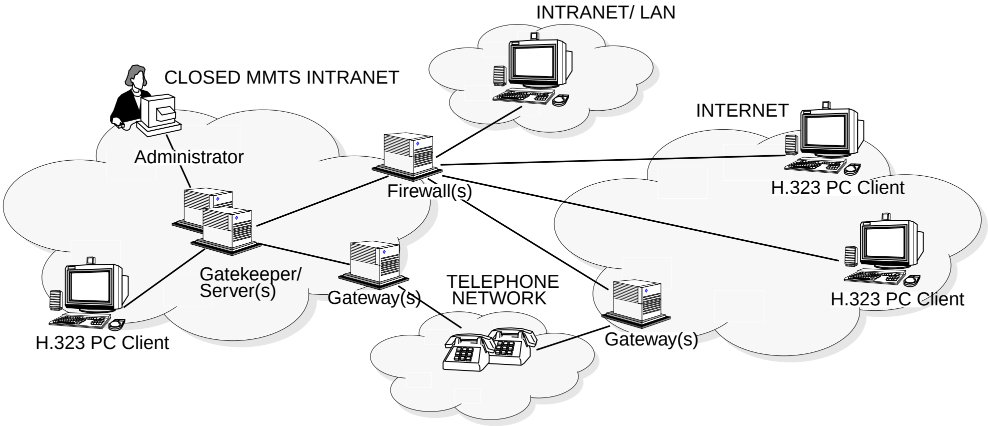
IMiS - Ericsson

Norsk Regnesentral
ANVENDT DATAFORSKNING
Norwegian Computing Center/Applied Research and Development

RAPPORT / REPORT



Report nr. 926

Peter D. Holmes
Lars Aarhus
Jan-Roger Sandbakken
Anders Frøyhaug
Espen Gjøstøl
Børge Nilsen
Morten Haavaldsen

Oslo
May 1998

Norsk Regnesentral
ANVENDT DATAFORSKNING

Rapport/Report

Tittel/Title:
IMiS - Ericsson

Dato/Date: May
År/Year: 1998
ISBN: 82-539-0446-0
Publikasjonsnr/
Publication no: 926

Forfatter/Authors:
Peter D. Holmes, Lars Aarhus, Jan-Roger Sandbakken, Anders Frøyhaug, Espen Gjøstøl, Børge Nilsen, Morten Haavaldsen

Sammendrag/Abstract:

IMiS Ericsson is a part of Ericsson's applied research efforts as defined by the Business Line for Internet Applications at ETO. The IMiS Ericsson project is an applied research project prioritized by Ericsson in order to monitor and have impact on the standardization efforts being made, and also ensure the ability to make system architectural choices based on knowledge of evolving technology and market trends.

The project has its primary areas of focus within the disciplines of multimedia, communication and security. The technologies in most sharp focus are:

- Ericsson's Multi-Media Telephone System (**MMTS**), and
- the ITU-T Recommendation H.323 (for packet-based multimedia communications systems).

With this focus, the project has consisted of two main parts: one on MMTS and Security, the other on the MMTS and Supplementary Services. The common denominator is openness and component technology, thus complying to and acknowledging the market requirements and trends in the Internet community.

Emneord/Keywords: H.323, Gatekeeper, Multi-Media Telephone System, Voice-on-the-Net, IP phone, supplementary services, desktop conferencing, multimedia messaging, communication support, H.235, security

Målgruppe/Target group: user, network and telecommunication organizations, research institutions

Tilgjengelighet/Availability: Open

Prosjektdata/Project data: Funded 40% by NFR and 60% Ericsson AS, Norway

Prosjektnr/Project no: 28008 / 28018

Antall sider/No of pages: 92

Norsk Regnesentral/Norwegian Computing Center
Gaustadalléen 23, Postboks 114 Blindern, N-0314 Oslo, Norway
Telefon (+47) 22 85 25 00, telefax (+47) 22 69 76 60

IMiS - Ericsson

Peter D. Holmes
Lars Aarhus
Jan-Roger Sandbakken
Anders Frøyhaug
Espen Gjøstøl
Børge Nilsen
Morten Haavaldsen

