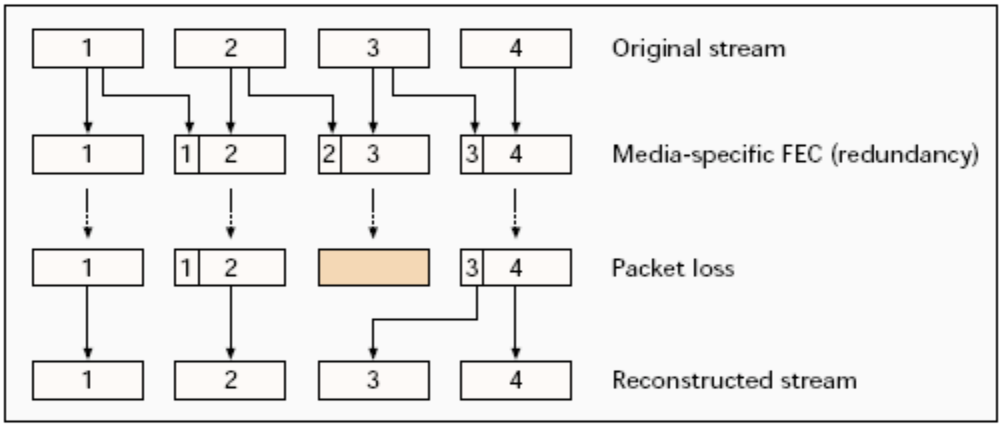

Figure 2.3 Repair process using media specific FEC

To protect against packet loss in the simplest form, each unit of audio is transmitted in multiple packets. When packet lost eventually occurs, another packet with the same unit is used to recover from the loss (Perkins, Hodson and Hardman 1998).

A Practical example of FEC

The gigabit networking research group at Bellcore developed a communication architecture called MECA, which stands for Multi-media End-to-end Communication Architecture. MECA provides multi-media applications with the service they require within the communication system (Biersack *et. al* 1992). MECA was build to support the following service classes:

- *Constrained latency*: these services are needed for human interaction in the form of voice and video, process control and remote sensing. Applications making use of constrained latency have constraints on latency variation such as jitter.
- *Transactions:* these services typically occur in distributed systems, such as distributed operating systems or databases where a client issues a request and a server responds to that request. A good example of such a service is the Remote Procedure Call (RPC). The transaction service requires low latency as the client is blocked while waiting for a response from the server.

- *Bulk data transfer:* this service carries a large amount of data between computers. In most cases data must be received reliably before the receiving application can proceed, but more often loose latency constraints are acceptable for higher efficiency.

MECA applies design principles to meet the requirements of low latency and efficient implementation. The system allows the reduction of end-to-end latency and jitter by using small packets as well as the processing of out-of-order packets to improve the utilization of protocol processors and to simplify processing. The error correction within MECA is controlled on an end-to-end basis as link error correction in high-speed networks (Bhargava *et. al* 1988). The transport protocol used, called TP++, makes use of a hybrid error control scheme that combines Automatic Repeat reQuest (ARQ) and FEC (Biersack *et al* 1992). ARQ is an error control scheme that is based on a combination of acknowledgements, time-outs and retransmissions. TP++ handles two types of network errors. These are bit errors due to noise on the transmission facilities and packet drops due to congestion. The ARQ component of the protocol uses selective retransmission and selective acknowledgement of the Transport Protocol Data Units (TPDUs). Thus, the transmitter decides when to transmit a TPDU or not. The receiver informs the transmitter of TPDUs that have been received correctly. The ARQ component generates error control messages and acknowledgments for all received TPDUs in every control message. The ARQ scheme cannot be used with applications that have tight latency constraints, as each retransmission adds at least one round trip time of delay. For the applications that require increased reliability, MECA makes use of FEC.

The FEC scheme, however, has the disadvantage of adding redundant information to the original data stream. The receiver uses the redundancy to reconstruct data in cases where there is a small amount of packet loss/corruption (Biersack *et al* 1992). Biersack *et al.* (1992) showed that FEC works well for autonomic video but struggles when it comes to homogeneous traffic scenarios. FEC extends traditional error concealment for bit error to packet loss (Shacham 1989; Shacham and McKenney 1990). As an example consider Figure 2.4:

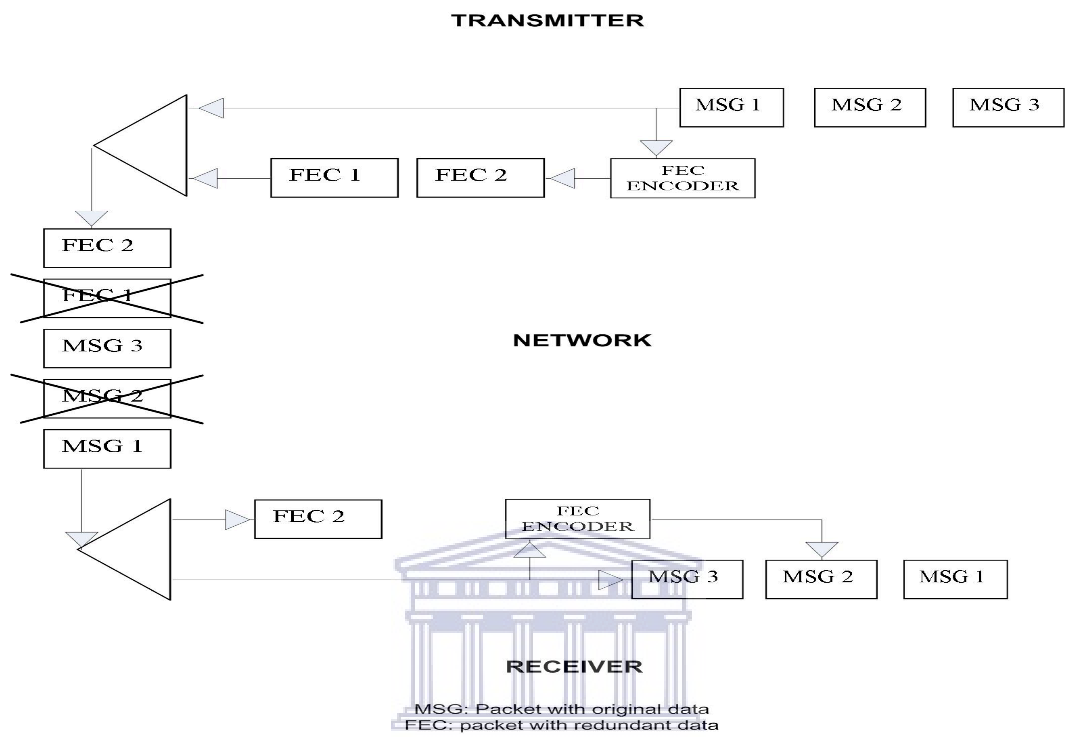

Figure 2.4 FEC operation

The FEC encoder produces two redundant packets (FEC 1 and FEC 2) for every three data packets sent (Biersack 1992). If, for example, a data packet (MSG 1) and a FEC packet (FEC 2) are dropped, the receiver can use MSG 2 to reconstruct the data packet.

The FEC transmitter sends a TPDU as k information packets and adds additional h redundant parity packets. The receiver can reconstruct the original k information packets if the network does not drop more than h of the $h + k$ packets sent. In Figure 6, k =3 and h=2 (k and h are MSG and FEC in the diagram). The FEC encoder thus produces two redundant packets (FEC 1 and FEC 2) for every three data packets sent. In this example, if one data packet (MSG 2) and one FEC packet (FEC 1) are dropped, the receiver can reconstruct the data packet (MSG 2). Biersack (1992) showed that for a small h/k ratio (e.g., h/k = 0.1), FEC is very effective and can reduce the TPDU loss rate by several orders of magnitude.

2.1.3.2 Interleaving

Interleaving is a very useful technique for reducing the effect of packet loss within a network (Ramsey 1970). However, this technique is only used when the unit size is smaller that the packet size, and the end-to-end delay is not important. An interleaver is a device that changes the order of a sequence of symbols. The corresponding device that restores the original order of the symbols is referred to as a deinterleaver (Perkins and Crowcroft 2000). The interleaver is placed within a transmission system with the purpose of randomizing the distributed errors after reception. As an example, a burst of loss on the channel is transformed into a sequence of isolated losses by the interleaving process. An interleaver that allows a function to change the order of a sequence of symbols repeatedly is referred to as a block interleaver (see Figure 2.5). A block interleaver is an $n \times m$ matrix that represents nm successive symbols stored in a buffer prior to transmission. For continuous interleaving, two matrices are required that allow symbols to be read into the one matrix and read out of the other.

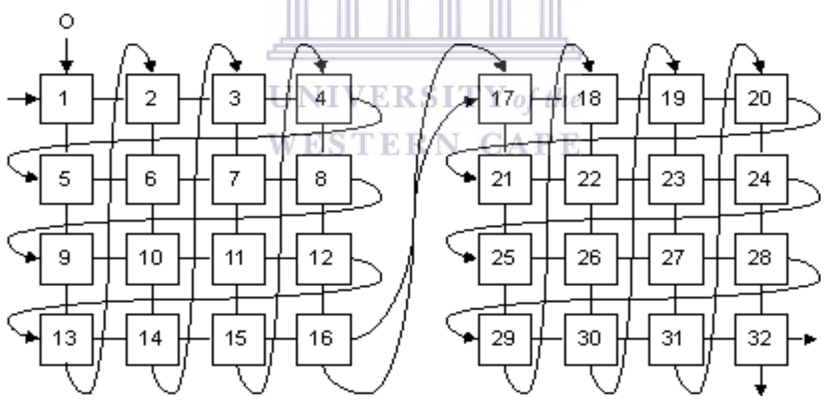

Figure 2.5 4X4 block interleaver

An $n \times m$ matrix represents nm successive symbols that are stored in a buffer prior to transmission. For continuous interleaving, two matrices are used with symbols read into the matrix row by row (Perkins and Crowcroft 2000).

The interleaver arranges the symbols in such a way that if m or fewer symbols are lost from a block, the original group of n symbols will contain at most one loss after de-interleaving. The interleaving process is illustrated in Figure 2.6. The burst of consecutive loss in an interleaved stream results in multiple small gaps within the reconstructed stream. This results in a spreading of loss that is important for two reasons. In the first case, applications that transmit voice packets transmit packets that tend to be similar in length to phonemes in human speech. When the loss is spread out in such a way that small parts of several phonemes are lost, listeners can mentally patch-over the loss (Miller and Licklider 1950). The second reason is that interleaving works best when the audio channel exhibits a burst of loss rather than isolated loss events. This is because error concealment techniques perform better when small gaps occur with the amount of change in the signal's characteristics being much smaller. The only drawback of interleaving is that it increases the amount of latency for a transmitted audio packet.

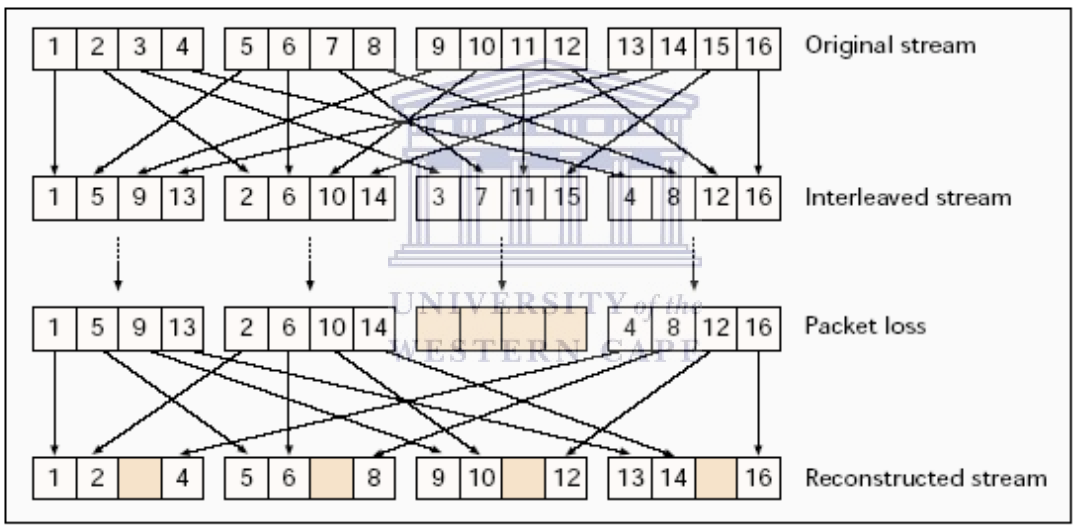

Figure 2.6 The interleaving process

Before interleaving commences, units are re-sequenced so that the original adjacent units are separated by a distance in the transmitted stream. They are thereafter returned to the original order at the receiver (Perkins, Hodson and Hardman 1998). The interleaving process is used to disperse the effect of packet loss.

2.1.3.3 Retransmission

Retransmission schemes are normally used when large end-to-end delays can be tolerated. These schemes are very impractical for wide area multimedia communications due to real-time constraints (e.g. RTP), small bit error rates of optical networks, and the ability of real-time media to tolerate loss. However, retransmission based schemes can be used for streaming audio in some instances. In particular, protocols that use retransmission and bound the number of retransmission requests for a given unit of data can justify the use of this scheme.

Due to the fact that ARQ schemes have a propagation delay small enough to permit retransmissions, they can be deployed in Local Area Networks (LANs), Metropolitan Area Networks (MANs) as well as Wide Area Networks (WANs) (Dempsey, Liebeherr, Weaver 1994). Dempsey, Liebeherr and Weaver (1994) demonstrated the applicability of retransmission-based error recovery schemes to packet voice streams for a wide range of end-to-end delays. In the most general multipoint applications, it was found that some participants will be local (vis-à-vis the transmitter(s)), others would be remote and it would not be justifiable to deprive the local users from the benefits of retransmission just because the remote users cannot take advantage of them. Retransmission schemes, such as ARQ, depend on two things: the end-to-end delay, and the amount of tolerance that human beings have for delay in interactive communications (Pejhan and Schwartz 1996). The end-to-end delay depends on three factors which include the time required to transmit a packet, the propagation delay, as well as the time required to process it. ARQ mechanisms are closed-loop mechanisms based on the retransmission of the packets that were not received.

2.1.4 Receiver-based packet loss recovery

When sender-based recovery schemes fail to recover all packet loss or when the sender of an audio stream is unable to participate in the recovery process, receiver-based (error concealment) recovery schemes are used (Pejhan and Schwartz 1996). Receiver-based recovery schemes produce a replacement for a lost packet that is similar to the original packet. The scheme works best when the loss rate is relatively small and the size of an audio packet is small (4-40ms). The scheme breaks down when the loss length approaches the length of a phoneme (5 -100ms) (Perkins, Hodson and Hardman 1998). This is because whole phonemes may be missed by the listener. Error concealment is in no way a complete substitute for sender-based recovery schemes but instead works in tandem with sender-based recovery schemes (see Figure 2.7).

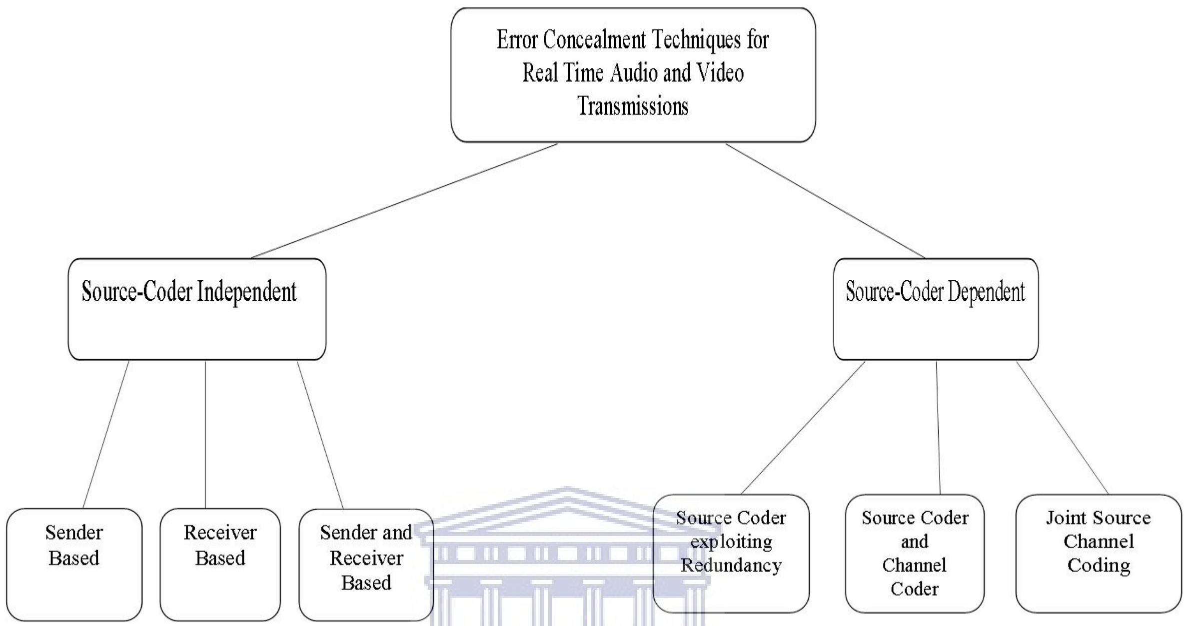

Figure 2.7 Classifications for error-concealment strategies for real-time audio.

Source coder-independent and dependent techniques differ in two extremes. Source coder-independent techniques assume no knowledge of the underlying coding algorithms. Sour coder-dependent techniques perform error concealment by exploiting features in individual coders (Wah, Su and Lin 2000).

Figure 2.7 classifies error concealment strategies in terms of source-coder independent and source-coder dependent techniques. A source coder-independent technique makes the assumption of being unknowledgeable of the underlying coding algorithms. In contrast, source coder-dependent techniques perform error concealment by exploiting features in individual coders. The source coder-independent schemes are further divided into three categories of schemes, according to the location of error concealment being carried out. They are sender-based, receiver-based and sender and receiver-based, respectively. Source coder-dependent schemes are also further categorized depending on the role of the source and channel coders in the concealment process. They are *source coder exploiting redundancy* in which error control is performed only in the source coder, *source coder and channel coder* in which a channel coder adds controlled redundancy to combat transmission error and *joint source-channel coding* in which the source and channel coder are designed cooperatively for the purpose of error control (Wah, Su and Lin 2000).

Loss-concealment actions, performed by receiver-based schemes, only occur at the receiver. The lost packets are recreated by 1) padding silence or white noise (Suzuki and Taka 1989), 2) repeating the last received packet (Tucker and Flood 1985), 3) using a form of pattern matching mechanism that uses small segments of samples immediately before or after lost packets has been received (Tang 1991; Wasem *et. al* 1988) or 4) replicating pitch-periods that are calculated/estimated using the speech segments immediately received before lost packets (Wasem *et al.* 1988). The above mentioned strategies work well when packet loss is infrequent and when the size of the packet is small (Hardman *et. al* 1995). Because the Internet has a high probability of dropping packets, these schemes are not very promising in a World Wide Web (WWW) environment. Sender receiver-based schemes are more promising as the senders convey knowledge about lost packets to receivers (Wah, Su and Lin 2000). Schemes that demand packets demarcated from the underlying network are not useful as the Internet does not provide such a service. Good examples are priority-based schemes which assign different priorities to different voice packets as well as an underlying network that drops packets according to their priorities when congestion occurs. These priority-based schemes assign priority according to 1) signal energy difference to previous packets, as well as voice onset or transition indicators (Yong 1992) and 2) whether a packet can be well-predicted in advance from previous packets (Lara-Barron and Lockhart 1992; Lockhart and Lara-Barron 1991; Yong 1992). Loss-concealment schemes based on retransmission control the playback time for the first packet in each talk-spurt thus managing to perform timely retransmission of lost packets (Dempsey, Liebeherr and Weaver 1993; Dempsey and Zhang 1996). These schemes however, are designed for local networks and will thus not work for the Internet. There are error control schemes that add redundancy to protect parts of a packet but not the whole packet. They take the form of waveform substitution allowing redundant information sent to include voiced/unvoiced indicators and pitch information (Sanneck 1998; Valenzuela and Animalu 1989). They also allow redundant information to be sent as 1) background noise or fricatives indicators (DaSilva, Petr and Frost 1989) or 2) short-time energy and zero-crossing (Erdol, Castelluccia and Zilouchian 1993). In these schemes lost packets are replaced by receivers finding a "best match" on the redundant speech segments received.

Redundancies exploited by source-coding schemes enable loss concealment of source coders by exploiting redundancy in coded streams. Suzuki and Taka (1989) and Saito, *et. al* (1989) assign high priority to Most Significant Bits (MSB) and low priority to Least Significant Bits (LSB) in a waveform coder and assign high priority to pitch parameters and low priority to excitation information for linear prediction based coders (Yong 1992). Figure 2.8 shows a taxonomy of error concealment techniques.

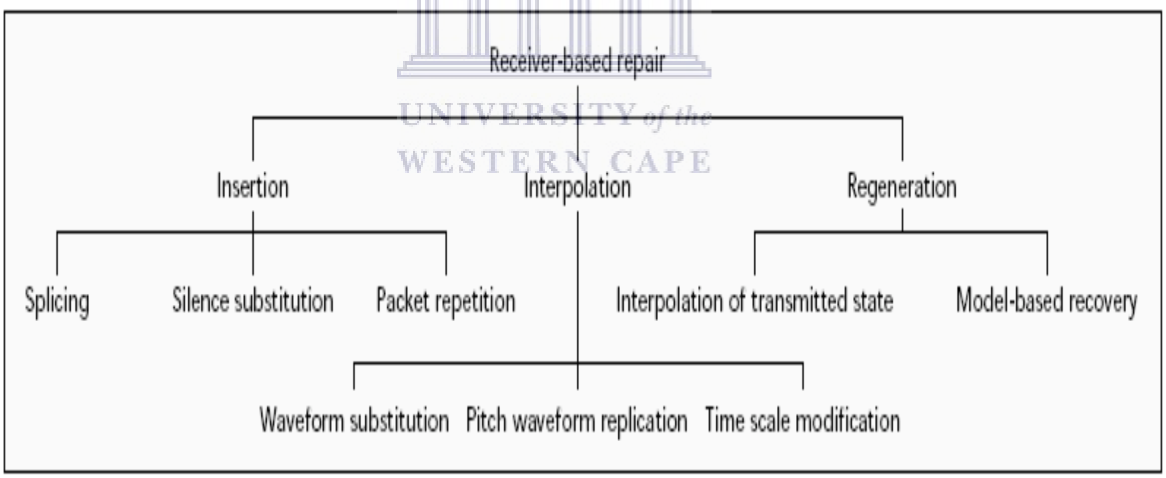

Figure 2.8 Taxonomy of receiver-based error concealment techniques

Insertion based schemes repair audio loss by inserting a silent or noise packet in the place of the lost packet. Interpolation based schemes uses pattern matching and interpolation to derive a replacement packet. Regeneration based schemes uses the decoder's state, from packets surrounding the loss, and uses this state to generate a replacement packet (Perkins, Hodson and Hardman 1998).

The sender-based packet loss recovery scheme repairs most losses, leaving a number of small isolated gaps to be repaired. When an effective loss rate (measured by the amount of isolated gaps) is reached, error concealment forms an effective means of patching over the remaining loss.

2.1.4.1 Insertion-Based Repair

The insertion-based repair scheme involves the derivation of a replacement for a lost packet by simply inserting a simple fill-in packet. The scheme uses one of three techniques to conceal packet loss: splicing, silence substitution and repetition.

Splicing

The lost units are concealed by splicing together audio on both sides of the lost packet thus closing the gap. This technique, however, disrupts the timing of the stream as shown by Grubler and Strawczynski (1985).

Silence Substitution

Silence substitution is concerned with maintaining the timing relationship between the surrounding packets. It maintains this relationship by filling the gap, left by the lost packet, with silence (Hardman *et. al* 1995).

Repetition

Repetition makes a copy of the unit that arrives immediately before the loss packet and uses this unit to replace the lost units.

2.1.4.2 Interpolation-Based Repair

Interpolation-based repair schemes account for the changing characteristics of a signal. The scheme uses one of three techniques to conceal packet loss: waveform substitution, pitch waveform replication and time scale modification.

Waveform Substitution

Waveform substitution uses audio before and/or after the lost unit to locate a suitable pitch pattern on either side of the loss (Goodman *et. al* 1986).

Pitch Waveform Replication

This technique is similar to waveform substitution but instead of using audio before and after a lost unit, a pitch detection algorithm is used instead (Wasem *et. al* 1988).

Time Scale Modification

This technique allows the audio on either side of the loss to be stretched across the loss, thus in effect, covering the gap left by the lost unit (Sanneck *et. al* 1996).

2.1.4.3 Regeneration-Based Repair

This repair scheme is a bit more complex than the error concealment techniques. It uses knowledge obtained from the audio compression algorithm to derive coded parameters in such a way that audio in a lost packet can be synthesized. The scheme uses one of two techniques to conceal packet loss: interpolation of transmitted state and model-based recovery.

Interpolation of Transmitted State

The advantage of this particular repair technique is that it inserts state information between audio packets rather than recoding audio either side of a lost packet. The disadvantage is that it requires a high amount of processing.

Model-based Recovery

This recovery technique uses speech on both sides of the lost packet and fits it against a model that is used to generate speech to replace the loss packet. Chen and Chen (1997) used the model-based recovery technique to repair encoded speech by combing the result of analysis done on the last received units with the amount of lost units.

2.2 Reliable Asynchronous communication

Asynchronous communication tends to work in a greater variety of circumstances than synchronous messaging. Synchronous communication suffers from 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 telephone 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. A persistent local queue has a primary receiver defined for it as well as failover receivers. If delivery to the primary receiver fails, messages are routed to a secondary receiver until the primary receiver reconnects to the system. These messages can be routed via brokers connected to a brokering network that provides reliability for the exchange of messages.

2.2.1 Instant Messaging

In recent years, IM has eclipsed more traditional asynchronous text chat technologies such as Multi User Dungeons (MUD) and Internet Relay Chat (IRC) (Grinter and Palen 2002). This is because MUD and IRC tend to be used for communication among people who do not know each other in the Internet. In many instances, technologies such as MUD and IRC are used to allow people to congregate around topics such as common interests and discussion groups. On the other hand, IM distinguishes itself by enabling communication to occur among known individuals. IM is based on a dyadic call model where users tend to communicate with one person even though several dyadic conversations with several individuals may occur simultaneously. IM also supports group chat, either with known participants or anonymously as with MUD. With IM, awareness information is provided about the presence of other users maintained in a buddy list. Instant audio and video are two distinct technologies that make IM popular among Internet users (Stone and Merrion 2004). Stone and Merrion (2004) look at security issues currently facing IM by looking at the functionality of IM and the possible ramifications these functions provide to the security of a cooperate environment. This thesis, however, looks at IM in terms of reliability provided for a Deaf Telephony application. Since the Internet provides a best effort delivery service with a high probability of dropping packets, IM and the functionality it provides could become problematic. Thus, a perennial reliable message delivery framework is needed to encapsulate text messages sent via the use of IM.

2.2.2 Error Control in Store and Forward Communication

Software running reliably in small-scale mockups tend to be unreliable as the number of users, the size of the network, and the transaction processing rates all increase. 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 or poor connectivity and throughput instability (Bolot and Garcia 1996). 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.

2.2.2.1 Retransmission

In a perfect world with no end-to-end delay and no unreliable networks, an end-to-end transport protocol would only need to mark the start of a transmission, and then place all data through the communication channel. While nice in theory in reality these kinds of networks don’t exist as unreliability manifests itself as packet loss, duplicates, and out-of-order deliveries. This calls for a reliable transport protocol that detects and recovers from all transmission errors. The protocol should do this by numbering the data units, and keeping transmission state information at the two communicating ends. The state information also permits recovery from errors whenever they are detected. Thus, the transmission state regulates the flow of traffic. In order for this to be successful both communicating ends need to be constantly synchronized by the arrival of new data and the control information such as acknowledgements. This allows the receiving end to easily check and correct duplicate or out of order packets. If data or control packets should be lost, the easiest recovery technique is for the transmitting end to wait for the retransmission timeout and then retry until a successful retransmission resynchronizes the end states. However, as shown by Clark, Lambert and Zhang (1987) the detection of lost packets by timers takes a relatively long time which sometimes causes false alarms leading to superfluous retransmission.

2.2.2.2 Selective Acknowledgement

With no communication delay between the transmitter and the receiver, the transmission state at both communication ends would be perfectly synchronized, thus having a consistent view of the status of every data packet all of the time. This would make a transmitter's job easier by allowing it to retransmit any packet in error immediately after it is detected by the receiver. Network Round Trip Delay (RTD) is usually non-negligible and varies randomly (Clark, Lambert and Zhang 1987). The RTD causes the control state at the two communicating ends to be out of synchronization leaving them with different views of the status of the data under transmission from time to time. This causes both parties to be unaware of each other's state changes. When, for example, a transmitter sends a packet P, it will only know if P was successfully received until at least an RTD period. Meanwhile it must continue sending subsequent packets to achieve high throughput.

The NETBLT protocol was designed as a high throughput protocol that is robust in the face of a network's long delay and high loss of the network (Clark, Lambert and

Zhang 1987). NETBLT uses selective acknowledgement to convey as much information as possible to the sending NETBLT client. Unneeded retransmissions are avoided by providing the status of each packet in the transmission. Because the overhead incurred for acknowledging each packet is very high, NETBLT splits its out-of-synchronization region into buffers that act as the synchronization points and recovery blocks. Both sending and receiving NETBLTs have their state synchronized upon the successful transmission of a buffer or upon determination by the receiver that information is missing from a particular buffer. By placing the data retransmission timer at the receiving end, NETBLT in turns makes error recovery more efficient. The receiver can thus at any point know which packets have arrived and which have not. With the occurrence of a timeout, the receiver can eliminate unnecessary retransmissions because it knows exactly which packets need retransmission. Estimating the timer value by the receiver becomes easy since it is based on the transmission rate and the number of packets expected in a particular buffer.

2.2.2.3 Positive and Negative Acknowledgements

Erramilli and Singh (1987) made use of a protocol for the reliable and efficient distribution of multicast data of broadband broadcast networks. The protocol provides reliable, connection-oriented packet communication among multiple users in a broadband broadcast network. The protocol is based on negative acknowledgments (NAK) with some enhancements so that it may match most of the functionality of a positive acknowledgement (ACK) based protocol. Robustness, within point-to-point communications, is typically achieved by error detection through an ACK from the receiver to the source. An alternative to the ACK scheme is the use of a NAK scheme which sequentially numbers data packets at the source. The receiver does not acknowledge the receipt of packets under normal operation but instead detects errors by gaps in the sequence number in the packet stream and sending a NAK about the non-receipt of the packet at the source. Erramilli and Singh (1987) enhanced the architecture of the NAK protocol to match most of the functionality of the ACK protocol allowing a throughput that stays relatively insensitive to the size of the multicast group. The enhancements included timers and status packets to demarcate the time taken to recover form packet loss. The enhanced ACK scheme implements a flow control mechanism so that the sender can choose to send a new packet only after it has received acknowledgments from every receiver in the multicast group. If it does not receive acknowledgements from any receiver within a specified period of time it retransmits the data packet. Retransmission occurs up to a certain maximum number of times and if the receivers still do not respond, then the assumption is made that the connection is no longer viable.

2.2.2.4 Other approaches

The Spinglass project employs gossip protocols at very high speeds. These protocols have an unusual style of providing probabilistic reliability guarantees (Birman, van Renesse and Vogels 2001). The Spinglass project works towards overcoming scalability barriers and presents a custom methodology to yield 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. Birman, van Renesse and Vogels (2001) disprove 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 shows 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.

2.2.3 Message Oriented Middleware

MOM technology allows the management of the distribution and computations of services to be transparent to developers. In many cases, an asynchronous approach appears to provide scalability, extensibility and openness of large scale communications. The use of component-based approaches deals with the issues of flexibility and configurability. Properties such as message persistency and causal ordering of messages are pre-defined in the middleware layer, and thus run on all nodes of a distributed system.

2.2.3.1 MOM architecture

MOM is increasingly being seen as a significant way to improve enterprise productivity and to facilitate the open service market. MOM is based on an asynchronous model which allows for application integration and information dissemination to many users (Tran and Greenfield 2002). 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 (Figure 2.9).

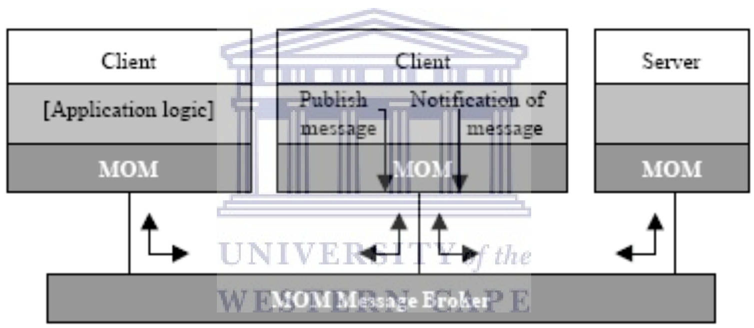

Figure 2.9 Architecture of MOM

The interaction between clients and MOM's brokering network. Each client is connected to a broker node. The broker node is connected to a broker network and is responsible for delivering messages to the broker network (Jung, Paek and Kim 1999).

The MOM infrastructure handles network communication. If a communication network becomes temporally unavailable, MOM will store and forward the destined messages once the network connection is re-established. Another interesting aspect related to the architecture of MOM is that it will only send a message to an actively executing receiving client (Quema 2002). 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 RPCs 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 abstract reliable entity. Subscriptions are persistent, and messages may not be lost, permuted, or duplicated, nor must spurious messages 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.

In component-based MOM applications, components are created and executed within a component server. A component server is a virtual machine which ensures component creation, execution, and communication. The component server consists of a local bus and a component factory. Multiple component servers are statically interconnected. Before an application is started, the component servers supporting the execution of an application are all known. The purpose of the local bus is to convey all the messages and make components react to these messages. It also handles local communication (messages to and from the components in the local component server) internally. When a message is sent to a remote component, the local bus relays the message to the local bus of the remote component server. The local bus is composed of two main components:

1) a channel component that relays all messages reliably. This is done by using a system of message queues. When a message fails to be sent to a remote node, the channel periodically retries to send it; and

2) an engine component: this component makes all other components react to the messages. The engine is the main execution entity of the component server. It performs a set of instructions in a loop, getting the next message from the channel, loading the proper component and making it react.

2.2.3.2 Reliable Message Delivery

MOM technologies deliver messages reliably. Products normally offer three levels of QoS. They are non-persistent, persistent and transactional. In non-persistent delivery, the MOM will do its best to deliver the message. Undelivered messages are only kept in memory and can be lost if a system or network fails before a message is delivered. With persistent delivery, the MOM guarantees to deliver messages despite system and network failures by logging them to disk as well as keeping them in memory. This means they can be recovered and subsequently delivered after a system failure. With transactional delivery, the MOM tightly integrates persistent messaging operations with application code, not allowing transactional messages to be sent or received until the application commits their enclosing transaction. Transactional messaging also allows message sending and receiving to be coordinated with other transactional operations, such as database updates. Combining these levels of QoS results in 3 different alternatives for performance measurement. They are non-persistent non-transactional (NPNT), persistent non-transactional (PNT), and persistent transactional (Tran and Greenfield 2002). There are two variants of PT. They are Persistent Local Transaction (PLT) and Persistent Global Transaction (PGT). In PLT messaging, there is no integration with external transactional operations such as database updates and the MOM product itself manages the transaction with single-phase commits. PGT messaging involves external transactional operations such as database updates.

2.2.3.3 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 2.10. 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). Message distribution is handled by an underlying content based routing network (Carzaniga and Wolf 2000). 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. The following subsections provide a brief survey of some publish-subscribe systems.

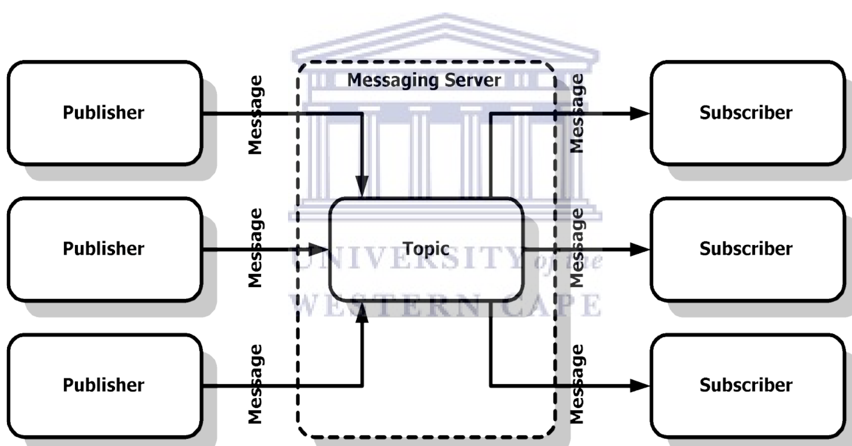

Figure 2.10 Publish-Subscribe interaction

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.

2.2.4 MOM environments

In many cases middleware solutions provide two important communication paradigms: asynchronous messaging and RPC. RPC is a synchronous method of requesting remote services execution allowing distribution to be transparent to the programmer. However, MOM solutions tend to be more robust to failures than that of RPCs. The difference between MOM and RPC lies in the fact that MOM-based application tends to be more cumbersome to program than RPC (Menascé 2005). Then next three subsections look at systems based on MOM.

2.2.4.1 NaradaBrokering

NaradaBrokering is a distributed brokering system, implemented on a network of cooperating broker nodes (Pallickara and Fox 2003; Pallickara and Fox 2004). The system makes use of MOM to ensure the reliability of messages sent via brokering nodes. Message delivery relies upon a number of supporting components that handle connection services, message routing and delivery, persistence, security, and logging (Figure 2.11).

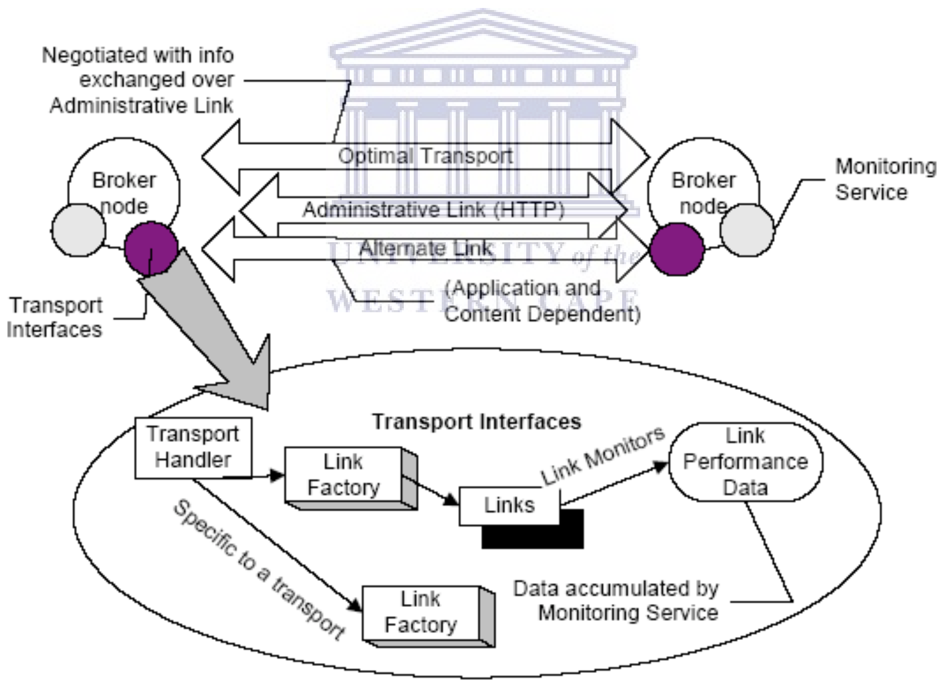

Figure 2.11 NaradaBrokering transport framework

The transport framework abstracts operations needed to enable efficient communication among broker nodes. These operation allows the easy addition of transport protocols within the distributed broker network (Pallickara and Fox 2003).

In order for NaradaBrokering to perform message delivery, a broker must establish communication channels with clients, perform authentication and authorization, route messages appropriately, guarantee reliable delivery, and provide data for monitoring system performance. Communication within NaradaBrokering is asynchronous and the system can be used to support different interactions by encapsulating them in specialized events. Once clients reconnect after a prolonged disconnect, a connection to a local broker is established. Within Naradabrokering the addition of brokers in the broker network is managed by the use of a broker organizational protocol. Naradabrokering's transport framework abstract the operation needed to provide efficient communication among broker nodes. The abstracted operations allows: 1) an easy addition of transport protocols within the distributed broker network, 2) the deployment of specialized links to deal with specific data types, 3) the negotiation of the best available communication protocol between two or more broker nodes, 4) network adaptability for communication by responding to a changing network environment and 5) accumulating performance data that is measured by underlying protocol implementations. To guarantee the delivery of messages within the broker network, Naradabrokering make use of a MOM, a combination of acknowledgements and a storage medium to persistently log messages.

2.2.4.2 JORAM

The Java Message Service (JMS) specification (*http://java.sun.com/products/jms*) abstracts interactions in publish-subscribe environments. These interactions can be implemented on top of a variety of transport protocols. The specification defines a communication protocol between message producers and consumers. Communication occurs through an asynchronous communication system. Java Open Reliable Asynchronous Messaging (JORAM) is a middleware component that is entirely based on the JMS specification (Figure 2.12) (Bellisard *et. al* 2000). The software package is provided by the ObjectWeb code base and is entirely open source.

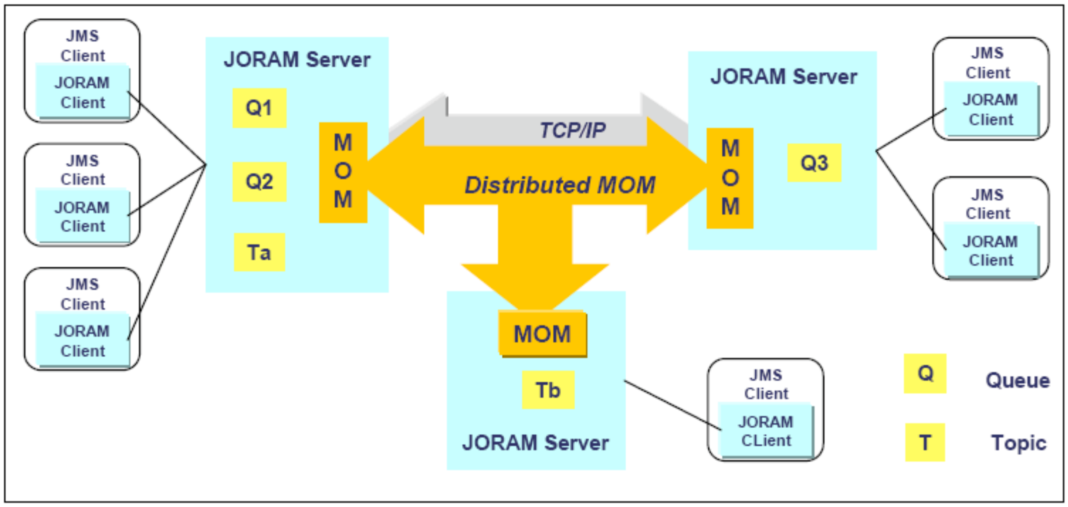

Figure 2.12 JORAM platform architecture

JORAM is based on a "snowflake" architecture that allows a JORAM platform to be composed of a set of JORAM servers interconnected by a message bus, offering various communication protocols (Balter 2004).

JORAM is designed in two parts: a JORAM server that manages the JMS abstractions and JORAM clients that are bounded to a JMS application. The JORAM platform is composed of JORAM servers interconnected by a message bus which offers various communication protocols (Balter 2004). JORAM uses a proxy object to implement a store and forward function for reliability and persistence in terms of message delivery (Figure 2.13).

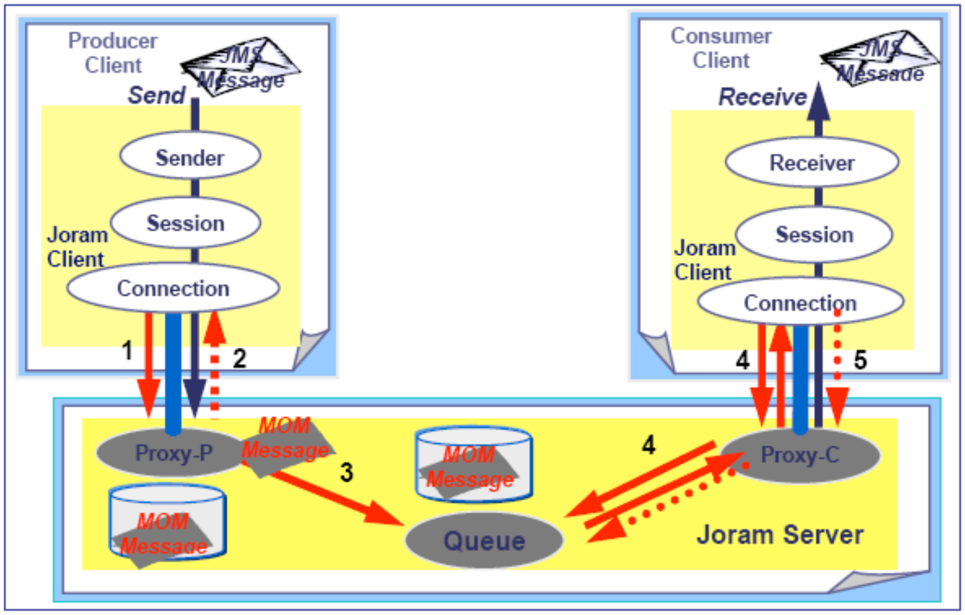

Figure 2.13 JORAM message flow

All messages sent in JORAM are encapsulated in a MOM message. This MOM message is then persistently logged at a message queue. The process of sending and receiving messages in JORAM are all monitored by the use of ACK messages (Balter 2004).