Norsk Regnesentral
May 1998

Table of Contents

Chapter 1

Introduction 1

1.1 Background 1

Ericsson 1

Norsk Regnesentral 1

IMiS 1

IMiS Kernel 2

IMiS Ericsson 2

IMiS-Veritas 2

Market Situation 3

1.2 Project Information 3

Goals and objectives 3

1.3 Project Strategy 4

Teams and Themes 4

MMTS and Security 4

MMTS and Supplementary Services 4

Enabling Mechanisms 5

1.4 Project Achievements 5

Achievement of objectives 5

Connections to other projects 6

Relation to the MMTS Project 6

Relation to IMiS - Veritas 6

Relation to IMiS - Kernel 6

Relation to the ENNCE Project 6

1.5 Structure of the report 7

Chapter 2

Technical Context 9

2.1 The Public Intranet 9

2.2 Service Network 10

2.3 The MMTS Platform 10

2.4 H.323’s Gatekeeper Entity 11

Chapter 3

Multimedia Telephone System (MMTS) 13

3.1 General Overview 13
Phone-to-Phone 13
PC-to-Phone 13
PC-to-PC 14

3.2 Functional Overview 14
MMTS User Functions 14
MMTS Provider Functions 15
Support Functions 15
Communication Functions 16

Chapter 4

Assessment of MMTS Threat Scenarios 19

4.1 General 19

4.2 Basic Preconditions 19

4.3 System Description 20
Overall Description 20
Multimedia Components 22
Protocol Overview 22
Client Logins 23
Call Set-up 24
A Short Description of H.235 28

4.4 Threats 29
Classes of Threats 29
Improper identification or authentication 29
Unauthorized access 29
Inadequate traceability 30
Unauthorized modification or insertion 30
Unauthorized disclosure 30
Unauthorized reduction of availability / QoS 30
Other threats 30
Sources of Threats 31

4.5 Case Studies 31
Terminal-to-Terminal Communication 31
Multipoint Communication 32
Gateway Communication 33
Firewall Communication 34
Other use cases 35

4.6 General Assessment 36

Chapter 5

Intelligent Answering Service (IAS) 39

5.1 Overview 39

5.2 Use and Operation of IAS 41

The life of Bill 42

Bill’s IAS configuration 43

Effects of the configuration 43

5.3 IAS Configuration Characteristics 44

5.4 Other Issues 45

Chapter 6

User Interface Specification for Intelligent Answering Service . . 47

6.1 Introduction 47

6.2 Service Interface Framework 48

6.3 Design Premises 49

6.4 IAS’ Graphical Interface 51

6.5 Functionality: system services 53

Configuration 53

Activation / deactivation 56

Statistics. 56

6.6 Implementation 56

Chapter 7

Architectural Approach for an H.323 MMTS Supplementary Service Execution Environment 57

7.1 Introduction 57

7.2 C-Layer 58

Points of Invocation 59

The MSSA. 61

7.3 Supplementary Service Independent Protocol 61

7.4 M/C-layer 62

7.5 Service Logic and Data 63

7.6 Open Issues 63

Chapter 8

QoS Support and Adaptation 65

8.1 Adaptation: a General Description 65

8.2 Assessing Approaches to Adaptation 66

Stringent resource reservation 66

User-controlled adaptation 67

Strict application-control 67

8.3 QoS Support through Hybrid Forms of Adaptation 68

Chapter 9

QoS Architectures and a Generalized QoS Framework 69

9.1 Motivation for QoS Architectures 69

9.2 Elements of a Generalized QoS Framework 70

QoS Principles 70

QoS Specification 71

QoS Mechanisms 73

QoS Provision Mechanisms 73

QoS Control Mechanisms 73

QoS Management Mechanisms 75

9.3 Focus of the Generalized QoS Framework 76

9.4 Resource- vs. Function-Oriented QoS Support 76

Chapter 10

Focal Areas for Future Work 77

10.1 Exploiting Possibilities for Synergy 77

10.2 Characterization of IMiS-Ericsson II 79

Suggested goals and objectives 79

Rationale 79

References 81

Appendix A

List of Seminars and Talks 85

Appendix B

Input drafted for the H.235 standardization effort 89

Chapter 1

Introduction

1.1 Background

1.1.1 Ericsson

Ericsson Norway (**ETO**) has a corporate responsibility for applications in the Internet domain. This includes everything from development till industrialization and marketing, as well as research activities prior to projects. In an area with rapid change in technology and pre-requisites for successful project conclusions, deep knowledge in key technological areas is of the utmost importance.