Each message sent is encapsulated into a MOM message that is saved on a local persistent storage medium which implements a message queue. When messages are received by a subscriber, a control message is sent to a proxy object which in turn forwards it to a queue object. The message is then retrieved from a queue object and sent to a proxy object. Within JORAM, each message sent and received by publisher and subscriber is monitored by the use of ACK messages. Before messages are stored in a message queue the MOM message is stored separately. This allows messages to be recovered once failure occur which in the process guarantees message delivery.

2.2.4.3 GRYPHON

Bhola, Zhao and Auerbach (2003) developed a system that supports durable subscriptions within a publish-subscribe architecture called GRYPHON. GRYPHON (*http://www.research.ibm.com/gryphon/*) is a scalable, wide-area content-based publish-subscribe system, employing a redundant overlay network of brokers. The system is based on a knowledge graph that models propagation of knowledge from publishers to subscribers and demands of knowledge from subscribers to publishers (Figure 2.14). The protocol used in the system is tolerant to broker crashes as well as dropped and re-ordered messages (Bhola *et. al* 2002). Figure 2.14 shows that nodes contain states (called streams). The streams can either contain knowledge about which ticks of time contain data messages or curiosity about how urgently streams require the knowledge of each tick.

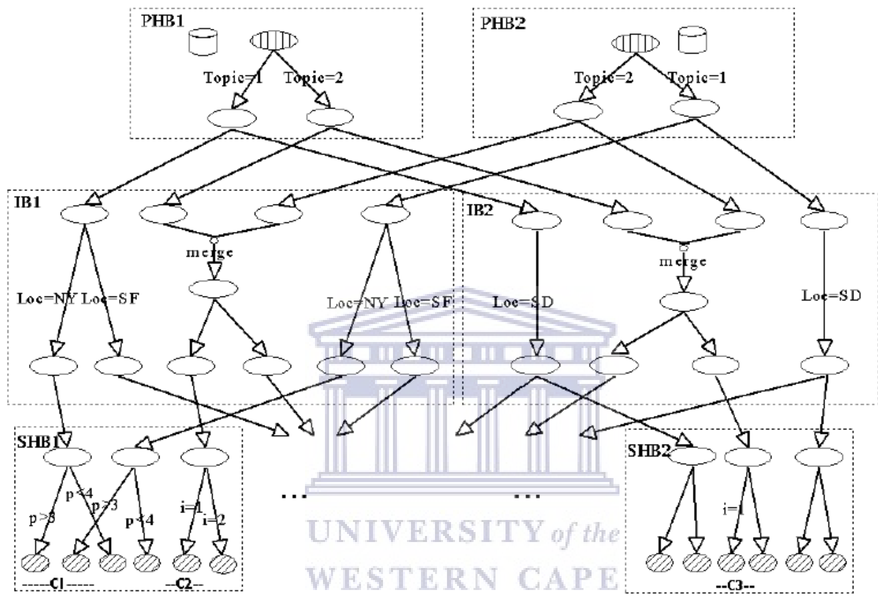

Figure 2.14 Example of a knowledge graph

A knowledge graph models propagation of knowledge from publishers to subscribers through filter and merge operations and propagation of demands, from publisher end points (pubends), for knowledge in the reverse direction. In the diagram the pubends are identified with vertical hash-marks where the subscriber end points (subends) are shown with diagonal hash marks (Bhola et. al 2002).

As clients publish messages to the brokers hosting the publishing endpoint (pubend), knowledge is created at the pubends and then later consumed at the subscribing endpoints (subends) and in the process delivered to subscribing clients. The Subscriber Hosting Broker (SHB) maintains a persistent event log for each durable subscriber in which each event that matches the subscriber is placed. If a subscriber is disconnected from the system involuntarily, it reconnects to the SHB that maintains its persistent event log (Figure 2.15). The SHB is responsible for providing knowledge and curiosity for each connected client that is in a “catchup” mode.

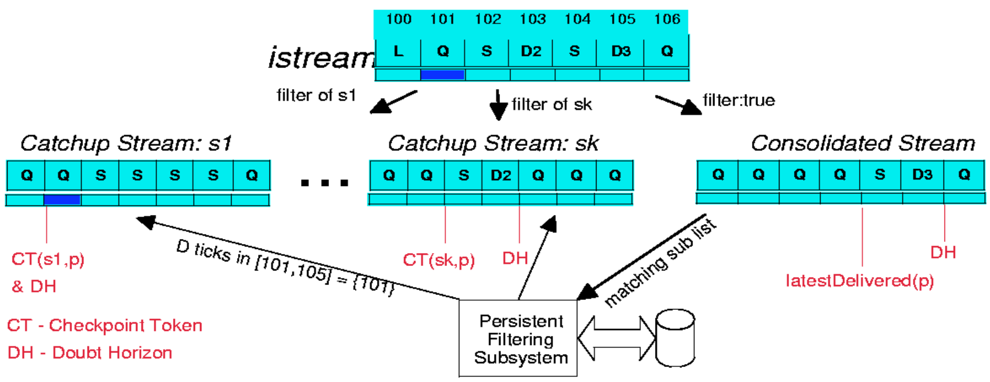

Figure 2.15 An example of a knowledge graph inside a GRYPHON SHB.

The durable subscriber can either be in a catchup or non-catchup mode for a given pubend. The SHB maintains a knowledge and curiosity stream for each connected subscriber that is in catchup mode. The consolidated stream is meant for non-catchup subscribers (Bhola et. al 2003).

Within the system each Publisher Hosting Broker (PHB) maintains one or more pubends. Each pubend maintains a persistent and ordered event stream that uses timestamps for indexing which is assigned to the event when it was added to the stream. On first connection to the system, the durable subscriber is provided with a starting point (timestamp) for each pubend in the system. All subscribers are delivered monotonically increasing timestamp information for each pubend using messages containing the timestamp information. A Persistent Filtering Subsystem (PFS) stores information about which events matches a durable subscriber (Figure 2.16). Each time information is written to PFS a write call containing monotonic timestamps, representing a time tick and a list of subscribers associated to the time tick, is made. A PFS may house four kinds of time ticks: Q (unknown), S (silence), D (data) and L (lost). The D tick represents events published by an application. The S tick means that there was either no event at that particular time stamp or that the event was filtered upstream within the knowledge graph. An L tick shows that a pubend is not sure if it is a S or a D tick.

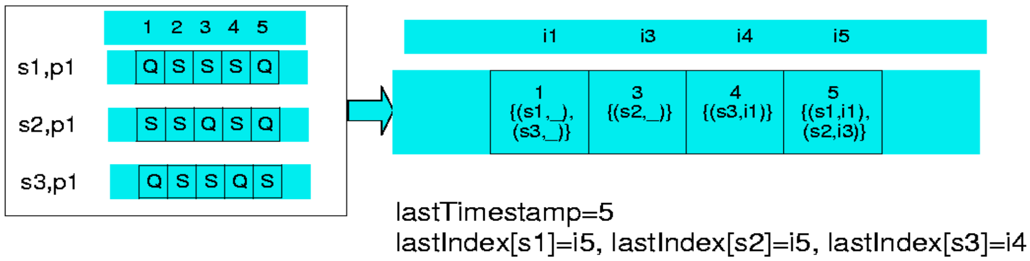

Figure 2.16 Persistent filtering subsystem

The PFS stores information about which events match a durable subscriber. In this example three subscribers and their knowledge stream for time one to five is shown (Bhola et. al 2003).

2.5 Summary

This chapter provided a brief survey of related work in the fields of reliable synchronous and asynchronous communication. Error control in real time communication was organized into three techniques for providing a reliable form of synchronous communication: ARQ, FEC and EC. Applications that make use of these techniques can ensure reliable communication in a real time medium. Reliable asynchronous communication systems can make use of protocols incorporating ACK and NAK to guarantee data delivery. Publish/subscribe systems are also identified as a solution for guaranteeing asynchronous communication. MOM is offered as a solution, and a survey of systems were presented in this regard. We see that synchronous communication has a set of reliability solutions and asynchronous communication has a different set of approaches. In the next chapter, the information from the literature review is used to determine a solution for guaranteed delivery of messages in a semi-synchronous environment. The concept of reliability is abstracted and applied to a specific Deaf Telephony application.

Chapter 3 RESEARCH METHODOLOGY AND APPROACH

In general, reliability is paramount in communication. The previous chapter covered related work in the field of reliable communication. For the rest of this thesis, reliability is discussed within the context of an IP-based communication infrastructure to support Deaf Telephony. This infrastructure facilitates both synchronous and asynchronous communication. The previous chapter's discussion included implemented systems and practical examples for both synchronous and asynchronous forms of data exchange. Key technology concepts were distilled which help inform the formulation of a methodology for the research question. For synchronous communication three potential solutions were introduced: ARQ, FEC and EC. MOM was identified as a feasible solution for asynchronous communication.

This chapter surmounts the problem of reliable communication but more specifically within the context of a Deaf telephony infrastructure. The chapter thus focuses on the ascertainment of a solution to the research question. The method of finding a solution as well as the actual solution is introduced and discussed. Once the solution is introduced and discussed, a methodology is introduced to evaluate the proposed solution. In this chapter three methods for the software implementation are identified and discussed. These include an evolutionary development process called exploratory prototyping, Component-Based Software Engineering (CBSE) as well as ethnographic observation.

3.1 Approach to the research question

This section describes the approach taken to surmount the problem of reliable communication within a semi-synchronous environment. Chapter 1 introduces this problem within the context of a Deaf Telephony application. This section thus re-introduces the research problem, shows how the problem statement was formulated, what solution is proposed and what methods were followed to help make the design decisions.

3.1.1 Research question

The main research question is: "How does one guarantee exactly-once delivery of mixed synchronous and asynchronous messages within a Deaf Telephony application?" Since the application domain is Deaf Telephony, the research question is asked within the context of a Deaf person's perspective. Within this thesis, reliable IP-based communication is perceived to be both a reliable synchronous as well as reliable asynchronous form of communication. This is due to the path messages traverse between communicating entities. The application domain includes synchronous communication, in the form of a telephone conversation over the PSTN, as well as asynchronous communication in the form of IM. Thus, messages traverse through both synchronous and asynchronous paths before delivery commences. The thesis thus looks at ways of delivering these messages with respect to the type of the transport medium.

3.1.2 Formulation of the research question

The need to guarantee the delivery of messages within the context of a Deaf telephony application came from initial Deaf Telephony trials conducted between a Deaf and a hearing person (Tucker, Glaser and Lewis 2003; Sun and Tucker 2004). It was deduced that within an emergency scenario, the Deaf person needs assurance that messages have reached their intended destination. Most Deaf people in the Western Cape use SMS as their primary communication service. Even though SMS does not facilitate synchronous communication, it is still popular among the Deaf because of its wireless nature and somewhat inexpensive form of data exchange. Since SMS is also an unreliable form of asynchronous communication, it cannot substitute for a medium that provides synchronous, acknowledged communication. Deaf telephony distills the fundamentals of synchronous and asynchronous communication to provide an application that is similar to SMS, but with aspects of synchronous communication, thus having obvious benefits for the Deaf community.

3.1.3 Proposed technical solution

Since the research question is divided into two parts the proposed solution is twofold: MOM and FEC for the reliable exchange of asynchronous and synchronous messages, respectively. MOM is based on an asynchronous model that acts as an intermediary between end points (see Figure 2.9). As mentioned in the previous chapter, MOM handles all network communication and thus stores messages if a network connection is unavailable. Once a connection is established it then forwards the stored messages to its destination.

FEC is proposed to mitigate the impact of packet loss for real-time audio over an IP-based network. In the literature, two classes of FEC repair data are identified that may be added to an audio stream: media independent FEC and media specific FEC. In this thesis, media independent FEC is proposed as a solution for reliable synchronous communication. The media independent FEC scheme requires little computation for the derivation of the error control packets. However, the media independent FEC scheme requires an increase in the amount of bandwidth for the amount of redundant information it sends for each packet in subsequent packets. At the moment, DCCT, which forms part of our testing environment, has an Asymmetric Digital Subscriber Line (ADSL) which provides a sufficient amount of bandwidth. This allows us to adopt the media independent FEC scheme with little concern. Further development over a broadband network will allow the gateway and operator to be placed just about anywhere on the network. Such a network should allow a media independent-based FEC scheme to work with little problem.

3.1.4 Ethnographic observation

Ethnography bridges the gap between the generation of knowledge conducted by researchers and that of practitioners. Ethnography is born out of the social science called anthropology and is widely used in the study of Information Systems (IS) in organizations which include the study of the development of IS (Hughes, Randall and Shapiro 1992; Orlikowski and Robey 1991), the study of the use of Information

Technology (IT) forms, as well as certain aspects of IT management (Davies and Nielsen 1992). Ethnography provides software developers with rich insights into the human, social and organizational aspects of software development (Harvey and Myers 1995). Ethnography deals with real world scenarios where both practitioners and researchers can use frameworks developed as a result of actual practices conducted in real world situations. This allows researchers to deal with real situations and not be limited to artificial situations for the purpose of experimental investigations. As a result of ethnography, researchers are able to study organizations as social and cultural entities rather than just a test bed for experimental purposes.

In this thesis, DCCT is the organization under study. Ethnography forms the basis for observation conducted within the community that DCCT serves. Observation started from October 2004 with basic computer literacy introduced to a number of Deaf participants (Glaser, Young and Porteous 2005). Literacy skills such as typing tutor, e-mail, Internet and IM were introduced gradually as more Deaf participants were involved as time passed. In general it was observed that most Deaf participants were not making use of the "touch" typing technique resulting in slow typing. This came to be a very important factor within the overall observation. A Deaf Telephony application that facilitates an emergency service would have to provide a reliable form of communication as retyping the message, due to failure of delivery, would be time consuming. In an emergency scenario time is of the outmost importance and messages needs to be delivered the first time, and on time, if possible.

After a period of five months, a selected number of Deaf participants were trained to use SIMBA (Sun and Tucker 2004). These initial Deaf participants were chosen based on their performance in the literacy training program. From this training program another critical observation was made. The observation was based on the level of education of the observed Deaf participants. It was found that because Sign language is the Deaf participant's first language, some of them were struggling to communicate in English, as they made use of the IM client to communicate with a hearing participant. This in effect influenced the way the synthesized voice of the TTS engine pronounced words that were misspelled. Experiments conducted with Deaf and hearing participants showed that words that were misspelled, by Deaf participants, did not influence the hearing users perspective on what was said negatively (Zulu, Glaser and Le Roux 2005). SIMBA could not afford to have the audio quality be influenced by audio packet loss. Using ethnography as a tool to observe the human interaction with SIMBA, the relevance and importance of a medium that guarantees both synchronous (to prevent any degrading in audio quality) and asynchronous (emergency scenario) is seen.

3.2 Technical Methods

Prototyping within software development approaches are able to dynamically respond to changes in user requirements, reduce the amount of rework required as well as control the risk of incomplete requirements. With a prototype technique the development process is partitioned into smaller steps which is easier to handle, cost-effective (Gordon and Bieman 1994), improves the quality of communication between the development team, helps determine the feasibility of the technical solution and is a good risk management technique (Tate and Verner 1990). There are vast differences between prototypes and prototyping. Prototypes are models, simulations or partial implementations of systems that are used to test the feasibility of certain technical aspects of a system. Prototyping, on the other hand, is a process or phase within the software development life cycle (Budde *et. al* 1992). In this thesis, two approaches that compliment each other are fused together to create a feasible solution for the research question. These are exploratory prototyping and the fundamentals of CBSE.

3.2.1 Exploratory prototyping

The objectives of exploratory prototyping are to work with the user with the intention of getting a better understanding of the user requirements and in the process deliver a final system (Figure 3.1).

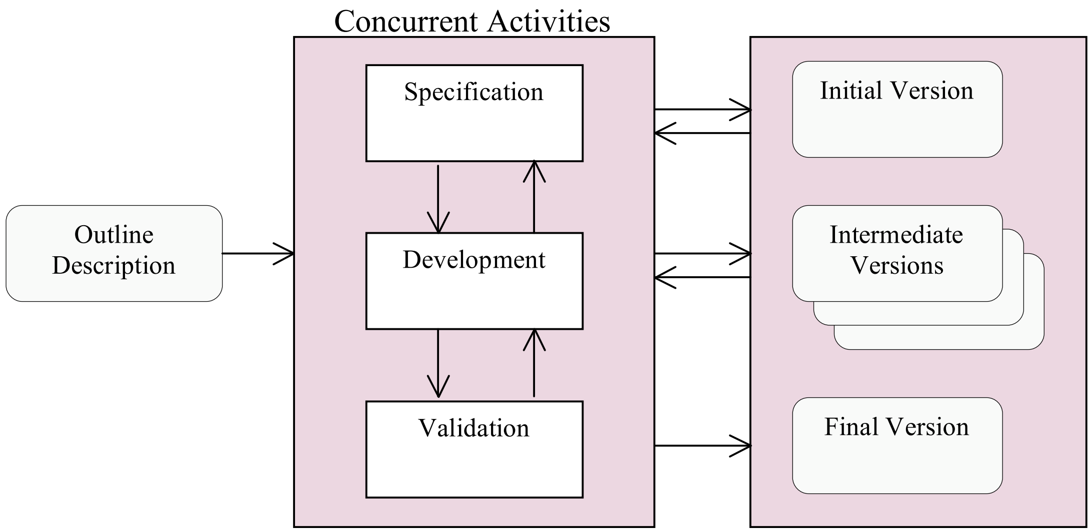

Figure 3.1 Exploratory prototyping

The prototyping model includes steps for requirements specifications, analysis, rapid development, refinement, validation and a final system release.

The exploratory phases are defined as follows:

1. *Requirement specification*: work with the user to clarify the software requirements. In this phase of the software development process, the requirements are analyzed and a requirements analysis document is produced. In our case the user requirements came from three different research cycles: 1) trials conducted with a TTY device (Glaser 2000), 2) initial Deaf Telephony trials conducted between a Deaf and hearing person using a Deaf Telephony application (Tucker, Glaser and Lewis 2003) and 3) a series of trials conducted with SIMBA (Sun and Tucker 2004).
2. *Initial version*: after the requirements are gathered, an initial prototype of the software is produced. There were initially two prototypes produced for this thesis. The first prototype was developed to test the reliability component of MOM (Julius and Tucker 2005). The first prototype thus includes a separate implementation of the reliable asynchronous component, before integration with SIMBA (Sun and Tucker 2004). The second prototype is based on the reliable synchronous component. This prototype is also implemented

separately from the integration with SIMBA. The second prototype includes the implementation of FEC and a test base to test the overall interaction of FEC.

3. *Development*: the development phase refines the initial prototype (intermediate versions) until an adequate version is delivered. This phase is used for both initial prototypes. During development the reliable asynchronous and synchronous components had a number of intermediate versions before an adequate version was selected.
4. *Validation*: the proposed adequate version goes through a validation phase that includes experimental users. The validation phase includes "laboratory" users where the feasibility of both the reliable synchronous and asynchronous component is tested and a "real world" exercise with the Deaf community.
5. *Final version*: after validation is completed successfully, the final version is then released. The final version includes the integration of the refined reliable synchronous and asynchronous components within SIMBA.

3.2.2 Component-based software engineering

One of the benefits of using the evolutionary development process, that includes exploratory prototyping, is the fact that it uses a rapid development and deployment mechanism. CBSE focuses on the process of defining, implementing and integrating loosely coupled independent components into systems (Sommerville 2004). It is based on a reuse-oriented approach that relies on a base of reusable software components and an integration framework for the reusable components (Figure 3.2).

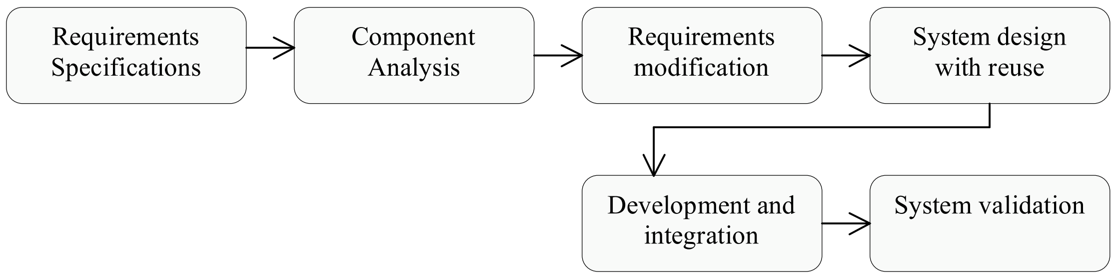

Figure 3.2 Component-based software engineering

Component-based software engineering leads to a faster delivery of software as it makes use of software component reuse as well as an integration framework. The reusable components are modified in such a way that it meets the user requirements.

The CBSE stages are defined as follows:

1. *Component analysis*: once the requirement specifications are received, a search is made for components to implement the specification. In this research a search for reusable components were made for reliable asynchronous communication. Two Application Programming Interfaces (APIs), based on MOM, were identified in aid of finding a feasible solution: JORAM and NaradaBrokering. Both these APIs provided reliability in terms of message delivery, was based on an open-source platform and implemented in Java. For this thesis the NaradaBrokering API was selected as the reusable component to be used in the building of a robust application. This was because NaradaBrokering's reliability scheme was less complicated to understand and easier to integrate within the overall communication framework.
2. *Requirement modification*: after finding a valid reusable component(s), the user requirements are modified to reflect the available components. If requirement modification is impossible, the search for reusable components continues. In our case the NaradaBrokering API provided a good solution for guaranteeing the asynchronous component of the messaging framework. Thus, no further search was made in aid of finding a solution to reliably delivery asynchronous messages.
3. *System design with reuse*: during this stage a framework is organized to cater to the reusable components that were identified. If no reusable component are found,

new software are implemented at this stage. In this phase the NaradaBrokering API was meticulous studied, components used to provide reliability were extracted and an exact framework was developed to situate these components within a Deaf Telephony application.

4. *Development and integration*: all reusable components are then integrated to develop the new system. With the use of the NaradaBrokering API, components used to ensure reliability were integrated in the framework mentioned in the previous step.

Software reuse compliments the evolutionary process as both of these process try to rapidly deploy software. The benefits of software reuse are listed in table 3.1.

Benefit	Explanation
Increased dependability	Reused software is usually tested making it more dependable than new software. This is because the design and implementation faults are already defined and rectified.
Standards compliance	Software components that are standardized, by a known standardization body, are very dependable as many aspects of the components are critically analyzed by a neutral body.
Accelerated development	Reusing and modifying software components speed up system deployment as both development and validation time is reduced.

Table 3.1 Benefits of software reuse

Three compelling reasons are listed that motivates the use reusable software components. These are situated within the Deaf Telephony enhancement process to provide reliable delivery in a semi-synchronous environment.

3.3 Empirical Evaluation

The overall approach of an evaluation methodology is to determine the key software lifecycle activities (Weiderman *et. al* 1987). The activities help to construct a basis for which experiments maybe conducted to extract evaluative information. There are six phases within an evaluation methodology (Figure 3.3).

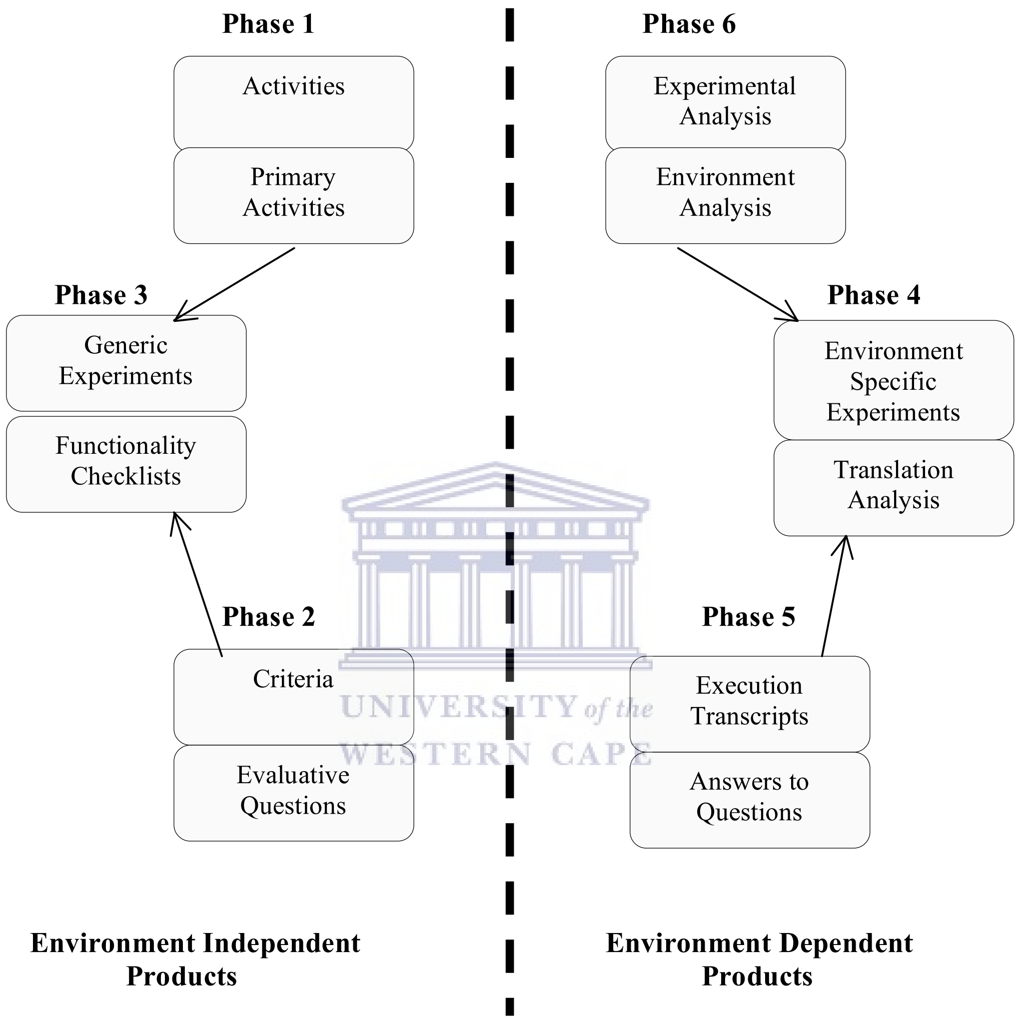

Figure 3.3 Components of the evaluation methodology

The overall approach of the evaluation methodology is to determine the key software lifecycle activities. The activities are used to execute experiments that are designed to extract evaluative information.

Phase 1 identifies and classifies the software development activities. In this phase exploratory prototyping and CBSE is introduced. Phase 2 establishes evaluative criteria for the developed software. There are two different evaluative criteria for both synchronous and asynchronous forms of communication. In phase 3 generic experiments are conducted that represents potential real scenarios. These experiments are conducted in the computer science laboratory for both mediums of communication. After the generic experiments are created and conducted and a successful outcome is observed, it is then implemented within the target environment. Our target environment is DCCT. This is observed in phase 4. Once the implementation within the target environment is successful, raw output data is generated that is used to answer the evaluation questions. The last phase is concerned with the analysis of the results. The results are summarized and an impression is obtained from implementing and performing the experiments within the specific environment.

3.3.1 Laboratory experimentation

To evaluate the integration of NaradaBrokering's MOM architecture within SIMBA an experimental test is conducted to examine the efficacy of NIMBA (Naradabrokering integrated with SIMBA). The evaluation strategy is based on a software agent approach to collect data from an aerial view point (Figure 3.4) (Hilbert and Redmiles 1998). The software agent approach is used in a "bounce" scenario where clients are temporally disconnected and reconnected from the system to determine the feasibility of the reliable asynchronous transport medium. Client(s) are disconnected from the system and reconnected after a period of time. During the period of disconnectedness messages are persistently logged in a database. During this time, the logging facility is monitored to make sure that messages are logged successfully.

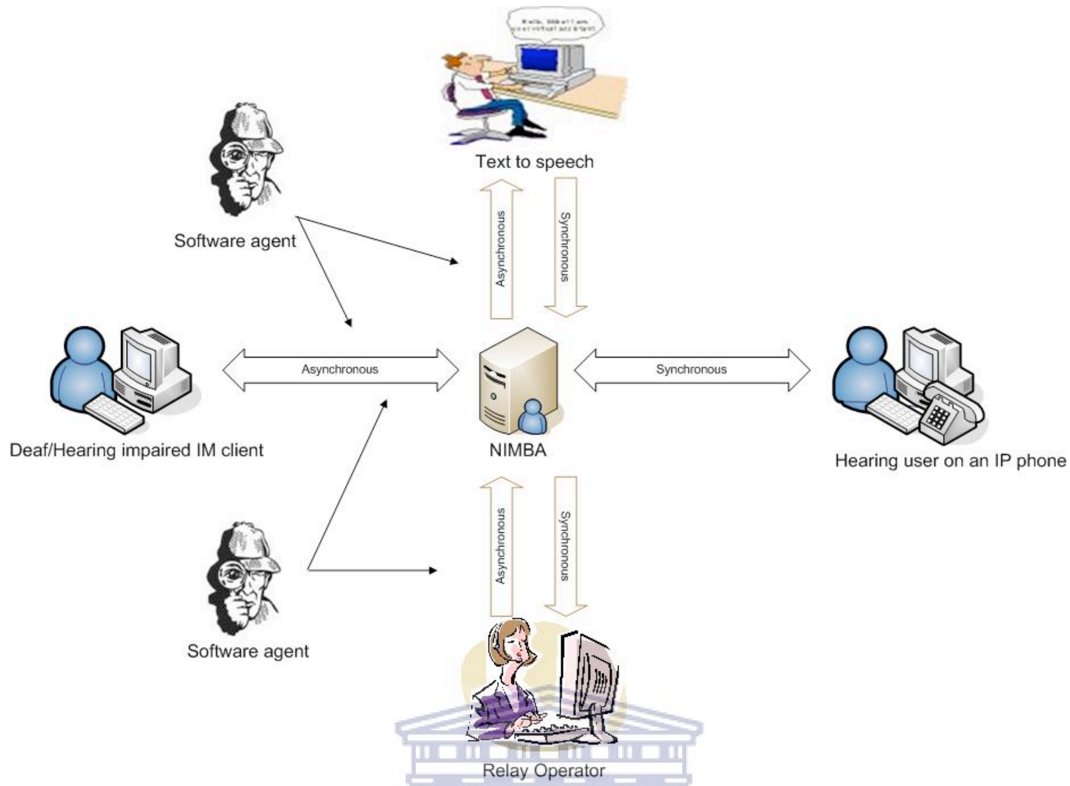

Figure 3.4 Automated software agent approach to data collection

In this scenario an automated software agent approach is used to collect data about asynchronous communication in NIMBA. Clients are temporally disconnected from the system and interaction within the system is monitored using the "bird's eye" approach.

3.3.2 Ethnography

Kitchenham and Pfleeger (2002) delineate research questionnaires in terms of having a benefit on the community under study which in turn motivates them to provide a complete and accurate response. In their paper, it is noted that several key pieces of information should be supplied with the questionnaires:

- What the purpose of the study is.
- Why it is of relevance to the community under study.
- Why each individual's participation is important.
- How and why each participant was chosen.
- How confidentiality will be preserved.

An important consideration with regard to research questionnaires is the impact of the researchers own bias towards the research. The researcher has a good idea of what is sought within the research, and the research could be structured in such away that reveals the researcher's own bias. Bias towards a particular outcome can be linked towards the influence of respondent replies in terms of:

- The way a question is structured.
- The number of questions asked.
- The range and type of response categories.
- The instructions to respondents.

In this thesis, bias is avoided by conducting the following:

- Neutral questions are developed in such a way that wording does not influence the way the respondent thinks about and answers a particular question.
- Enough questions are asked to cover the topic with regard to the research question.
- Special attention is paid towards the order of questions so that the answer to one particular question does not influence the response to the next question.
- Unbiased, clearly written instructions are presented towards the respondents.

3.3.2.1 Pre-Trial Questionnaire

In order for us to understand the importance of providing reliability in a Deaf Telephony application, a couple of background questions are asked in terms of the usage of SIMBA. These questions are asked before NIMBA is introduced to the community under study. The pre-trial questionnaire thus focuses on familiarizing the Deaf community under study with the SIMBA system. The intent of this questionnaire is to determine how unreliable communication within SIMBA influences the Deaf person's perception of message delivery. The questions are asked in such a way to determine if reliable message delivery is a factor within a Deaf person's viewpoint of communication. With this questionnaire, the intent of the outcome is to clarify the Deaf person's expectations of reliable communication in their day-to-day lives.

3.3.2.2 Post-Trial Questionnaire

After the pre-trial questionnaire is completed, NIMBA is introduced to the Deaf community. Observations are made after which a post-trial questionnaire is conducted. The post-trial questionnaire focuses on determining how reliable communication, within NIMBA, influences the Deaf person's perception on message delivery. The questions are asked in such a way to determine if reliable communication increases user confidence levels to use NIMBA. Again, the aim of the questionnaire is to determine if the software that was developed met the expectations of the Deaf community and if the relationship between developer and community had any benefits.

3.4 Summary

In this chapter, several methods, for software development and evaluation, are listed and discussed. These include an evolutionary development process called exploratory prototyping, CBSE and ethnography. Within each of these methods, the general process is discussed and its contribution to the system implementation is mentioned.. To evaluate the developed software, two different methods for synchronous and asynchronous communication are described. For both synchronous and asynchronous communication an automated software agent approach to data collection is described. Software development for actual communities is supported by ethnographic observations and evaluation with questionnaires. The next chapter gives a detailed explanation of the developed system. This includes all technical aspects surrounding the implementation of the reliable synchronous and asynchronous components. The next chapter uses the design and implementation decisions made in this chapter to reach a feasible solution to answer the research question

Chapter 4 SYSTEM DESIGN

In the previous chapter FEC and MOM were proposed as feasible guaranteed delivery solutions for synchronous and asynchronous communication, respectively. In this chapter a detailed discussion is included about the architectural considerations for both solutions. Both solutions are integrated around SIMBA to provide reliable communication for a Deaf Telephony application. For integrating MOM within SIMBA, the NaradaBrokering API is used. This API is used to provide a solution for reliable asynchronous communication. To guarantee synchronous communication, FEC is used to encode redundant data that is conveyed within a RTP header.

This chapter looks at the technical solution in more detail by introducing and discussing the integration of NaradaBrokerings's MOM and FEC for the development of an extended version of SIMBA called NIMBA. The chapter revisits the software requirements, specified in chapters 1 and 3, after which a detailed discussion of the technical solution for the software requirements is specified.

4.1 System overview

The overall goal of NIMBA is to provide a platform for reliable communication in a mixed synchrony environment. This is achieved by providing reliability for both synchronous and asynchronous communication within a Deaf Telephony application. For synchronous communication, reliability is achieved by using an error detection and correction scheme, called FEC, while a combination of ACKs and NAKs is used to provide reliability in asynchronous communication. According to the principles of software development, user requirements should be specified and analyzed to develop an appropriate solution. The next section looks at the requirements in more detail. Thereafter, an overall system deployment strategy is presented.

4.1.1 User requirements specification and analysis

Chapters 1 and 3 mentioned the initial requirements for NIMBA. A series of experiments, conducted with SIMBA, revealed an extension to SIMBA in the form of reliable communication in a mixed synchrony environment. Analyzing the requirements made it clear that the best way to address the user requirements was to treat the problem as two separate entities, that being synchronous and asynchronous communication. After an extensive investigation into the field of both reliable synchronous and asynchronous communication was conducted, an API was identified to help mitigate the problem of reliability. In the embryonic stage of development it was thought that the NaradaBrokering API would be used to address both reliable synchronous and asynchronous communication within SIMBA. This was however not to be. The API does provide a platform to build synchronous and asynchronous applications with its theme of "reliability". The API provided a good solution for reliable asynchronous communication but was incorrectly considered to be a solution for reliable synchronous communication and was thus elided as a potential solution for synchronous communication. Since the thesis is concerned with an empirical method of finding a solution, the NaradaBrokering API was seen as an incorrect API for a synchronous medium of data transport within SIMBA. Thus, another cycle of investigation was made in the field of reliable synchronous communication. This cycle revealed protocols, algorithms and standards for reliable synchronous communication. These are identified in Chapter 2. FEC was selected as the best solution for reliable synchronous communication because it has the capabilities of allowing the receiving device to detect and correct less than a predetermined number of fractions of bits or symbols that is corrupted by transmission errors.

4.1.2 System deployment

NIMBA allows communication to occur over both IP and PSTN networks as the users of the system can be located on a cell phone, landline phone, IP phone or even on a computer (Figure 4.1).

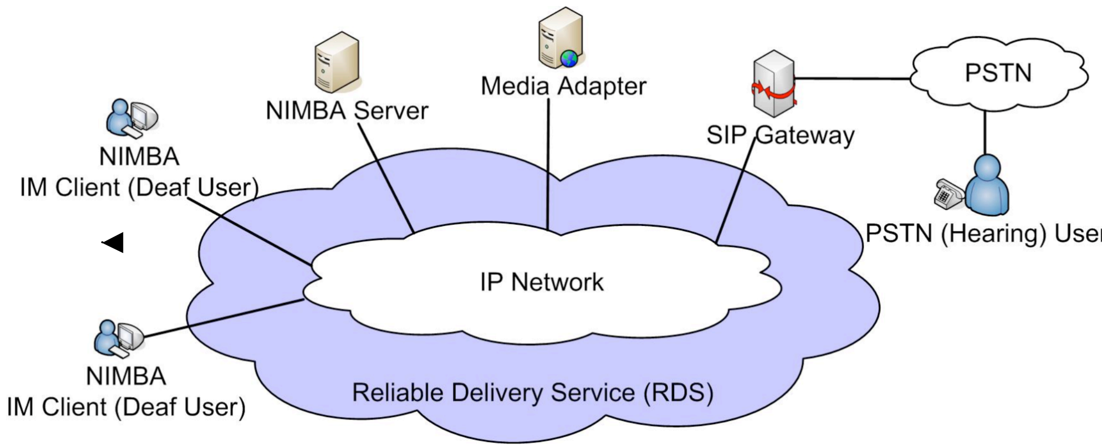

Figure 4.1 Partial guarantees in NIMBA

Reliability is provided within the RDS cloud. This cloud guarantees both asynchronous and synchronous messages sent over the IP network. MOM guarantees the asynchronous messages with FEC providing reliability for synchronous messages, provided that the transportation network is IP-based.

Allowing communication to occur over both IP and PSTN networks became a problem when reliability constraints were considered in SIMBA. When the initial trials with SIMBA were conducted, SIMBA had no control over the PSTN network which made it impossible to modify RTP packets before it was sent to its recipient. To solve this problem one of two things had to be done:

- Only guarantee communication up to the point where data is sent over the PSTN network. This would in effect only be asynchronous communication and would thus only guarantee IM messages sent within SIMBA.
- Instead of using a hearing user on a telephone use an IP phone as the primary communicating device. This would allow RTP sessions, between recipients, to be monitored over an IP network.

In the first scenario data is partially guaranteed over an IP network as the system guarantees the transmission of a Deaf person's message to the MAS only. For a Deaf Telephony application, this is unacceptable since communication occurs between a hearing and a Deaf user. Thus the second scenario looks more plausible than that of the first since communication is guaranteed between both parties even though it is based entirely on an IP network. The use of the second scenario allows the gateway to be removed from the architecture since both synchronous and asynchronous communication occurs over the IP network (Figure 4.2).

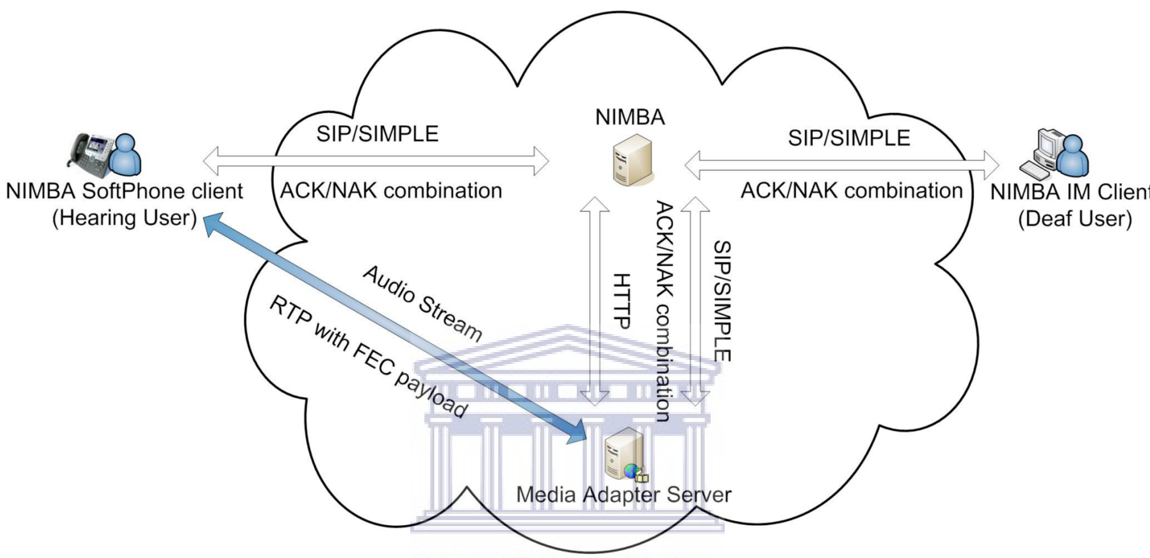

Figure 4.2 Mixed synchrony guarantees in NIMBA

Since the hearing user now communicates over an IP network there is no need for a gateway to bridge to the PSTN network. Thus all communication occurs over the IP network making it possible to guarantee both synchronous and asynchronous forms of communication. To guarantee the exchange of synchronous messages FEC is used to detect and correct error during real-time transmission. A combination of ACK and NAK is used to guarantee the delivery of asynchronous messages with NaradaBrokering.

Message guarantees are treated as separate entities since messages traverse through a synchronous and asynchronous medium. NIMBA thus acts as a SIP proxy, SIP presence and message persistence server. When a Deaf IM user sends a message to NIMBA, MOM ensures data delivery from the Deaf client to the MAS by combining both ACK and NAK acknowledgement schemes for data recovery. It is here that the Deaf person’s IM message is converted to an audio stream, by the use of a TTS engine. The converted audio is then sent back to NIMBA. To send the audio messages reliably to the hearing client, on the IP phone, FEC is used for error control and correction. Thus an FEC algorithm is used to encode redundant data, used for error recovery, which is added to the RTP payload. When the hearing user responds to a Deaf person's message the exact same operation occurs in reverse. That is, FEC is used to reliably transport the hearing user's audio to NIMBA. NIMBA then forwards the audio to the MAS and instead of using an ASR engine, to convert the audio to text; a human relay operator is used. One of the main reasons a human relay operator is used is because the recognition rate of available ASR tools is not good enough for Deaf Telephony purposes. After speech is successfully converted to text, the human relay operator sends it, via NIMBA, back to the Deaf user. The message transported from the relay operator to the Deaf user is monitored by MOM and hence reliability is ensured. The next two sections explain how MOM and FEC are used to guarantee delivery of both asynchronous and synchronous communication respectively.

4.2 Reliable Asynchronous Communication

In the previous section an overview of the overall NIMBA architecture is given with a diagrammatical representation of the data flow as it occurs in NIMBA. In this section focus is placed on the components that are used to guarantee asynchronous communications within NIMBA. In chapter 2 mention was made of the CBSE process and its applicability to software development. The CBSE process was applied to NIMBA to guarantee the exchange of asynchronous messages. The process was exposed to NaradaBrokering to help achieve the goal of reliable messaging within NIMBA. NaradaBrokering is a distributed event brokering system that is based on a publish-subscribe paradigm and is designed to run on a very large network of broker nodes. Besides its audio and video conferencing capabilities, NaradaBrokering guarantees asynchronous messages in order and in a timely fashion. This is one of two reasons why NaradaBrokering was selected as a solution to guarantee messages within SIMBA; the other being that it is entirely based on Java making it easy to integrate since SIMBA is based on Java. It is important to note that asynchronous messages are guaranteed between a Deaf IM client and a hearing relay operator.

When a Deaf IM client sends a message to NIMBA, he/she becomes a publisher and NIMBA a subscriber. When the relay operator sends a message to the Deaf IM client, he/she then becomes the publisher with the Deaf IM client becoming the subscriber. Next, a detailed method and class relationship of NaradaBrokering's integration within SIMBA is presented.

4.2.1 The Reliable Delivery Service

NIMBA makes use of a Reliable Delivery Service (RDS), which forms part of MOM, to guarantee the delivery of asynchronous messages (Figure 4.3). The messaging infrastructure can be seen as a network of messaging channels. The channels are responsible for the delivery of messages under node or link failures.