Telephony over IP is considered an important area close to traditional core business for Ericsson. ETO has a corporate responsibility for the *Multimedia Telephony System* (**MMTS**). MMTS enables voice over IP according to the ITU H.323 suite of specifications [12] and data collaboration according to T.120.

1.1.2 Norsk Regnesentral

For a number of years, Norsk Regnesentral's (**NR**) departments for information technology have been carrying out applied research in the areas of object-orientation, data communication, organizational development, distributed systems, interactive media and security. Since the inception of WWW technology and its consequent impact upon the use and exploitation of the Internet, NR has participated in an ever-increasing number of projects concerning state-of-the-art use of media streaming, security and distributed system technologies.

1.1.3 IMiS

IMiS -- **I**nfrastructure for **M**ultimedia Applications **i**n **S**eamless Net --is a long-term, applied research area supported by NFR and industry. Some of the actors involved in its inception include UNINETT, NR and Sintef Tele and Data, based upon work these parties performed within the IMiS Pre-project [1] and the IMiS Pilot Project [2]. Other actors involved include academic environments (IFI UiO, USIT), as well as industry- and user-environments (DnV and Ericsson).

IMiS functions as an umbrella for several different applied research projects. These projects are based upon cooperative efforts between R&D institutes and university environments, as well as R&D institutes, industrial and user organizations. Projects underway within IMiS include:

- IMiS Kernel: Experimental platform for infrastructure in seamless net;
- IMiS-Ericsson: API for the next generation of multimedia services;
- IMiS-Veritas: Mobile work-place and “Assistance upon demand”.

1.1.3.1 IMiS Kernel

The goal of the IMiS Kernel project is to establish a national lab environment for experimentation with multimedia services in a seamless network. The aim of the environment is to support research and higher education within Norway. Within the project, work will be performed concerning research and development of the lab technology itself, as well the use of this technology within education and research. The current actors within IMiS Kernel are Ericsson, NR, UNINETT and IFI UiO.

1.1.3.2 IMiS Ericsson

IMiS Ericsson is a part of Ericsson’s applied research efforts as defined by the Business Line for Internet Applications at ETO. The IMiS Ericsson project is an applied research project prioritized by Ericsson in order to monitor and have impact on the standardization efforts being made, and also ensure the ability to make system architectural choices based on knowledge of evolving technology and market trends.

The project consists of two main parts: one on MMTS and Security, the other on the MMTS and Supplementary Services (see section 1.3.1). The common denominator is openness and component technology, thus complying to and acknowledging the market requirements and trends in the Internet community.

1.1.3.3 IMiS-Veritas

The objective of the IMiS-Veritas project is to investigate how the possibilities inherent in information networks can be exploited for those working for Det Norske Veritas (DnV). The focus is upon preparing the way for seamless access to IT support for its mobile workforce, as well as supporting workers which require assistance during some work activity. The subgoals in the project include:

- identification of the need for seamless access to IT support
- investigate possible new forms of work for mobile workers
- investigate possible new forms of work involving “assistance-on-demand”
- investigate how mobile workers shall gain access to DnV’s intranet
- develop a plug-and-play solution for network connection
- adjust/modify applications to different net solutions
- develop security solutions and security strategies for network connections for mobile workers.

1.1.4 Market Situation

The Internet market situation is of a character similar to chaos in terms of changing technology, emerging markets and de-regulation of traditional telecommunications markets.

For quite some time, Internet technology has been used for electronic mail and news. In recent years, this technology has also been used for static documentation retrieval in terms of the World Wide Web (WWW).

One clear trend in the market is that applications and services offered to the end-users are based upon Internet technology and realized as servers attached to the Internet. Thus the Internet is evolving into a service platform for a myriad of applications.

Ericsson acknowledges Internet technology's strengths and possibilities, but wishes to further enhance the usability of IP-based services through improving the infrastructure upon which they are based. Ericsson shall build new functions into the network, to further support and simplify applications. For this reason, changes in the network infrastructure must accord to evolving application requirements.