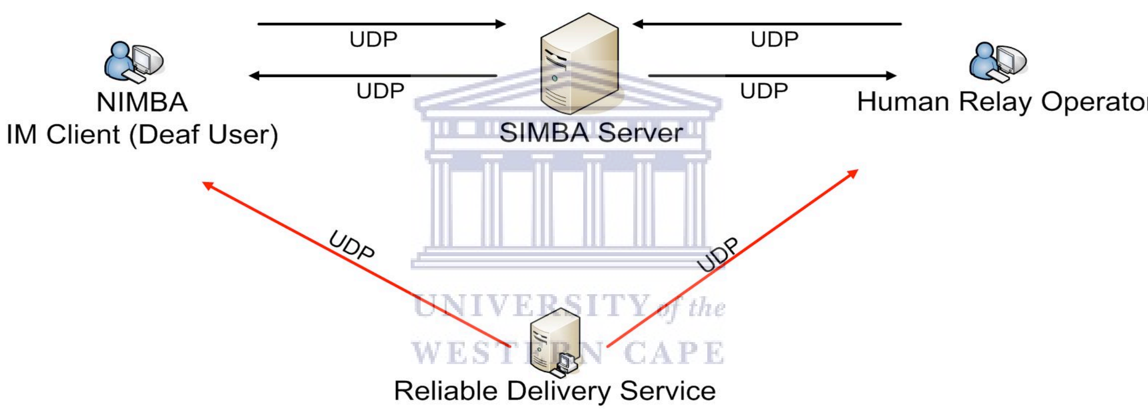

Figure 4.3 Asynchronous monitoring process

RDS is used to monitor all asynchronous messages sent within NIMBA. Thus, in conjunction with MOM, RDS uses a combination of ACKs and NAKs to reliably transport asynchronous messages to their respected recipients.

The main entities of RDS are acknowledgements, a MySQL database, IM clients called entities, profiles and event templates. Each message[3] event, within NIMBA, consists of a collection of headers, content descriptors as well as the payload (actual

[3] Within NIMBA there are two forms of asynchronous messages: IM messages that pertain to the SIMPLE standard and messages conforming to the NaradaBrokering scheme. Both these forms of messages contain the actual payload. To avoid confusion in this section the term *message* refers to the messages used in NaradaBrokering.

data) encapsulating the content. The headers are used to provide QoS related information such as the type, timestamp and dissemination traces. Content descriptors are used to describe information pertaining to the encapsulated content. A combination of content descriptors and the values that these descriptors hold is used to create a content synopsis. A collection of headers and content descriptors are used to create a template for a message event. Two or more events that contain identical sets of headers and content descriptors conform to the same template. Within NIMBA, every entity is assigned a unique identifier that it subscribes with to ensure that message events targeted to it are routed and delivered.

4.2.2 Publish and Subscribe

Entities, such as a Deaf IM client or a hearing relay operator, signify interest in message events that conform to a certain template. Both the Deaf IM client and the relay operator have a unique identifier associated with it. Both entities subscribe to this unique identifier to ensure that messages targeted to it are routed and delivered by RDS. RDS makes use of both ACK and NAK acknowledgments for the delivery of messages, which conform to one of its managed templates, between entities. For every managed template, RDS also maintains a list for which it facilitates reliable delivery. RDS keeps track of entities that are supposed to receive a given template event, as well as entities that have not explicitly acknowledged receipt of these events. All event templates are given a unique identifier (template ID) that is used by RDS to advertise its archival capabilities for a specific event template. This template ID is shared between the Deaf IM client and the relay operator.

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 that allows either the Deaf IM client or the relay operator to keep track of the template events it has received while facilitating error detection and correction.

Publishing

Each time a Deaf IM client or the relay operator sends a message (publishing), archival negotiations occurs between the publisher and RDS. Archival negotiations include the process of acknowledging received template events as well as requesting retransmissions whenever a template event is lost. The entity acting as the publisher creates a companion event, for every template event it generates, to ensure that the archival process is a success (Figure 4.4). This companion event includes information to detect losses that might occur when the template event is transmitted.

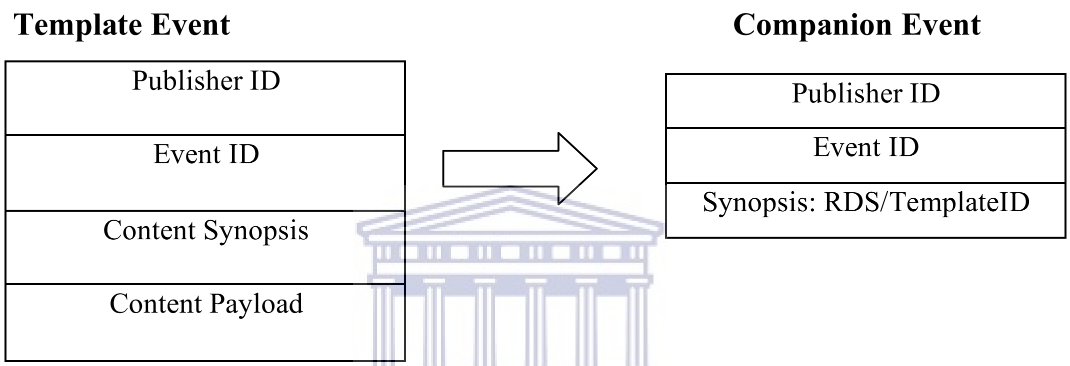

Figure 4.4 An example of a template event and its companion event

The process of archiving template events, at RDS, may be inconvenienced by events lost in transit. To prevent this from happening, the publishing entity produces a companion event with the necessary information that helps detect and prevent losses (Pallickara and Fox 2004).

When RDS successfully receives and archives the template event, it sends a negotiating ACK to the publisher. If it was unsuccessful it sends a negotiating NAK. The publishing entity holds the template event in its buffer until it receives a negotiating ACK that confirms a successful archival of the template event. Whenever the publishing entity receives a negotiating NAK from RDS, it creates another template event with a field indicating retransmission and sends it to RDS. Because the template event's companion event may get lost in transit, RDS issues an archival negotiating NAK event to the publisher requesting the event to be resent. Within NIMBA, both the Deaf IM client as well as the relay operator's entityID is assigned by an Entity Manager (Figure 4.5). This Entity Manager has the responsibility of ensuring that messages are logged with a templateID corresponding to the correct publisher and subscriber in the system. The Entity Manager uses the architecture of RDS to ensure that messages are logged consistently. Thus, the Entity Manager is seen as a separate entity within the overall architecture of NIMBA.

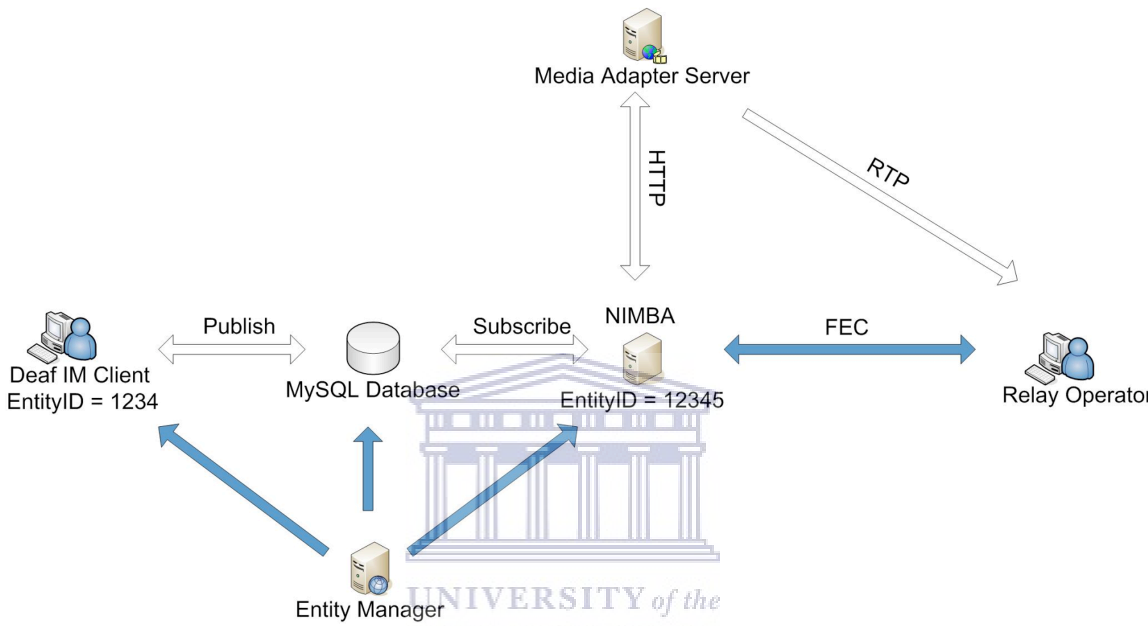

Figure 4.5 An Entity Manager within NIMBA

The EntityManager is responsible for creating an unique EntityID for both NIMBA and the Deaf IM client. This EntityID is used by RDS to match an entity to a managed template. The EntityManager is also responsible for archiving template events to the MySQL database.

Subscribing

RDS issues an archival notification (Figure 4.6a) once a given template is successfully archived. This archival notification is sent to the subscribing entity which is either the Deaf IM client or NIMBA. An invoice event is created to encapsulate the exchange of information between the subscribing and publishing entity as well as RDS. This invoice event corresponds to the set of template events received and can also request to transmit missed templates 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 (Figure 4.6b).

Template ID
Event ID
Sequence Number
Previous Sequence Number
Synopsis: Content Synopsis of archived event

(a)

Template ID
Entity Identifier
ACK/NAK
Sequence Number
Synopsis: RDS/Template ID/ Invoice

(b)

Template ID
ACK-Response
Sync-point advance
New sync-point
Missed sequencing information
Synopsis: Entity ID/ Invoice

(c)

Template ID
NAK-Response
Sequence Number
Inventory Event
Synopsis: Entity ID/Invoice

(d)

Figure 4.6 Message creation for reliable asynchronous communication.

This figure represents the data structures for the archival notification, ACK invoice event, ACK Response invoice and the recovery 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 messages which were not explicitly acknowledged using invoice events. When RDS receives an 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 (Figure 4.6c). 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 receives 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 recover event (Figure 4.6d).

4.3 Reliable Synchronous Communication

There are two common software layers that constitute the overall architecture of VoIP technology. These layers are defined as the application layer, which consists of the main program and user interface, as well as the RTP stack used at the lower layer. Within a VoIP application, the application layer invokes the RTP stack and controls its operation accordingly. In this section a detailed discussion of RTP is introduced to manifest the strenuous task of integrating FEC within the RTP header. The discussion includes a detailed architectural discussion of the standardized RTP stack (Schulzrinne *et. al* 1994). Thereafter, the integration of the redundant encoded data within the RTP header is discussed.

4.3.1 RTP

On the 22nd of November 1995 the Internet Engineering Steering Group (IESG) approved RTP as an Internet proposed standard (www.cs.columbia.edu/~hgs/rtp/). The protocol has since then been published as:

- RFC 1889, "RTP: A Transport Protocol for Real-Time Applications".
- RFC 1890, "RTP: Profile for Audio and Video Conferences with Minimal Control" (Schulzrinne 1996).

RTP supports the transport of real-time media, such as audio and video, over packet networks and is used by signaling protocols such as SIP and H.323. RTP is independent of both the network and the transport protocol and is used, in most cases, over UDP. The protocol allows the types of data being transported to be identified as well as the determination of which order the packets should be transported in. The RTP specification does not, however, guarantee the delivery of data packets timely and in order. This process is left up to the receiver to reconstruct the sender's packet sequence and use the packet header to detect any packet loss that might have occurred. RTP does, however, allow the quality of data packets to be monitored by using a control protocol called the Real Time Control Protocol (RTCP) (Figure 4.7). This protocol provides control and identification mechanisms for all RTP data transmissions.

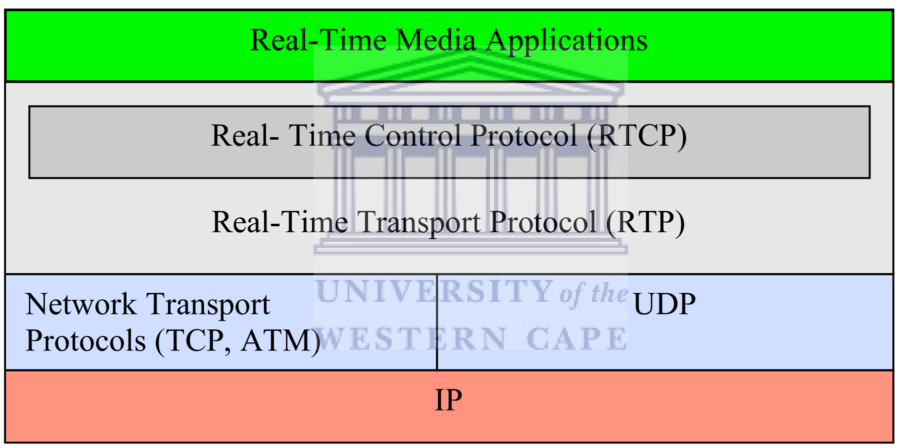

Figure 4.7 RTP architecture.

The architecture of the RTP specification is created out of four "layers" that include actual real time applications built on top of the RTP/RTCP stack. Below such a stack, the actual transport medium is identified and in most cases UDP is used for the transmission of RTP data packets.

The RTP Header Format

Media data transmitted across a SIP established session, for example, is actually a series of data packets collectively called an RTP stream. Each RTP data packet within the stream has two parts associated with it: a header and the actual payload. The RTP header provides necessary information to the receiver that allows the reconstruction of media (Figure 4.8).

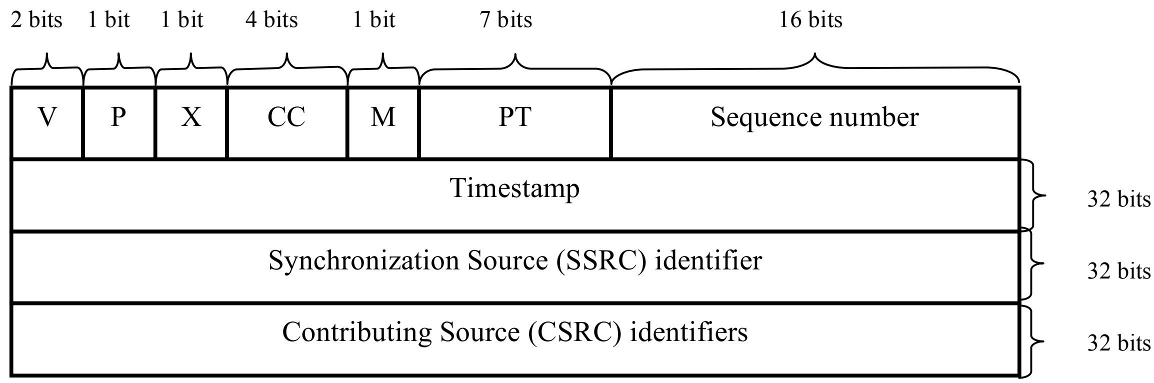

Figure 4.8 RTP data packet header format.

The RTP header is used to provide information to the receiver that allows it to reconstruct media according to the accompanied information provided by the RTP header.

The header contains the following fields:

- *RTP Version number (V):* The current version number is two.
- *Padding (P):* This bit is used to indicate that some encryption algorithm is padded to the end of the payload.
- *Extension (X):* The extension bit allows the header to be extended by adding additional information to the RTP header.
- *CSRC Count (CC):* Indicates how many CSRC identifiers follow the fixed header.
- *Marker (M):* This bit is used to allow frame boundaries to be marked in the packet stream.
- *Payload Type (PT):* This bit is used to describe the payload format.
- *Sequence Number:* The sequence number is used to identify a particular packet's position in a sequence of packets.
- *Timestamp:* The timestamp is used to reflect the sampling instant of the first octet in the RTP header.
- *SSRC:* SSRC is used to identify the synchronization source. When the CRSC count is equal to zero the payload source is used as the synchronized source. When the CRSC count is non-zero, the SSRC field is used to identify the mixer.

- *CSRC:* There can be up to 16 contributing sources for the payload. The contributing sources form a mixed form of data that is used to construct payload.

4.3.2 Redundant information for RTP

When packet loss occurs, the missing information can be reconstructed at the receiver from the redundant data that arrives in the following packets, provided that the average number of consecutively lost packets is small (Perkins *et. al* 1997). Yajnik, Kurose and Towsley (1996) proved that the packet loss patterns observed in the Internet allows the use of a scheme to transmit redundant information. The next two sections concentrate specifically on the RTP payload format for the transmission of audio data encoded in a redundant fashion. The first section describes the problem of implementing an RTP payload incorporating redundant information. Thereafter, the architecture of the RTP payload with redundant information is introduced and discussed.

4.3.2.1 The problem of redundant information

A scheme that encodes redundant information within the RTP payload format needs to address the following issues:

- All RTP packets have to carry a primary encoding data scheme as well as one or more redundant encoding data schemes.
- For each piece of redundant information that is encoded, an encoding type identifier needs to be produced.
- Redundant information can have a size of variable length which in turn allows encoding to be of variable length. Thus each encoded block in the packet has to have a length indicator.
- The redundant blocks of data will have a time interval different to that of the primary encoded data. Thus, each block of redundant encoding requires its own timestamp.

There are two solutions that mitigate the problem of conveying redundant encoded data within an RTP packet. The first solution extends the RTP header with the purpose of holding the redundant encoded data. The second solution specifies one or more additional payload types to hold the redundant information. Extending the RTP header involves the process of appending redundant data to the primary payload and conveying the FEC technique using an RTP payload type. The advantage of having such a technique is that it allows applications, which do not make use of redundancy, to discard the redundant data and process the primary data. This however is impractical for two reasons:

- For every primary and redundant encoded data a new payload type needs to be produced. This is a problem when a multitude of redundant data is encoded and an encoding indicator is required for each redundant data block.
- Each time data is encoded; a timestamp is created and placed in the timestamp field before transmission. When redundant encoded data is appended to the primary payload, the time of creation is referred to the primary encoded data only. Redundant data is encoded within a different time interval and thus requires its own timestamp. The amount of bytes needed to carry this timestamp can be reduced by encoding it as the difference between the timestamp of the primary encoded data and the timestamp of the redundant encoded data.

There are two possible schemes for assigning additional RTP payload types for redundant encoded data in audio applications (Schulzrinne 1996):

- For each combination of primary and redundant payload types a dynamic encoding scheme may be defined that makes use of the RTP dynamic payload type range.

- An RTP packet, containing redundant information, may be presented by using a single fixed payload type. The fixed payload type may then be assigned to a static or dynamic RTP payload type.

In the next section a detailed discussion of the second scheme is presented.

4.3.2.2 Payload format for redundant information

A single fixed payload type, representing redundant encoded data, is in essence a container that encapsulates multiple payloads into a single RTP packet (Figure 4.9). This scheme allows any amount of redundant data to be encapsulated within a single RTP packet. The only drawback of this scheme is that it allows a small amount of overhead to occur as each encapsulated payload is preceded by a header indicating the type of data that is enclosed.

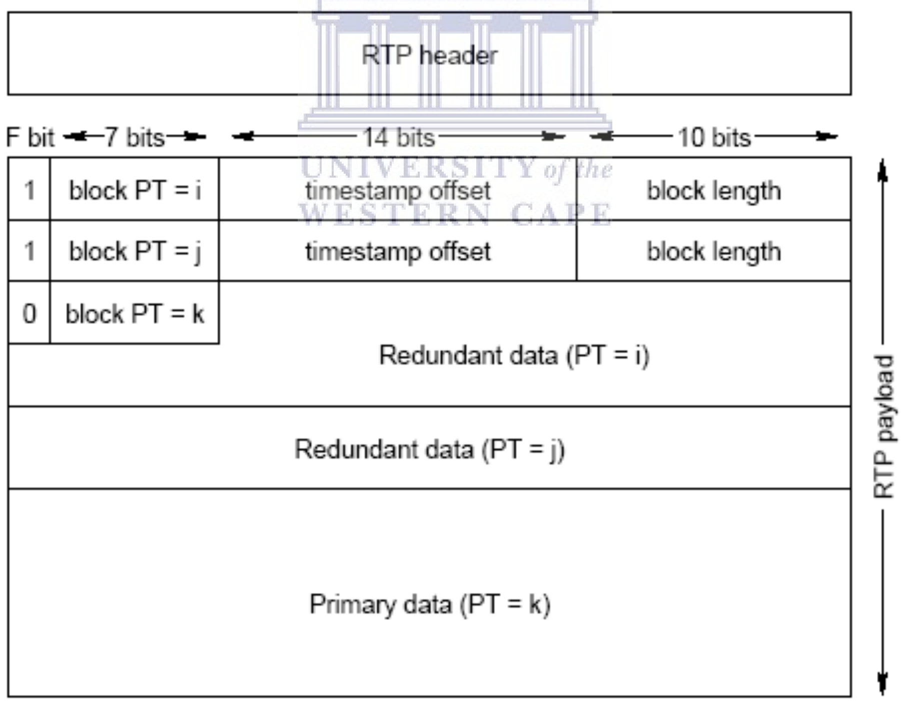

Figure 4.9 Payload format for redundant data

In this figure a number of additional headers are defined which is placed after the RTP header. Below the additional headers a number of data blocks are described which contain the standard RTP payload as well as the redundant encoded data (Perkins et. al 1997).

Redundancy is indicated by the payload type within the RTP packet. Below the RTP header a number of additional headers are defined that are used to specify what kind of encoding is carried by the packet. The bits in the header are described as follows:

- *F bit:* This is the first bit in the header and is used to indicate whether another header block follows. 1 indicates that another header block will follow and 0 indicates that this is the last header block.
- *block PT:* The size of this data block is 7 bits and is used to represent the RTP payload type for the particular block.
- *timestamp offset:* This data block is created out of 14 bits. It is used to hold an unsigned offset of the timestamp relative to the timestamp given in the RTP header. The reason an unsigned offset of the timestamp is used is to show that redundant data must be sent after the primary data. Because the timestamp of the redundant data was created in a different time interval, it can be calculated by subtracting the unsigned offset from the current timestamp.
- *block length:* This block of data is represented by 10 bits. It is used to indicate the length in bytes of a particular data block excluding the header.

The use of an unsigned timestamp offset does not allow redundant encoded data to be sent before the primary encoded data. If a signed timestamp offset is used instead of an unsigned offset, the range of the timestamp offset is reduced as the size of the field is taken above 14 bits, limiting the block length field. Limiting the redundant encoded data to be transmitted after the primary encoded data causes fewer problems than limiting the size of the other fields. The timestamp offset for redundant encoded data is measured with the same units as that of primary encoded data. This includes audio samples with the same clock rate as the primary encoded data.

The primary encoded block header is placed last in the packet making it possible to leave out the timestamp and block-length fields from the header. This is because these fields can be determined from the RTP header and the overall length of the packet.

4.3.2.3 Redundant encoded data

To encode data redundantly, an FEC algorithm is used that is based on the Vandermonde matrices (Rizzo 1997). The algorithm makes the assumption that in computer communications, error detection is provided by the lower protocol layers which uses checksums to discard corrupted packets. This effectively leaves the upper protocol layers to deal with missing packets in a stream. The redundant data is encoded in such a way that k blocks of source data are encoded at the sender to produce n blocks of encoded redundant data such that a subset of k blocks is used to reconstruct the source data. This technique allows the receiver to recover from up to n-k losses of n redundant encoded data. The Vandermonde matrix is a $n \times k$ matrix that represents a linear code such that the source data can be reconstructed by using k data from the encoded $n \times k$ matrix.

4.3.3 Tools used for Implementation

For implementing redundant data with RTP, Java's open source JMF implementation was used. This API makes use of an RTP packetizer class that is responsible for encapsulating multiple audio samples and distributes them into packets that can be streamed over the Internet. The RTP depacketizer class does the reverse process by extracting audio samples from a stream of RTP packets. For the purpose of integrating FEC within the RTP header, a custom packetizer and depacketizer were built. The SessionManager class allows information to be entered into a dynamic payload used within RTP. This dynamic RTP payload information is mapped from the RTP payload ID to the FEC redundant encoded data. For the mapping process the Format object is used to associate the RTP payload ID with the FEC redundant encoded data. This information is sent to the SessionManager class by using its addFormat method. The custom packetizer and depacketizer are assigned a payload number that is associated with the RTP format in the SessionManager's registry. To encode the redundant data, with the FEC algorithm, a Vandermonde matrix API was used. This API is provided by OnionNetworks (http://onionnetworks.com/developers). The mathematical API provides methods and procedures to implement a Vandermonde matrix for the construction of the $n \times k$ matrix.

4.4 Summary

In this Chapter a detailed discussion of both reliable synchronous and asynchronous implementations, within NIMBA, was presented. For asynchronous communication, acknowledgements were used between a Deaf IM client and NIMBA to ensure that messages are reliably delivered and persistently logged. This acknowledgement-based interaction is based on the NaradaBrokering API that was used in the CBSE process. Within the architecture, a MySQL database was used to log all messages for later retrieval. Synchronous messages were guaranteed by the use of an FEC algorithm that made use of Vandermonde matrices. To take care of the problem of matrix encoding, an open source implementation of the Vandermonde matrix was used. This API was developed by the onionnetworks group and the code was made available for re-use and modification. The next chapter gives an architectural overview of the experimental process undergone to show that the integration of both NaradaBrokering and FEC was successful and that messages were delivered reliably between sender and receiver.

Chapter 5 EXPERIMENTAL DESIGN

In chapter 3 an empirical evaluation method was introduced in terms of software agents. These agents are used to monitor reliability within NIMBA by using an aerial overview of the system and in the process using a birds' eye scenario to uncover error resilience. This chapter discusses the process of monitoring reliability of both synchronous and asynchronous communication channels using this overview. Included in this chapter are discussions of both hardware and software environments that were used to test the feasibility of the overall system. Since the thesis is situated within a Deaf Telephony domain, a social study in terms of ethnographic observations, trials and questionnaires is presented to evaluate the social aspects of the developed system.

5.1 Experimental Setup

NIMBA is an extension to SIMBA and thus uses SIMBA's hardware environment with integration and modification occurring in the software environment. Both software and hardware environments were used to help manifest the efficacy of using a socio-technical evaluation strategy. Within the social evaluation strategy users are exposed to the software environment only as it would be incongruous to leave the arduous task of network/hardware configuration up to a Deaf person. The next couple of sections introduce both hardware and software environments used to evaluate NIMBA.

5.1.1 Software Environment

The software environment includes original components used within SIMBA as well as additional components that were used to provide reliability for both synchronous and asynchronous forms of communication (Table 5.1).

Name	Description

Operating System (OS)	Microsoft Windows 2000 with service pack 3.	
SIMBA	A Deaf Telephony application	The initial SIMBA components
MAS	A web server used to convert text to speech using a TTS engine.	
NIST-SIP IM	A Java based IM client used for IM communication within NIMBA.	
SIP Residential Gateway (SIPRG)	An IP telephony gateway used to bridge between an IP and PSTN network.	
Java 2 Platform Standard Edition (J2SE) 1.5	A software development package that provides a complete environment for application development on desktops and servers (http://java.sun.com/j2se/index.jsp).	Tools used to integrate reliability.
Operating System (OS)	Microsoft Windows XP professional with service pack 2	
MySQL 4.1 database	A database management system used as the storage medium for asynchronous messages (www.mysql.com).	
Mysql-connector-java-3.0.16-ga.zip	A MySQL database connector package written in Java.	
Apache Ant	A Java-based build tool used to compile and run NIMBA	

Table 5.1 Software environment for evaluating NIMBA

5.1.2 Hardware Environment

For the socio-technical evaluation strategy two hardware environments were used. The technical environment was laboratory based and included a laptop computer and Personal Computer (PC) connected to a wired network. The social environment included PCs connected to a wireless network. A complete hardware list is presented in table 5.2.

Name	**Description**	
PC	**Laboratory testing:** Laptop computer with Intel Celeron 1.4 GHz processor, 256MB RAM, 40GB Hard drive. PC with Pentium 4 2.4 GHz processor, 512 MB RAM, 80 GB Hard drive.	**DCCT testing (refer to Figure 33):** SIMBA 1: Pentium 4, 3.2 GHz processor, 1024MB RAM, 120GB Hard drive. SIMBA 2 + 3: Pentium 4, 2.8 GHz processor, 512MB RAM, 80GB Hard drive. CLIENT 1: Celeron 2.4 GHz processor, 512MB RAM, 80GB Hard drive CLIENT 2 + 3: Pentium 4, 2.8 GHz processor, 512MB RAM,
Ethernet	Laboratory testing: 100Mbps, TCP/IP 100M network adapter	DCCT testing: Wireless LAN (WLAN) 802.11b.
Headset	Headset with microphone is used as the input/output device.	

Table 5.2 Hardware environment for evaluating NIMBA

5.2 Technical Evaluation

The technical evaluation strategy of NIMBA was carried out within a laboratory simulated environment between a laptop computer and a PC (Figure 5.1). For simplicity and the use of less hardware, the laptop computer was used by both hearing and Deaf client simultaneously. The PC was used to house the NIMBA server, media adapter as well as the relay operator.

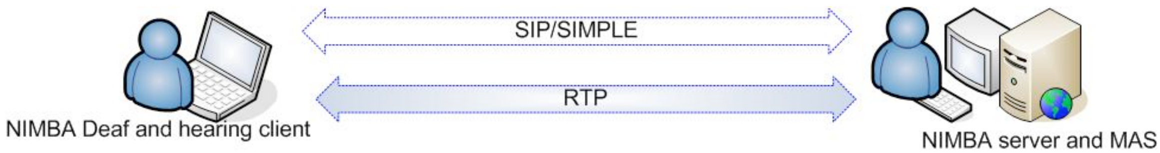

Figure 5.1 Laboratory evaluation of NIMBA

For evaluating NIMBA within the laboratory a laptop computer and PC was used. The laptop computer was used as a Deaf IM client as well as a hearing client on a softphone. The PC was used to house the NIMBA server, media adapter server as well as the relay operator.

This setup allowed observation to occur easier as only two computers were monitored. Testing was conducted in three phases that included unit tests for both synchronous and asynchronous communication and a combined test for both forms of synchrony. The next two sections give a detailed explanation of the evaluation procedure executed for both synchronous and asynchronous forms of communication.

5.2.1 Reliable Synchronous Communication

RTP includes payload type identifiers, sequence numbers, end-of-frame markers, and timestamps to provide loss detection, synchronization and identification. More importantly, a control component is provided (RTCP) to generate feedback of RTP traffic. To evaluate the quality of RTP traffic, RTCP packets are monitored. There are five packets that are considered within RTCP monitoring. These packets are: sender report (SR), receiver report (RR), source description (SDES), BYE and application specific extensions (APP). To monitor RTP and RTCP packets, within NIMBA, a Java-based RTP monitoring tool, called RTPMonitor, was used. The monitoring tool was developed in the department of Computer Science at the University of Columbia. The monitoring tool makes use of a driver and a monitoring application. The driver is based on the Network Driver Interface Specification (NDIS) (www.ndis.com) and is used to monitor all RTCP traffic from a Network Interface Card (NIC). NDIS is a device driver interface that allows a single NIC to support a multitude of network protocols (Figure 5.2).

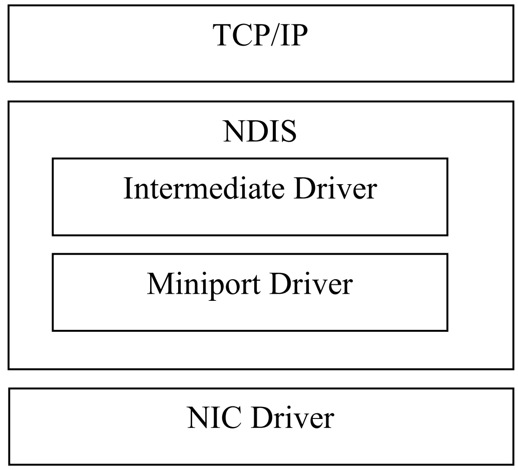

Figure 5.2 NDIS with an intermediate driver

NDIS is used to monitor all RTCP traffic from a NIC. An NDIS intermediate driver is situated between the miniport driver and the protocol driver and is used to monitor any dropped packets.

There are three different drivers defined for NDIS: NDIS miniport driver, NDIS intermediate driver and NDIS protocol driver. To monitor NIMBA's RTP and RTCP traffic a NDIS intermediate driver is used. As can be seen from Figure 5.2, the intermediate driver sits between the miniport driver and the protocol driver such as the Transmission Control Protocol over IP (TCP/IP). An NDIS intermediate driver is layered over an existing NIC driver and in the process creates a virtual adapter. Within the evaluation of NIMBA, the NDIS intermediate driver is used to monitor any dropped packets.

For evaluating the synchronous component of NIMBA, a bottleneck link was created between the MAS and a hearing client on a softphone. This link was congested with a rapid delivery of audio messages. Using the bottleneck link, audio with and without redundant encoded data was tested. All RTCP packets were monitored incase any lost packets were reported by the RTP monitoring tool. For a series of audio messages that were delivered, RTCP packets were analyzed to evaluate error resilience for the synchronous component of NIMBA. The observed quality information is transported from the NDIS intermediate driver to a Graphical User

Interface (GUI) – based monitoring application (Figure 5.3). The GUI used the Windows IO API to communicate with the NDIS intermediate driver.

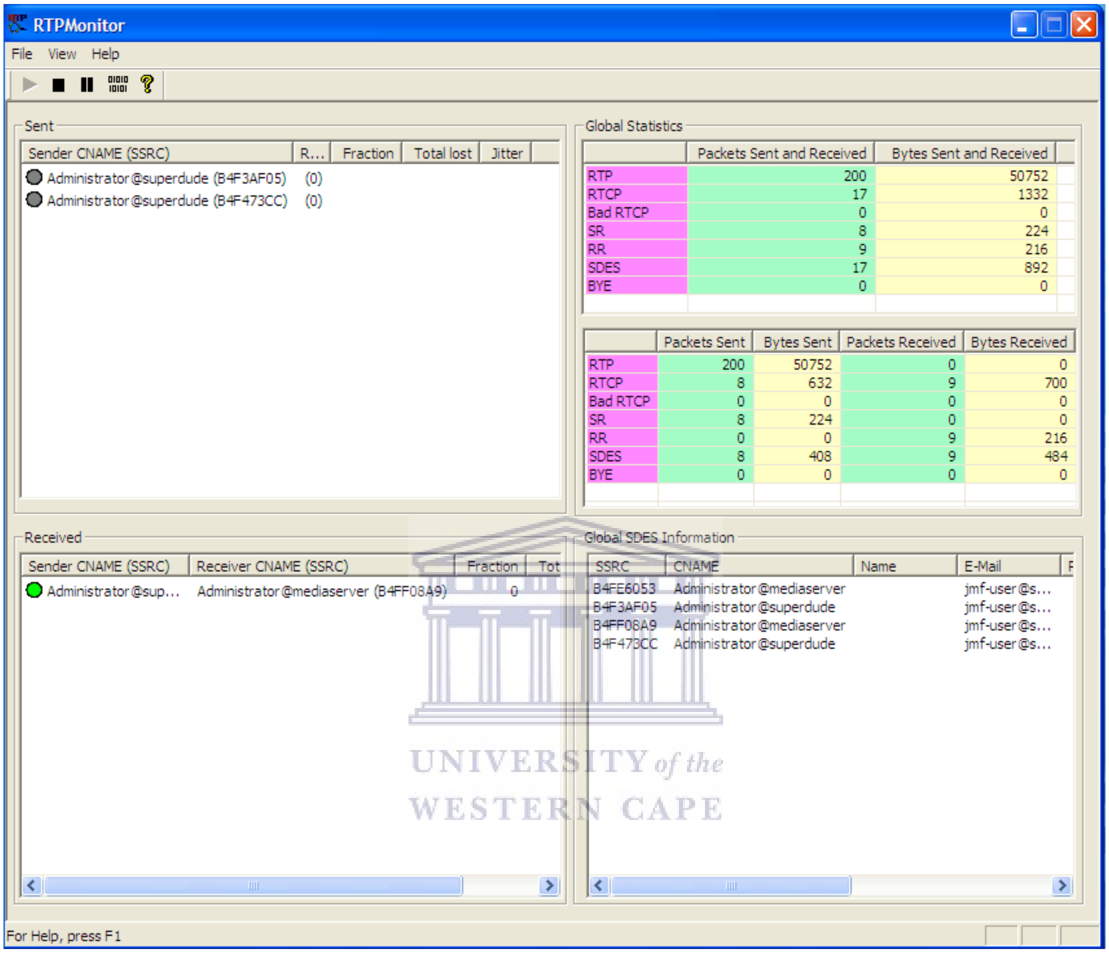

Figure 5.3 A Java-based RTP monitoring tool

The GUI-based RTP monitoring tool is used to visibly depict all RTCP quality information that is retrieved from the NDIS intermediate driver's buffer. This information is periodically retrieved from the buffer to give an overview of the quality of RTP traffic sent and received.

When the monitoring application is started a message is sent to the NDIS intermediate driver requesting it to monitor all incoming and outgoing RTCP and RTP traffic, which is stored in the NDIS intermediate driver's buffer. RTCP quality traffic is periodically retrieved from the driver's buffer. Once all quality information is retrieved from the NDIS intermediate driver's buffer, it is then represented by the GUI.

5.2.2 Reliable Asynchronous Communication

In chapter 4 a detailed discussion of reliable asynchronous communication was presented and a number of communicating entities, used to ensure reliability, were listed. All the mentioned entities were tested, evaluated and documented by the NaradaBrokering development team. Thus re-testing these components within NIMBA is not necessary. There are, however, components that are important to ensure asynchronous message delivery that were tested in NIMBA. The asynchronous component of NIMBA uses a combination of acknowledgements and a database to guarantee message delivery and was thus tested. Thus, for the evaluation of the asynchronous component of NIMBA, three separate tests were conducted to ensure message delivery is guaranteed. These three tests included: persistent logging of messages to the storage medium (database), acknowledgements sent from the NIMBA server to the Deaf client and acknowledgements sent from the Deaf client to the NIMBA server.

Once a message is sent from the Deaf client to the NIMBA server the message is logged within the database. To ensure messages are persistently logged, the number of messages sent to the NIMBA server were recorded and compared to the actual amount of messages stored in the database. Once a message is successfully logged within the database, an acknowledgement is sent to the Deaf IM client. To ensure acknowledgements between the Deaf client and the NIMBA server are successfully sent, communication channels are broken after an acknowledgement from either Deaf client or NIMBA server is sent. Communication channels are disconnected and reconnected after x (where x is defined in the results chapter) amount of time to observe if any acknowledgements were dropped or discarded. Thus, evaluating the asynchronous component of NIMBA resulted in the process of comparing the amount of acknowledgement sent and received after a communication link was broken.

5.2.3 Reliable Semi-synchronous Communication

Unit testing provides a good mechanism for isolating the evaluation of either synchronous or asynchronous components. Evaluating these components separately allows NIMBA's performance to be observed much easier than if it was combined. However, since communication flows between synchronous and asynchronous mediums an overall system evaluation must be performed to identify any problems that might not have been identified during unit testing. Thus instead of evaluating reliability separately both forms of synchrony are evaluated together as one single observation.

A combined system evaluation is thus defined as the process of monitoring a message as it is traversed from a Deaf client to a hearing client and back again while passing through the NIMBA server and relay operator. This form of evaluation gives an overview of message delivery and is used to measure the overall performance of NIMBA. Within our combined system test, the evaluation criteria used for both synchronous and asynchronous components were combined to give an overall performance measurement of NIMBA. Thus, the aforementioned evaluation metrics were used in tandem.

5.3 Social Evaluation

NIMBA was developed to provide reliability for both synchronous and asynchronous forms of communication within the context of a Deaf Telephony application. Since the development of NIMBA was community-centered, a social evaluation scheme seems applicable. The social evaluation scheme is used to indicate whether technical development is complete and if NIMBA addresses the needs of the community under study. It is also used to prove that NIMBA operates successfully outside of a laboratory simulated environment. Social evaluation occurred at DCCT and included three Deaf and two hearing participants. The evaluation environment included a LAN with wireless Internet connectivity (Figure 5.4). The social evaluation was completely

IP-based since communication occurred between a Deaf user using an IM client and a hearing user using a softphone.

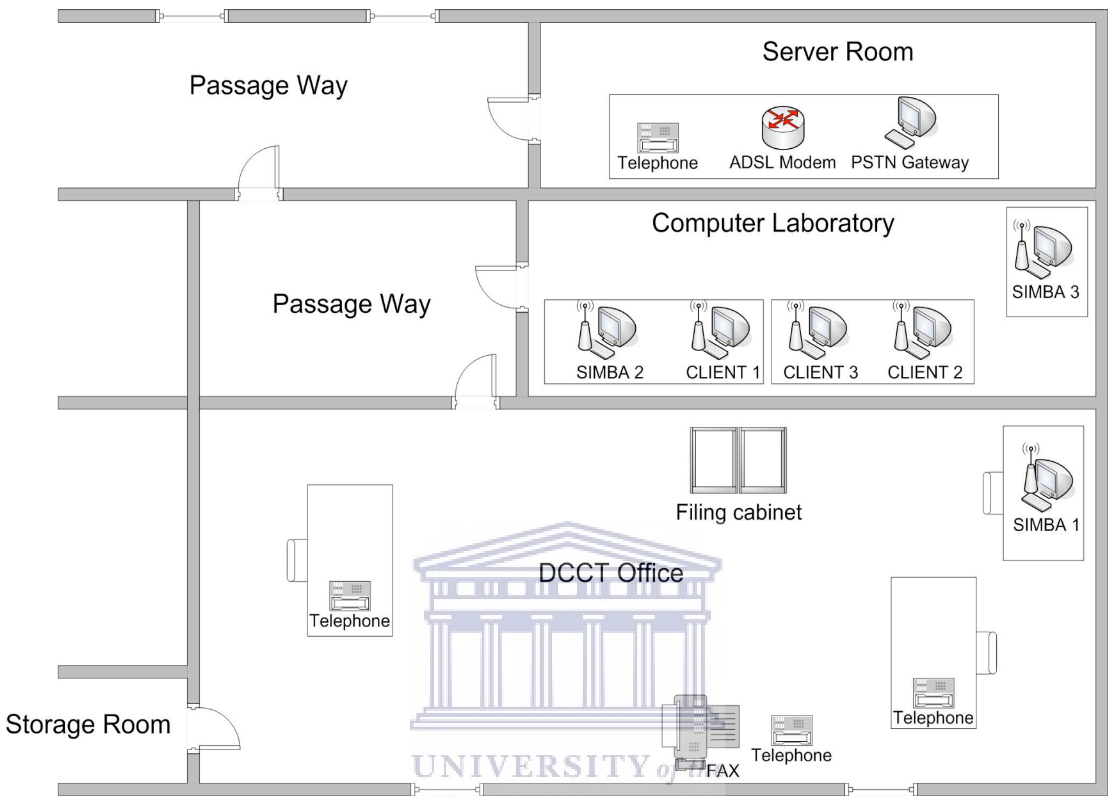

Figure 5.4 Floor plan for the Bastion

The Server Room is used to bridge between the IP and PSTN network. It houses a machine with a telephone card used to connect to the PSTN network. For experimental purposes only the Computer Laboratory and DCCT Office was used since no bridge to the PSTN network was required.

5.3.1 Ethnography

As communication occurred between the Deaf and hearing participants, making use of NIMBA, general observations were made by the developer. These observations included the monitoring of both synchronous and asynchronous communication as well as the Deaf person’s perspective on message delivery notification. Allowing end-users to test the feasibility of NIMBA gave the developer the “room” to technically evaluate NIMBA in terms of its social “space”.

5.3.2 Trials and Questionnaires

Three experiments were conducted between three different Deaf participants and two hearing participants. Each Deaf participant were handed a pre and post-trial questionnaire that was answered with the help of an interpreter. Before the questionnaires were handed out to the participants a detailed explanation, with the help of an interpreter, was given for the purpose of the questionnaires. After the pre-trial questionnaire for all three Deaf participants was conducted three experiments were created between a Deaf and hearing participant. After each Deaf participant completed his/her trial a post-trial questionnaire was handed to him/her. The questionnaires used for the community-centered experiments can be found in Appendices A and B.

5.4 Summary

This chapter presented a detailed discussion of the experimental design that was used to evaluate the overall performance of NIMBA. The discussion included evaluation procedures for reliability in terms of synchronous and asynchronous forms of communication. NIMBA's overall performance was evaluated within a socio-technical environment that made use of laboratory testing as well as a community-centered approach. The chapter also looked at both hardware and software environments used for the evaluation of NIMBA. The next chapter presents and discusses the outcome of the experiments conducted in terms of results that were observed. These results stem from unit testing, overall system testing and social evaluation in terms of trials and questionnaires.

Chapter 6 DATA COLLECTION AND RESULTS

The previous chapter gave an experimental design in terms of the layout of the experimental procedure. The experimental evaluation was discussed in terms of a socio-technical environment that involved experiments conducted in a laboratory as well as in an actual Deaf community. The technical aspects of the evaluation procedure involved the process of using automated software agents to monitor communication links for synchronous and asynchronous transport mediums within NIMBA. The social aspect of the evaluation procedure involved ethnography in terms of trials conducted and questions asked within the Deaf community. These questions were asked by using pre- and post-trial questionnaires. This chapter produces and discusses the results obtained for technical and social experiments conducted.

6.1 Technical Evaluation

The technical evaluation procedure involved three phases that included an evaluation on the synchronous component, asynchronous component, and the combination of both. The main objective of all three phases was to measure NIMBA's reliability for a semi-synchronous environment.

6.1.1 Synchronous environment

To evaluate the synchronous component of NIMBA, a bottleneck link was created between the MAS and a hearing client on a softphone. To congest the link audio, messages were rapidly delivered to a hearing user on a softphone. A packet was considered lost if it was delayed for more than 180ms. This seemed to be an acceptable threshold value as most other researchers used a threshold between 180ms and 200ms to evaluate reconstruction (Wah and Lin 1999). To conduct our experiments, the experimental data had to be prepared prior to experimentation. Three pieces of data had to be selected before an audio packet could be sent to the congested link on the network. They are the frame length, a base coding rate, and a redundancy coding rate. To evaluate FEC, a frame length of 20ms and a base coding rate of 64 kilobits per second (Kbps) were selected. The base coding rate was equivalent to that used in Pulse Code Modulation (PCM). The redundancy coding rate was selected to be 25 Kbps. The total bit rate for an audio stream increased from 64 Kbps to 89 Kbps as a result of the redundancy coding rate that was added to the audio stream. The average times to encode and decode an audio packet are shown in tables 6.1 and 6.2 respectively.