1.2 Project Information

The IMiS-Ericsson project has a total budget of 3.4 MNoK, with a period of activity which stretches from June 1997 until June 1998. Forty percent (40%) of the project's total funding has been supplied by NFR.

As indicated earlier, the project has its primary areas of focus within the disciplines of multimedia, communication and security. The technologies in most sharp focus are:

- Ericsson's Multi-Media Telephone System (see chapter 3), and
- the ITU-T Recommendation H.323 (for packet-based multimedia communications systems)[1]

1.2.1 Goals and objectives

At the outset of the work, the goals of the project were stated in a relatively open manner. The goals have been — with focus upon MMTS — to:

- Stimulate creative development new service concepts, and
- Gain further insight into the security requirements for MMTS.

To direct the work itself, a number of more concrete objectives were further specified. These were to:

[1] In one sentence, the H.323 standard "...covers the technical requirements for multimedia communication systems, where the underlying transport is a packet based network which may not provide a guaranteed QoS..." (from [12], page 1). Further details about this standard — in the contexts of MMTS and security issues — are included in chapters 3 and 4.

O1: Engage in competence development and research in data communication and IP-based services

O2: Contribute to work upon MMTS architectural design principles

O3: Gain an increased understanding of security issues within application- and network-levels

O4: Provide input to standardization efforts

O5: Publish general results

O6: Engender synergy with related projects
(e.g., IMiS-Kernel, IMiS-Veritas, ENNCE)

O7: Establish a concrete basis of cooperation between industrial and research work

O8: Generate mutual interest for a follow-up project

1.3 Project Strategy

To further delineate the two primary areas of work in the project, two working teams were defined, each with their own respective themes for study. Certain enabling mechanisms for achieving some of the objectives were also made explicit

1.3.1 Teams and Themes

1.3.1.1 MMTS and Security

The MMTS and Security Team had as its focus:

- To clarify technical requirements and security threats
- To discern to what degree the existing H.323 and H.235 standards meet the needs
- To provide input to the H.235 standardization process.

1.3.1.2 MMTS and Supplementary Services

The MMTS and Supplementary Services Team had as its focus:

- To devise an approach for an H.323 MMTS supplementary service execution environment
- To identify and describe (at least) one new supplementary service concept for improved personal communication and mobility[2]
- To describe a user interface for customer-configuration of personal IAS service data
- To initiate investigation into QoS issues and architectures.

[2] A service called *Intelligent Answering Service* (**IAS**) was identified and described; see chapter 5 f or further details.

1.3.2 Enabling Mechanisms

Certain enabling mechanisms were planned and employed in order to make most effective use of the resources within the project. These mechanisms have include:

- internal and external seminars and workshops
- invited speakers
- exchange of results with other IMiS projects (e.g., via seminars, reports, etc.).

1.4 Project Achievements

1.4.1 Achievement of objectives

When compared to the original objectives in section 1.2.1 above, the IMiS-Ericsson project has specifically achieved:

O1: Competence development and research
- Each team has properly addressed their respective themes
- A large number of (both open and closed) presentations have been held[3]

O2: Contribute to MMTS architectural design
- A preliminary approach for a supplementary service execution environment has been conceived, based upon the principles of Intelligent Networks

O3: Increased understanding of security issues
- Initial identification and general assessment of MMTS security threats has been performed

O4: Input to standardization efforts
- Input to ITU-T Recommendation H.235 was drafted (though not submitted[4])

O5: Publishing
- An open Project Report has been created (this document).

O6: Synergy with related projects
- Open seminars and presentations by guest lecturers have been held (see Appendix A)
- The IMiS Forum and IMiS Reference Group have been established[5]

O7: Cooperation between industrial and research work

[3] A list of the seminars and talks either initiated, given and/or attended by members of the IMiS-Ericsson project is presented in Appendix A.

[4] Lack of submission here was due to the fact that other priorities made it impossible to attend the standardization meeting itself. The project therefore decided against submitting “undefended” input.

[5] The IMiS Forum is a shared forum for the presentation of interim results for each of the IMiS projects. *All* members of the project teams are invited to attend. The IMiS Reference Group is a panel of experts representing the various research fields addressed by the IMiS project family. The Reference Group assesses the ongoing work of the projects, and contributes with feedback about those projects’ content, relevant contacts and new technologies, new project possibilities, areas of critical study, etc.