Audio packet	Encoding time per packet (n = number of experiments)			
	n = 1	n = 2	n = 3	n = 4
8	5ms	5ms	6ms	7ms
16	11ms	8ms	9ms	10ms
32	15ms	14ms	17ms	19ms

Table 6.1 Average time taken to encode an audio packet

Four trials were conducted with three different sets of source packets: 8, 16 and 32. All four trials were conducted over the Computer Science departments network. Results indicate a fluctuation of average encoded times as a result of the load on the network.

Audio packet	Decoding time per packet (n = number of experiments)			
	n = 1	n = 2	n = 3	n = 4
8	11ms	9ms	12ms	14ms
16	21ms	16ms	14ms	20ms
32	24ms	23ms	29ms	21ms

Table 6.2 Average time taken to decode an audio packet

Much the same as with the encoding procedure, the decoding testing was conduced. Four trials with three different set of source packets revealed a fluctuations in average time taken to decode redundant data. This was due to the load of the network, which at times had a high and low of amounts of network traffic.

After the average times for both encoding and decoding were calculated, the average number of packets lost was calculated. Packet delay was measured against the threshold of 180ms. The first experiment conducted in this regard measured SIMBA's capabilities of sending audio messages through the bottleneck. Thus audio messages were sent through with no guarantee measurements put in place. To help aid in detecting lost packets, RTPMonitor was used (see Figure 5.3). Fifty messages were sent and the results obtained are shown in Figure 6.1:

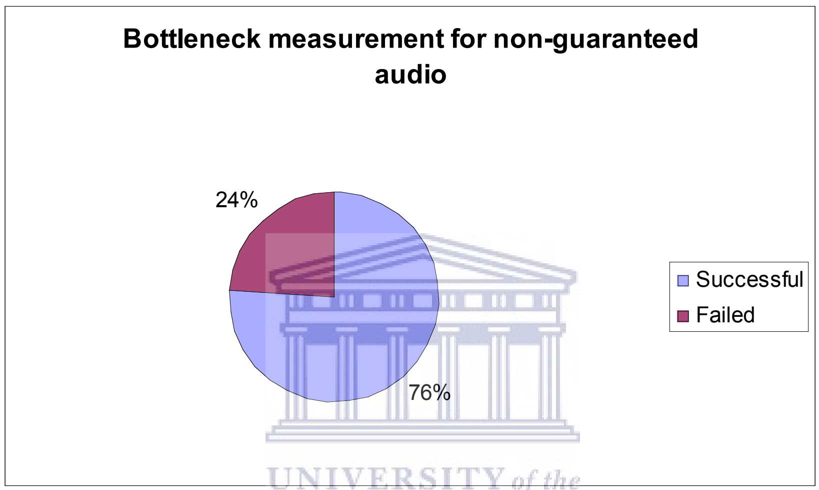

Figure 6.1 Bottleneck measurements for non-guaranteed audio

Even though audio guarantees were not provided, a substantial amount of audio data was still successfully sent. There was packet loss that occurred though, and this was due to the bottleneck link created and the amount of congestion on the network.

Figure 6.1 shows that the success rate for message delivery accounted for seventy six percent of all messages sent. However, when FEC was measured in terms of the threshold set in place, different results were obtained, as shown in Figure 6.2:

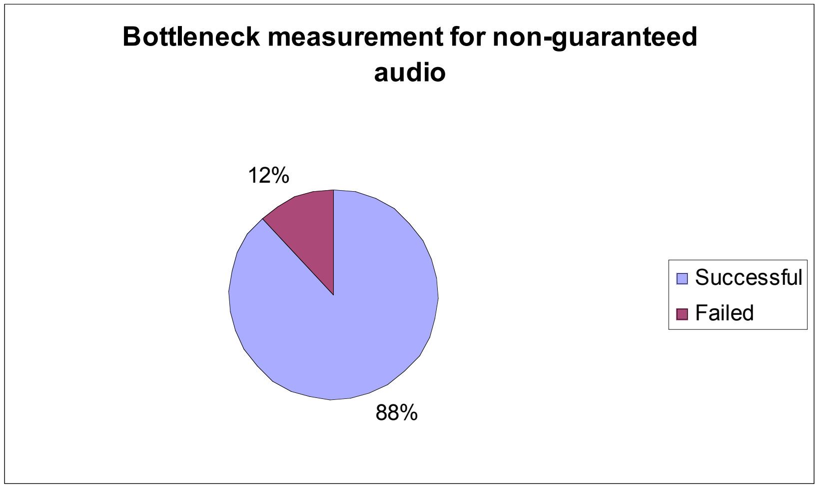

Figure 6.2 Bottleneck measurements for guaranteed audio

With FEC in place the success rate increased from 76% to 88%. Packet loss still occurred due to the amount of congestion on the network. This was, however, not a problem as the observed audio quality was still tolerable by a human ear (Miller and Licklider 1950; Wasem et. al 1988).

There was thus a twelve percent decrease in audio packet loss, a result that is still tolerant to the human ear (Miller and Licklider 1950; Wasem *et. al* 1988). When the observed audio quality was measured in terms of user perception it was found that even though a small amount of packet loss occurred, it did not affect the users ability to interpret what was said.

6.1.2 Asynchronous environment

To evaluate the asynchronous environment three separate tests were conducted: persistent logging of messages to the storage medium (database), acknowledgements sent from the NIMBA server to the Deaf client and acknowledgements sent from the Deaf client to the NIMBA server. When the persistent logging feature was tested, all messages sent for storage were persistently logged. Even when communication links were broken, messages were persistently logged. Thus, persistently logging to a database was successful and passed through the evaluation stage. The next feature that was tested was the ACK and NAK scheme. Recall from Chapter 4 that a NAK message is sent once a message was not successfully received and an ACK message is sent once a message is successfully received. Thus, to test the NAK scheme clients were requested to retrieve messages from a database. A request for message retrieval was sent by each client, requesting all messages logged to the client to be sent. Once a request for a message was successful, communication links were broken and the number of NAK messages were recorded. If the NAK scheme was successful, the amount of NAK messages for a particular client would be the same as the amount of messages logged in the database for that client. This procedure was measured and the results are shown in Table 6.3:

Number of messages sent	Number of NAK messages received	Average time taken per message sent (ms)
30	30	3ms
40	40	5ms
50	50	3ms

Table 6.3 NAK success rate

Three trials were conducted at the Computer Science department's network. Internet traffic fluctuated from high to small loads thus in the process influencing the average time taken to transport NAK messages.

The same procedure was used to test the ACK scheme. Clients were requested to receive messages destined to it, in the same manner as that of the NAK scheme. This time though, communication links remained steady and the amount of ACK messages were compared to that maintained in a database. A similar result was obtained to that observed in the NAK scheme. The results are shown in Table 6.4:

Number of messages sent	Number of ACK messages received	Average time taken per message sent (ms)
30	30	5ms
40	40	7ms
50	50	9ms

Table 6.4 ACK success rate

Just as with the NAK scheme, the ACK scheme was tested over the Computer Science department's network. With high to low variation of network traffic, the average time taken to send ACK messages was also affected. Thus, the average time fluctuated from 5ms when loads of traffic was small, to 9ms when loads of traffic increases.

6.1.3 Semi-synchronous environment

The semi-synchronous environment involved both synchronous and asynchronous test environments. Thus test procedures for both environments were used to indicate whether the implemented system was developed correctly. The results obtained were much the same as that obtained for separate test cases. The only difference with this test scenario is that messages were monitored from sender to receiver to test the overall hearing user's perception of the received audio quality. The results for an overall perception on ACK and NAK messages received are shown in Table 6.5:

Number of messages sent	Number of ACK messages received	Number of NAK messages received	Average time taken per message sent (ms)
50	50	50	7ms
60	60	60	10ms
40	40	40	5ms

Table 6.5 NAK/ACK success rate

This table represents the results obtained of all NAK and ACK messages constructed and received within the overall testing environment of NIMBA.

Thus, asynchronous as well as synchronous messages were monitored but more emphasis was made on the quality of the perceived audio message. From a developers view point, the quality of audio messages received via the bottleneck link created on the network was tolerable. This was measured according to a technical and social study conducted on human perceived audio quality over the Internet by Miller and Licklider (1950) and Wasem *et. al* (1988). Although these experiments were conducted some time ago they still hold true in today's day and age as proved by (Rizzo 1997; Perkins and Crowcroft 2000).

6.2 Social Evaluation

DCCT provided a good environment for the evaluation of NIMBA on a social stage. This is because of the role that DCCT plays within the Deaf community in the Western Province. As mentioned in chapter 1, DCCT was created to address the needs of Deaf people in the Western Cape. It was thus quite fitting that NIMBA was evaluated at DCCT since it shared a distinct commonality with DCCT: addressing the needs of Deaf people in the Western Cape. Thus, NIMBA was evaluated in terms of a social space where message delivery was measured in terms of a Deaf person's day to day communication. It is important to note that technical evaluation was not really important at DCCT but more the acceptance of NIMBA into a Deaf person's daily life. The next two subsections looks at the results obtained from trials and observations conducted at DCCT as well as the questions raised and advice obtained from interviews conducted with a small group of Deaf participants.

6.2.1 Ethnographic observation and reflection

Since ethnography applies to the relationship between practitioner and researcher (see section 3.1.4), a close relationship was obtained between three Deaf participants and the researcher. Within a space of twelve months the researcher was given an insight into the daily lives of Deaf people. This insight was mainly obtained from weekly interaction with three main Deaf participants but also from interaction with other Deaf participants during the literacy training program. Valuable information was obtained in terms of Deaf communication. This information gave a better understanding in terms of cell phone device usage among the Deaf and the popularity of SMS among the Deaf community. With ethnography, a detailed analysis of NIMBA could be conduced to determine if NIMBA actually addressed the needs of the Deaf people under study. An interesting finding showed that even though NIMBA provided a better form of communication than SMS, it could not currently substitute for the usage of SMS among the Deaf community. From interaction with the Deaf participants, it was learned that cell phone device usage is very popular because of its mobile capabilities and ease of use when it comes to SMS communication. Most Deaf people whether part of the literacy training program or not, were in possession of a cell phone to use SMS.

6.2.2 Questionnaire answers

As mentioned in Chapter 5, two sets of questions were asked to the three Deaf participants under study. However, for the post-trial questionnaires, only two Deaf participants were interviewed because of scheduling constraints. The researcher had to fit in with the Deaf participant's available schedules because of their responsibilities to DCCT. Thus, three Deaf participants were interviewed during the pre-trial questionnaire phase and only two Deaf participants were interviewed during the post-trial questionnaire phase. The questionnaires were used as guideline to interview the Deaf participants but the interview was really open to comments, ideas and suggestions. Both pre- and post-trial interviews were conducted with the aid of Sign language interpreter. Interviews were recorded using a voice recorder. Conversations were not transcribed but were rather listened to for any further information that might clarify answers given to questions asked using the questionnaires. Trials with NIMBA were conducted in much the same manner as that of SIMBA. From a user's perspective the only difference to SIMBA was that NIMBA provided a notification message that informed a Deaf participant of a successful message delivery. This delivery notification came to be important during the post trial questionnaire phase. In the post-trail phase, the aim was to find out how important message notification was in terms of delivery and response. The questionnaires used in the pre- and post-trial interview can be found in Appendices A and B.

Pre-trial Questionnaire

It is important to note that NIMBA was only introduced to the Deaf persons under study after the pre-trial questionnaire phase was completed. As mention in Chapter 5 the aim of the pre-trial questionnaire was to determine if unreliable communication with SIMBA affected a Deaf person's perception of message delivery. Guaranteed message delivery was asked in terms of SMS and SIMBA usage. The following results were obtained from the interviews. All three Deaf people were asked what they thought of reliability in terms of SMS's sent to destined parties. Two of the three had confidence that when the notification service on cell phone devices were enabled, all messages would be sent reliably to the intend destination. This result is shown in Figure 6.3.

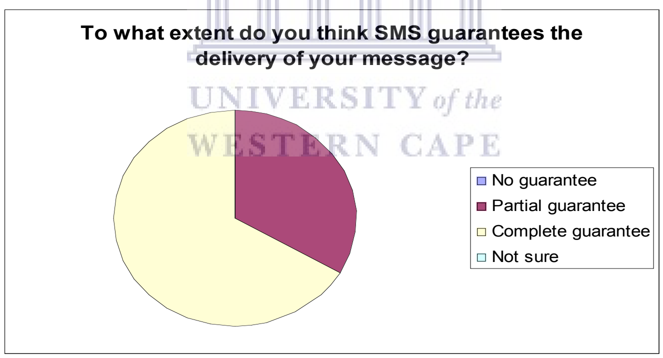

Figure 6.3 Deaf perception of SMS delivery

Even though SMS is perceived to be an unreliable form of data exchange, the majority of Deaf persons under study believed that all messages sent are delivered to their destination.

When the same question was asked in terms of SIMBA's capabilities to reliably send messages across a network, all three replied saying SIMBA would definitely deliver all messages sent. It almost seemed that all three Deaf persons really wanted SIMBA to deliver messages reliably. This became more apparent when all three Deaf participants answered, with an overjoyed, yes to the following question: “Are message guarantees an important factor for communication within SIMBA?”

What became clear is that SIMBA’s capability to provide a real time framework for communication was very important for a Deaf person’s day to day communication. The following question was posed: “How does communication with SIMBA compare to that of SMS?” Two of the Deaf participants said they don’t know and one said that SIMBA is better. When this question was posed, all three Deaf participants had something extra to add to the question. One believed that SMS is primarily a “Deaf” communication service that enabled Deaf to Deaf communication to occur seamlessly. However, the Deaf person also mentioned that SIMBA was very important for a Deaf person as it provided a relay operator that enabled Deaf people to communicate with hearing people. This, according to the Deaf person, is very important since Deaf to hearing communication via cell phones does not always occur seamlessly. A Deaf and hearing person’s grammar is completely different and this creates difficulties when communicating via SMS. Two of the Deaf people believed that SIMBA would rectify a Deaf person’s grammar before relaying the message to a hearing user. One of the two Deaf participants also made mention that patience is very important for a Deaf person using SIMBA. The delays that SIMBA imposed are a bit bigger than that of SMS, according to the Deaf person. There are various things that need to be prepared before a call can be placed. The relay operator needs to get into position and a hearing user needs to be available to chat. So, as long as a Deaf person is patient SIMBA is a good service to make use of. However, the other Deaf person made mention that cell phone technology is on the increase. There are video capabilities on cell phone devices that allow a Deaf person to sign a message and send this via Multimedia Messaging Service (MMS). According to the Deaf person under study this feature is very important to a Deaf person since it allows him/her to communicate in his/her first language. The overall picture that emerged is that both SIMBA and cell phones have very unique and beneficial technologies and that a combination of the two would give a more soluble solution to the Deaf person's day to day communication requirement.

One last question was posed: "If SIMBA fails to deliver SOME messages, would you still continue using it? Yes was the order of the day. All three agreed that a relay service which allows Deaf and hearing individuals to communicate is very important especially when hearing users find it difficult to understand a Deaf person's SMS.

Post-trial Questionnaire

As mentioned earlier, one of the Deaf persons involved in the pre-trial questionnaire was unavailable during the post-trial interview stage. This was, however, not a problem since valuable information was obtained from the other two Deaf participants during the post-trial interview stage. After the two Deaf participants were exposed to NIMBA, five questions were asked in terms of the reliability NIMBA provided.

The first question was asked in terms of the confidence rating when NIMBA informed the Deaf user that his/her message was sent. The one Deaf participant was confident and the other very confident that NIMBA actually delivered the message. However, questions were posed at the researcher. One Deaf person asked what would happen if a Deaf person sends a message to a hearing person but the hearing person does not pick up the phone. Would NIMBA then notify the Deaf person of such an event? According to the Deaf person, NIMBA provides a real time service and should give real time feedback to what is happening on the other side of a call that is placed. The other Deaf person mentioned that SMS sometimes imposes delay which in turn causes messages to be delivered late. This causes the Deaf person to be less confident in terms of believing the message notifier when using SMS. According to the Deaf participant, as long as the delay within NIMBA is not too long, communication confidence in NIMBA's abilities to deliver a message would be much higher than that of cell phones. The second question asked if message guarantees are important in emergency environments. Both Deaf people replied with a yes and one made mention that NIMBA would be used more than SMS in this regard.

The last question that was asked gave good feedback in terms of NIMBA and cell phone usage. The question was asked: “Knowing that NIMBA guarantees message delivery, would you prefer it over SMS?” One Deaf participant said yes and the other no. The one that said yes believed that real time communication is very important to the Deaf community. The Deaf person mentioned that with SMS one must wait before one knows if a message was read by the recipient. “With an SMS I don’t know, I will wait and wait and wait. I would prefer SIMBA because I could communicate directly”. The one that said “no” mentioned that both SIMBA and NIMBA had a big load of responsibilities to get the message across. These responsibilities included the readiness and availability of the relay operator as well as the awareness of SIMBA/NIMBA among both the Deaf and hearing communities. When a relay operator is available twenty four hours a day and a SIMBA/NIMBA client is available at a Deaf person’s house which has Internet connectivity, only then would SMS and SIMBA/NIMBA be balanced out. Thus, once again the overhead of NIMBA/SIMBA call setup became a bit of a problem for some Deaf people.

6.3 Summary

This chapter presented a discussion of the results obtained from experiments conducted in a laboratory as well as at DCCT. Results show that FEC and MOM are good solutions to guarantee the delivery of semi-synchronous messages sent from sender to receiver. However, ethnographic observations suggest that even though message guarantees in NIMBA were seen as crucial, the system itself could not currently substitute for the ever popular SMS on cell phones. The next chapter concludes the whole thesis and discusses the contribution the thesis made not just in the field of computer science but in the lives of the Deaf people under study.

Chapter 7 CONCLUSION AND FUTURE WORK

Most people are born with the physical ability to hear and speak. Each individual has his/her own way of understanding verbal and nonverbal meanings. Whether knowledge about a certain language is mundane or profound, we all start with basic communication skills. These skills are as simple as modeling behaviors based on the observation of other people's actions. "The field of communication focuses on how people use messages to generate meanings within and across various context, cultures, channels, and media" (U.S. Department of Education 2000).

Communication is one of the most valuable assets to man kind. Whether it is in the form of eye contact, hand written mail, electronic mail (e-mail) or the use of a telephone, it is used by everyone to express meaning and thought. It has been estimated that 75% of a person's day is dedicated to some form of communication. As a student at a tertiary institution, 69% of communication time is spent on speaking and listening. Seventeen percent of communication time is spent on reading and 14% on writing (Barker *et. al* 1980). Buckley (1992) described communication succinctly by saying: "We listen a book a day, we speak a book a week, read the equivalent of a book a month, and write the equivalent of a book a year".

So then, if communication is so very important, how important is communication in a semi-synchronous environment? Synchronous communication such as face to face or telephone conversation is guaranteed implicitly. Asynchronous communication such as SMS, e-mail, fax or any other transport media, only gives a guarantee once a reply has been received. For this thesis, a combination of synchronous and asynchronous media in a semi-synchronous environment was used as a framework to guarantee messages sent and received. The next four subsections looks at situating the results obtained from this thesis, technical conclusions made in terms of a communication in a semi-synchronous environment, limitations of the study conducted as well as a concise conclusion of the thesis.

7.1 Situating the results

Although this thesis addresses reliability issues pertaining to a semi-synchronous environment, the overall contribution of NIMBA was to realize the potential of a Deaf Telephony application in South Africa. As mentioned in Chapter 1, voice and video relaying in South Africa is simply not provided. NIMBA used researchers to act as relay operators for Deaf and hearing users. In order for NIMBA awareness to be realistic among the Deaf community, a full time relay operator needs to be provided. Chapter 6 showed that NIMBA provided great potential and benefits to a small group of Deaf participants at DCCT. There are however many other Deaf participants in South Africa that have not been exposed to technologies such as Deaf Telephony. In South Africa, deaf people (not just Deaf) constitute 10 percent of South Africa's population. Even though NIMBA was trialed with a Deaf community, deaf individuals could also find NIMBA to be an assistive software technology. Although NIMBA is based on technical concepts, it is also seen in terms of a social space. What is this social space? The ability to assist Deaf and hearing individuals to communicate reliably and without the aid of an interpreter. This thesis contributes to a Deaf person's social space by providing guarantees in terms of personal messages transported across an IP-based communication infrastructure. NIMBA has the potential to evolve into an assistive communication technology for emergency scenarios. In an emergency scenario with no other person in sight, who or what form of technology does a Deaf person turn to? Cell phones are mobile and easy to carry around with but the SMS service does not notify a Deaf person if a message was actually *read* by someone else. Instead, SMS only notifies a user that a message was delivered. Is there another option? This all depends on the technical infrastructure. SIMBA/NIMBA has been developed but to roll it out on a large scale, the major "players" such as Telkom and other service providers need to get involved.

7.2 Technical considerations

This thesis demonstrated that reliability can be brought to a semi-synchronous environment. FEC for synchronous and MOM for asynchronous communication were found to be complimentary solutions. Although the overall solution was brought about in terms of two separate transport media, synchronous and asynchronous, the thesis shows that both can be applied in a semi-synchronous environment to guarantee the delivery of a message from sender to receiver. Even though messages traverse through a combined asynchronous and synchronous path, it was still guaranteed for a Deaf Telephony application. Perhaps the best way to transport a message would be to have it sent across one medium such as a synchronous medium. This would in effect allow one of the solutions, e.g., FEC, to be employed for such a scheme. However, this is seen as future work and is discussed more in the last section.

7.3 Limitations of the study

Although valuable information was obtained as a result of ethnography applied to a small group of Deaf participants, a better understanding of Deaf communication could have been obtained if in-depth ethnography was applied to a bigger group of Deaf people. As mentioned in chapter 1, the deaf community constitutes 10 percent of South Africa's population of which 1.5 million Deaf people in South Africa use South African Sign Language. In this thesis, three Deaf participants were placed under study. In a statistical measurement, a bigger sample size would produce a better result. However, because of the limited amount of time given to complete this thesis, such a study could not be conducted. MOM provided a good solution for asynchronous communication but failed to the same for synchronous communication. This is because MOM is entirely based on an asynchronous store-and-forward architecture. Thus, such an architecture could not allow synchronous messages to be guaranteed in a point-to-point or even point-to-multipoint environment. In chapter 6, results were shown for FEC tested in an environment that produced a bottleneck on a network. Details pertaining to network congestion were out of the scope of this thesis. Once again, time was very crucial which allowed network congestion, a big topic in the field of computer science, to be left as future work. Throughout the thesis there was talk about the placement of Deaf Telephony in emergency scenarios.

Theoretically, NIMBA could be applied to an emergency situation but practically it could also provide challenges and difficulties since NIMBA was not tested in an emergency environment. The main aim of the thesis was to address reliability issues in terms of synchronous and asynchronous communication environments and not applying Deaf Telephony solutions to an emergency scenario. Thus, Deaf Telephony emergency environments are left as future work.

7.4 Future work

As mentioned in the previous subsection, a study of network congestion could be conducted in terms of NIMBA's message throughput and transport mediums. Even though NIMBA was theoretically described in terms of an emergency scenario, it could be practically implemented in such an environment and realized as a potential Deaf assistive service. Even though NIMBA provided a very good solution in terms of communication among Deaf and hearing people, it still could not really substitute for SMS on cell phone devices among the Deaf. This was due to the cell phone device's mobile capabilities, less time spent establishing communication, ease of use, decentralized form of communication and current popularity of SMS among the Deaf community. A very good solution would be to combine the functionality of a cell phone with the service provided by NIMBA. This would allow more Deaf people to be exposed to a Deaf Telephony service that provides reliability in terms of semi-synchronous message delivery. Currently, more and more services are being added to the cell phone package. This includes the sending of pictures, audio and video. For a Deaf person, signing to a video camera on a cell phone would be much easier than going to DCCT to make a synchronous call. Sign language, just like any other language, expects any two or more participants to be amenable to the rules of the language. Thus, if a Deaf person chooses a cell phone as communication device with sign language as the communication service, a relay operator needs to be introduced to allow a hearing person to be included in the conversation. Thus, the fundamentals of NIMBA could be extracted to provide reliability for a cell phone device's communication service. Currently, DCCT is a centralized environment that requires traveling to the centre to use NIMBA for communication. A better option would be to have the NIMBA service running on a cell phone device. However, just as NIMBA currently provides reliability in a semi-synchronous environment, so should the service running on a mobile device.

A further enhanced solution would be to have message use for only one transport medium. If everything, including text, were synchronous, then FEC could be applied to provide reliability for message sent from sender to receiver. This would allow less overhead to occur as a result of switching from asynchronous to synchronous mediums and vice versa. Alternatively, a purely asynchronous solution could be explored that employed only MOM-type techniques.

References

[1] Aarons D. and Akach P. (2002). Inclusion and the Deaf child in South African education, *Perspective in Education,* Pretoria, South Africa, 20(1), 153- 170.

[2] Agboola I.O. and Lee A.C. (2000). Computer and information technology access for deaf individuals in developed and developing countries, *Journal of Deaf Studies and Deaf Education,* 5(3), 286 – 289.

[3] Asmal K. (2001). Plenary address to the *International Conference on Deaf Education: Building Bridges To Literacy,* Johannesburg, South Africa.

[4] Balter R. (2004). A ScalAgent Distributed Technologies White Paper. JORAM: The Open Source Enterprise Service Bus, *White paper*.

[5] Barker L., Edwards R., Gaines C., Gladney K. and Holley F. (1980). An investigation of proportional time spent in various communication activities by college students, *Journal of Applied Communication Research*, 8, 101-109.

[6] Bellisard L., Felliot D., Freyssinet A., Lacourte S. and Tachkler N. (2000). JORAM: Java Open Reliable Asynchronous Messaging, http://joram.objectweb.org/current/doc/Joram-4.2-en.pdf.

[7] Bhargava A., Kurose J., Towsley D. and VanLeemput G. (1988). Performance Comparison of Error Control Schemes in High-Speed Computer Communication Networks, *IEEE Journal on Selected Areas in Communications,* Los Alamitos, CA, USA, 6(9), 1565-1575.

[8] Bhola S., Strom R., Bagchi S., Zhao Y. and Auerbach J. (2002). Exactly-once Delivery in a Content-based Publish-Subscribe system, *Proc. of the 2002 IEEE International Conference on Dependable Systems and Networks*, 7 – 16.

[9] Bhola S., Zhao Y. and Auerbach J. (2003). Scalably Supporting Durable Subscriptions in a Publish/Subscribe System, *Proc. of the IEEE International Conference on Dependable Systems and Networks,* San Francisco, CA, 57 – 66.

[10] Biersack E.W. (1992). A Simulation Study of Forward Error Correction in ATM Networks, *ACM SIGCOMM Computer Communication Review,* New York, USA, 22(1), 36 – 47.

[11] Biersack E.W., Cotton C.J., Feldmeier D.C., McAuley A.J. and Sincoskie D.(1992). Gigabit networking research at bellcore, *Proc. of the IEEE Network Conference*, Bellcore, Aurora, 6(2), 42-48.

[12] Birman K.P., van Renesse R. and Vogels W. (2001). Spinglass: Secure and Scalable Communication Tools for Mission-Critical Computing, *DARPA International Survivability Conference and Exposition II DISCEX*, 2, 85 – 99.

[13] Blake E. and Tucker W.D. (2004). Bridging Communications Across the Digital Divide, *Proc. of Social Intelligence Design (SID)*, Enschede, Netherlands, 29 – 38.

[14] Blake E.H. and Tucker W.D. (2005). User Interfaces for Communication Bridges Across the Digital Divide, *AI and Society*, Spriger – Verlag, 19(3).

[15] Bolot J.C. and Garcia A.V. (1996). Control mechanisms for packet audio in the Internet, *Proc. of the IEEE Infocom*, San Francisco, CA, 232 – 239.

[16] Buckley M.H. (1992). Focus on research: We listen a book a day; speak a book a week: Learning from Walter Lorban. *Language Arts*, 69, 623.

[17] Budde R., Kautz K., Kuhlenkamp K. and Zullighoven H. (1992). What is prototyping?, *Information Technology and people*, 6(2), 89 – 95.

[18] Campbell B., Rosenberg J., Schulzrinne H., Huitema C. and Gurle D. (2002). Session Initiation Protocol (SIP) Extension for Instant Messaging, *RFC 3428*.

[19] Carzaniga A. and Wolf A.L. (2000). Content-Based Addressing and Routing: A General Model, *Technical Report CU-CS-902-00*, Computer Science Department, University of Colorado.

[20] Chen Y.L. and Chen B.S. (1997). Model-based multirate representation of speech signals and its application to recovery of missing speech packets, *IEEE Transaction on Speech and Audio Processing*, 15(3), 220 – 231.

[21] Clark D.D., Lambert M.L. and Zhang L. (1987). NETBLT: A high throughput transport protocol, *Proc. of the ACM SIGCOMM' 87*, Stowe, Vermont, USA, 17(5), 353-359.

[22] Cox S., Lincoln M., Tryggvason J., Nakisa M., Wells M., Tutt M. and Abbot S. (2002). Tessa, a system to aid communication with Deaf people, *Proc. of the fifth international ACM conference on Assistive technologies*, Edinburgh, Scotland, 205 – 212.

[23] DaSilva L., Petr D. and Frost V. (1989). A class-oriented replacement technique for lost speech packets, *Proc. of the 8^{th} Annual joint Conference of the IEEE Computer and Communications Societies. Technology: Emerging or Converging, INFOCOM*, 3, 1098 – 1105.

[24] Davies L.J. and Nielsen S. (1992). An ethnographic study of configuration management and documentation practices in an information technology centre, *The impact of Computer-supported technology on Information Systems Development*, 179 – 192.

[25] Dempsey B.J., Liebeherr J. and Weaver A.C. (1993). A new error control scheme for packetized voice over high-speed local area networks, *Proc. of the 18th Conference on Local Computer Networks*, 91 – 100.

[26] Dempsey B.J., Liebeherr J. and Weaver A.C. (1994). On retransmission-based error control for continuous media traffic in packet-switching networks, *Computer Networks and ISDN Systems,* Amsterdam, The Netherlands, 28(5), 719 - 736.

[27] Dempsey B.J. and Zhang Y. (1996). Destination buffering for low-bandwidth audio transmission using redundancy-based error control, *Proc. of the 21st IEEE Local Computer Networks Conference*, Minneapolis, Minnesota, USA, 345 – 355.

[28] Erdol N., Castelluccia C. and Zilouchian A. (1993). Recovery of missing speech packets using the short-time energy and zero-crossing measurement, *IEEE Transaction on Speech and Audio Processing*, 1(3), 295 – 303.

[29] Erramilli A. and Singh R.P. (1987). A Reliable and Efficient Multicast Protocol for Broadband Broadcast Networks, *Proc. of the ACM Conference SIGCOMM*, 343-352.

[30] Glaser M. (2000). A Field Trial and Evaluation of Telkom's Teldem Terminal in a Deaf Community in the Western Cape, *Proc. of the South African Telecommunication Networks and Applications Conference (SATNAC),* Stellenbosch, South Africa.

[31] Glaser M. and Aarons D. (2002). A Deaf adult literacy collective, *Stellenbosch Papers in Linguistics,* 34, 1- 18.

[32] Glaser M. and Tucker W.D. (2004). 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 2004*, Grenada, Spain ,(CD-ROM publication).

[33] Glaser M., Young P. and Porteous L. (2005). An Information and Communication Technology (ICT) Needs Analysis and Training Program for Deaf People, *Proc. of the 20th International Congress on the Education of the Deaf, ICED 2005*, Maastricht, Netherlands, 231.

[34] Goodman D.J., Lockhart G.B., Wasem O.J. and Wong W.C. (1986). Waveform substitution techniques for recovering missing speech segment in packet voice communications, *IEEE Transaction on Acoustics, Speech and Signal Processing*, 34(6), 1440 – 1448.

[35] Gordan V.S. and Bieman J.M. (1994). Rapid prototyping: Lessons learned, *IEEE Software*, 12(1), 85 – 95.

[36] Grinter R.E. and Palen L. (2002). Instant Messaging in teen life, *Proc. of the 2002 ACM conference on Computer supported cooperative work*, New Orleans, Louisiana, USA, 21 – 30.

[37] Grubler J.G. and Strawczynski L. (1985). Subjective effects of variable delay and clipping in dynamically managed voice systems, *IEEE Transaction on Communications*, 33(8), 1440 – 1448.

[38] Hardman V., Sasse M.A., Handley M. and Watson A. (1995). Reliable audio for use over the Internet, *Proc. of Internet Society's International Networking Conference*, Oahu, Hawaii, 171 – 178.

[39] Harvey L., Myers M. (1995). Scholarship and practice: the contribution of ethnographic research methods to bridging the gap, *Information, Technology and people*, 8(3), 13 -27.

[40] Hilbert D.M. and Redmiles D.F (1998). Agents for collecting application usage data over the Internet, *Proc. of the second international conference on Autonomous agents*, Minneapolis, Minnesota, 149 – 156.

[41] Hughes J.A., Randall D. and Shapiro D. (1992). Faltering from ethnography to design, *Proc. of the 1992 ACM Conference on Computer-supported Co-operative Work: Sharing Perspective (CSCW)*, Toronto, Canada, 115 – 123.

[42] Julius E.P. and Tucker W.D. (2005). Guaranteed delivery of semi-synchronous IP-based communication, *Proc. of the South African Telecommunications Networks & Applications Conference* (*SATNAC)*, Drakensberg, South Africa, 1, 407 – 412.

[43] Jung D., Paek K. and Kim T. (1999). Design of MOBILE MOM: Message Oriented Middleware Service for Mobile Computing, *Proc. of the international conference on parallel processing workshops*, 434.

[44] Kitchenham B. and Pfleeger S. L. (2002). Principles of Survey Research Part4: Questionnaires Evaluation, *ACM SIGSOFT Software Engineering Notes*, 27(3), 20 - 23.

[45] Kiyaga N.B. and Moores D.F. (2003). Deafness in Sub-Saharan Africa, *American Annals of the Deaf*, 148(1), 18-24.

[46] Lara-Barron M.M. and Lockhart G.B. (1992). Packet-based embedded encoding for transmission of low-bit-rate-encoded speech in packet networks, *Proc. of the IEE International Conference*, 139(5), 482- 487.

[47] Lewis J., Tucker W.D. and Blake E (2002). SoftBridge: An Architecture for Building IP-based Bridges over the Digital Divide. *Proc. of the South African Telecommunications Networks & Applications Conference* (*SATNAC)*, Drakensberg, South Africa, 383 – 386.

[48] Lewis J., Tucker W.D. and Blake E. (2003). SoftBridge: A Multi-modal Instant Messaging Bridging System, *Proc. of the South African Telecommunications Networks and Applications Conference (SATNAC*), George, South Africa, 1, 235 – 240.

[49] Lockhart G.B. and Lara-Barron M.M. (1991). Implementation of packet-based encoding schemes for speech transmission, *Sixth International Conference on Digital Processing of Signals in Communications*, 326 – 330.

[50] Lombard S. (2005). Person Communication, June, Cape Town, South Africa.

[51] Menascé D.A. (2005). MOM vs. RPC: Communication Models for Distributed Applications, *IEEE Internet Computing*, 9(2), 90 – 93.

[52] Miller G.A. and Licklider J.C.R. (1950). The intelligibility of interrupted speech, *Journal of the Acoustical Society of America*, 22(2), 167 – 173.

[53] Nelson G. and Underschultz G. (2003). Commission on Technology, *World Federation of the Deaf News*, 16(3), 21.

[54] O'Neill J. and Martin D. (2003). Text Chat in Action, *Proc GROUP '03*, Sanibel Island, Florida, 40 – 49.

[55] Orlikowski W.J. and Robey D. (1991). Information technology and the structuring of organizations, *Information Systems Research*, 2(2), 143 – 169.

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

[57] Pallickara S. and Fox G. (2004). A Scheme for reliable delivery of events in distributed middleware systems, *Proc. of the IEEE International Conference on Autonomic Computing*, 328 - 329.

[58] Pejhan S. and Schwartz M. (1996). Error control using retransmission schemes in multicast transport protocols for real-time media, *IEEE/ACM Transaction on Networking*, 4(3), 413 – 427.

[59] Penton J., Tucker W.D. and Glaser M. (2002). Telgo323: An H.323 Bridge for Deaf Telephony, *Proc. of the South African Telecommunications Networks & Applications Conference* (*SATNAC)*, Drakensberg, South Africa, 309 – 313.

[60] Perkins C. and Crowcroft J. (2000). Effects of Interleaving on RTP Header Compression, *Proc. of the IEEE INFOCOM*, *The conference on computer communications,* Tel Aviv, Israel, 1, 111 – 117.

[61] Perkins C., Hodson O. and Hardman V. (1998). A Survey of Packet-Loss Recovery Techniques for Streaming Audio, *IEEE Network,* 12(5), 40 – 48.

[62] Perkins C.S., Kouvelas I., Hodson O., Hardman V., Handley M., Bolot J.C., Vega-Garcia A. and Fosse-Parisis S. (1997). RTP Payload for redundant audio data, *RFC2198*.

[63] Phasiwe K. (2004). Telkom Treads On Cellular Turf With New SMS Product, *Business day*, web publication, http://www.empowerment.co.za/befullmedia.asp?mediaID=2010

[64] Podolsky M., Romer C. and McCanne S. (1998). Simulation of FEC-based error control for packet audio on the Internet, *Proc. of IEEE INFOCOM,* San Francisco, CA, 48-52.

[65] Quema V. (2002). Application-Driven Customization of Message-Oriented Middleware, *Proc. of the workshop on software infrastructures for component-based applications on consumer devices.*

[66] Ramsey J.L. (1970). Realization of optimum interleavers, *IEEE Transaction on Information Theory,* 16(3), 338 – 345.

[67] Rizzo L. (1997). Effective erasure codes for reliable computer communication protocols, *ACM SIGCOMM Computer Communication Review*, 27(2), 24 – 36.

[68] Rosenberg J., Schulzrinne H., Camarillo G., Johnston A., Peterson J., Sparks R., Handley M. and Schooler E. (2002). SIP: Session Initiation Protocol, *Internet Engineering Task Force (IETF) Request for Comments (RFC) 3261.*

[69] Saito K., Fujiya H., Kohmura H. and Kanno S. (1989). Voice packet communication system for private networks, *Proc. of the IEEE Global*

Telecommunications Conference and Exhibition. Communications Technology for the 1990s and Beyond. GLOBECOM '89, 3, 1874 – 1878.

[70] Sanneck H., Stenger A., Bem Younes K. and Girod B. (1996). A new technique for audio packet loss concealment, *Proc. of the IEEE Global Internet,* London, England, 48 – 52.

[71] Sanneck H. (1998). Concealment of lost speech packets using adaptive packetization, *Proc. of the IEEE International Conference on Multimedia Computing and Systems,* 140 – 149.

[72] Schulzrinne H. (1996). RTP Profile for Audio and Video Conferences with Minimal Control, *RFC 1890*, http://www.ietf.org/rfc/rfc1890.txt

[73] Schulzrinne H., Casner S., Frederick R. and Jacobson V. (1994). RTP: a transport protocol for real-time applications, *Working Group Draft*, *IETF AVT.*

[74] Shacham N. (1989). Packet recovery and error correction in high-speed wide-area networks, *Proc. of the IEEE Military Communication Conference*, 2, 551 – 557.

[75] Shacham N. and McKenney P. (1990). Packet recovery in high-speed networks using coding and buffer management, *Proc. of the IEEE INFOCOM*, San Francisco, CA, 1, 124 – 131.

[76] Shah A., Atungsiri S., Kondoz A. and Evans B. (1996). Lossy multiplexing of low bit rate speech in thin route telephony, *Electronics Letters*, 32(2), 95.

[77] Sommerville I. (2004). Software Engineering seventh edition, *Pearson Education Limited*, *Addison-Wesley publishers Limited,* ISBN 0-321-21026-3.

[78] Stone J. and Merrion S. (2004). Instant Messaging or Instant Headache?, *ACM Queue*, 2(2), 72 – 80.

[79] Sun T. and Tucker W.D. (2004). A SoftBridge with Carrier Grade Reliability using JAIN SLEE, *Proc. of the South African Telecommunications Networks and Applications Conference (SATNAC*), Stellenbosch, South Africa, 2, 251-252.

[80] Suzuki J. and Taka. M. (1989). Missing packet recover techniques for low-bit-rate coded speech, *IEEE Journal on Selected Areas in Communications*, 7(5), 707 – 717.

[81] Tang J. (1991). Evaluation of double sided periodic substitution (DSPS) method for recovering missing speech in packet voice communications, *Tenth Annual International Phoenix Conference on Computers and Communications*, 454 – 458.

[82] Tate G. and Verner J. (1990). Case study of risk management, incremental development and evolutionary prototyping, *Information and Software Technology*, 42(4), 207 – 214.

[83] Tran P., Greenfield P. and Gortan I. (2002). Behavior and performance of Message-Oriented Middleware systems, *Proc. of the 22nd IEEE international conference on distributed computing systems workshops (ICDCSW'02)*, Vienna, 645 – 654.

[84] Tucker R.C.F. and Flood J.E. (1985). Optimizing the performance of packet-switch speech, *Proc. of the IEEE Conference on Digital Processing of Signal in Communications,* Loughborough University, 227 – 234.

[85] Tucker W.D., Blake E.H. and Glaser M. (2003). Building Bridges for Deaf Telephony in South Africa: A Community-centred Approach, *IFIP Working Group 9.4 Newsletter,* 13(2).

[86] Tucker W.D., Glaser M. and Lewis J. (2003). SoftBridge in Action: The First Deaf Telephony Pilot, *Proc. of the South African Telecommunications Networks and Application (SATNAC)*, George, South Africa, 2, 293 – 294.

[87] Tucker W.D., Blake E.H. and Marsden G. (2004). Macro Delay and Multi-modal Semi-synchronous IP-Based Communication Bridges, *Proc. of the South African Telecommunications Networks and Applications Conference (SATNAC)*, Stellenbosch, South Africa, 2, 261 – 262.

[88] U.S. Department of Education (2000), "Association for communication administration", *Summer conference on defining the field of communication.*

[89] Valenzuela R.A. and Animalu C.N. (1989). A new voice-packet reconstruction technique, *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2, 1334 – 1336.

[90] Wah B.W. and Lin D. (1999). Transformation-Based Reconstruction For Real-Time Voice Transmissions Over the Internet, *IEEE Transaction on Multimedia*, Ottawa, Canada, 1(4), 342 – 351.

[91] Wah B.W., Su X. and Lin D. (2000). A Survey of Error-Concealment Schemes for Real-Time Audio and Video Transmissions over the Internet, *Proc. of the IEEE International Symposium on Multimedia Software Engineering*, 17 – 24**.**

[92] Wasem O.J., Goodman D.J., Dvordak C.A. and Page H.G. (1988). The effect of waveform substitution on the quality of PCM packet communications, *IEEE Transaction on Acoustics, Speech and Signal Processing*, 36(3), 342 – 348.

[93] Weiderman N.H., Habermann A.N., Borger M.W. and Klein M.H. (1987). A methodology for evaluating environments, *Proc. of the second ACM SIGSOFT/SIGPLAN software engineering symposium on practical software development environments,* Palo Alto, California, 22, 199 – 207.

[94] Yajnik M., Kurose J. and Towsley D. (1996). Packet loss correlation in the Mbone multicast network, *Proc. of the IEEE Global Internet Conference*, London, England, 94 – 99.

[95] Yong M. (1992). Study of voice packet reconstruction methods applied to CELP speech coding, *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2, 125 – 128.

[96] Zulu T., Glaser M. and Le Roux K. (2005). A preliminary investigation into Text-To-Speech (TTS) software in a semi-automated relay service for deaf telephony, *Unpublished Honours*, Department of Health and Rehabilitation, Division of Communication Sciences and Disorders, University of Cape Town, Cape Town, South Africa.

Appendix A Pre-trial Questionnaire

1. To what extent do you think SMS guarantees the delivery of your message?
    a. No guarantee
    b. Partial guarantee
    c. Complete guarantee
    d. Not sure

2. How does communication with SIMBA compare to that of SMS?
    a. Similar
    b. Worse
    c. Better
    d. Don't know

3. To what extent do you think SIMBA guarantees the delivery of your message?
    a. No guarantee
    b. Partial guarantee
    c. Complete guarantee
    d. Not sure

4. Are message guarantees an important factor for communication within SIMBA?
    a. Yes
    b. No
    c. Maybe
    d. Not sure

5. If SIMBA fails to deliver SOME messages, would you still continue using it?
    a. Maybe
    b. Yes
    c. No
    d. Not sure

Appendix B Post-trial Questionnaire

1. How confident were you that NIMBA delivered your message after receiving an acknowledgement from it?
    a. Confident
    b. Very confident
    c. Not confident
    d. Not sure

2. In an emergency scenario, such as phoning the police, do you think that a system which supports message guarantees would be beneficial?
    a. Yes
    b. No
    c. Maybe
    d. Not sure

3. Do you think NIMBA would be used more often because of its ability to guarantee messages?
    a. Yes
    b. No
    c. Maybe
    d. Not sure

4. Knowing that NIMBA guarantees message delivery, would you prefer it over SMS?
    a. No
    b. Yes
    c. Maybe
    d. Not sure