- A high level of knowledge transfer has been successfully achieved.

O8: Interest for a follow-up project

- An IMiS-Ericsson II project proposal is currently undergoing negotiation

The IMiS-Ericsson project has also engaged in significant contact with other, related projects. These connections are described in the following section.

1.4.2 Connections to other projects

1.4.2.1 Relation to the MMTS Project

The project has been closely related to the ongoing MMTS development project at Ericsson.

Ericsson is taking state-of-the-Art technology and developing a Multimedia Telephony System: from client-based applications, to the network design in which the system operates. The MMTS project continues to reveal several shortcomings in the existing technologies and standards. Evolving technology, new products etc. which address such shortcomings must continue to be evaluated.

Aspects of the IMiS Ericsson project have been heavily influenced by requirements identified during early work within the MMTS projects' system studies. It is expected that any new IMiS Ericsson project must concern itself with topics relevant to the future of MMTS and related products and services.

1.4.2.2 Relation to IMiS - Veritas

The IMiS-Ericsson project is making use of the work being performed within IMiS-Veritas [3]. In particular, IMiS-Ericsson has assessed the functional requirements for the work scenarios selected within the IMiS-Veritas project. These assessments have been used to help develop the tasks and task relationships within the IMiS-Ericsson II project now under negotiation.

1.4.2.3 Relation to IMiS - Kernel

IMiS-Ericsson continues to cooperate with IMiS Kernel, in order that basic research problems encountered within the IMiS-Ericsson project are shared, perhaps even transferred, to the IMiS Kernel project [4]. This kind of problem sharing and transfer helps ensure that the focus of IMiS-Ericsson does not become side-tracked with problems which require new research results which are truly outside of the scope of the IMiS-Ericsson project's resource framework. This kind of activity also helps ensure that the basic research being performed within IMiS Kernel is of immediate relevance to the needs of the more applied problem domains.

1.4.2.4 Relation to the ENNCE Project

ENNCE (Enhanced Next-Generation Networked Computing Environment) is a project spanning 1997, Q4 - 2001, see [5]. The partners involved are the Department of Computer

Science at the University of Tromsø, UIO Ifi, UNIK, Telenor Research and Development, NR and UNINETT.

The work currently funded within the project includes:

- development of a generic reference model for distributed multimedia applications, specifying a technology-independent architecture, and modeling how the quality-of-service requirements should be handled in end-systems and network.
- selection of one or more distributed multimedia applications which can be used to demonstrate the capabilities developed in the project
- development of a flexible middleware that supports a broad range of QoS requirements including: continuous media streams, heterogeneous multicast and high throughput communications with constrained latency.

This work is obviously quite relevant to the IMiS Kernel work, and is therefore of relevance to this project as well. For this reason, efforts have been made to share intermediate results and competence where possible, through the use of seminars and reports (e.g., within the IMiS Forum).

1.5 Structure of the report

The structure of the report contains the following major elements:

- a description of the project's *Technical Context*
- an *Overview of MMTS*
- a chapter which presents an *Assessment of MMTS Threat Scenarios*
- a description of the *Intelligent Answering Service (IAS)*
- a description of the (preliminary approach) for a *Supplementary Service Execution Environment*
- an illustration of the *IAS user-interface* for service configuration
- two chapters related to QoS issues: one concerning *Adaptivity* and another concerning *Elements of a Generalized QoS Framework*
- a brief chapter which describes *Areas for Future Work*
- two appendices: a *List of Seminars and Talks*, and the *Input Drafted for ITU-T Recommendation H.235*
- a list of *References*

Chapter 2

Technical Context

2.1 The Public Intranet

The ***Public IntraNet*** defines a business concept and a functional framework for development and deployment of commercial IP based services. The framework comprises an IP network solution and a service network which includes end user services and platform support functions. Among the platform support functions being offered are functions for service subscriptions, service charging, security, management and resource control.

The ***IP network*** is modeled as a generic, high capacity IP network capable of supporting real time media and offering a high quality of service. The IP network layer is positioned as an interworking layer between the diversity of data transport (i.e. MAC layer) protocols, ranging from LAN accesses based on ATM, Ethernet and xDSL to ATM based WAN networks with interworking functions to ISDN and PSTN.

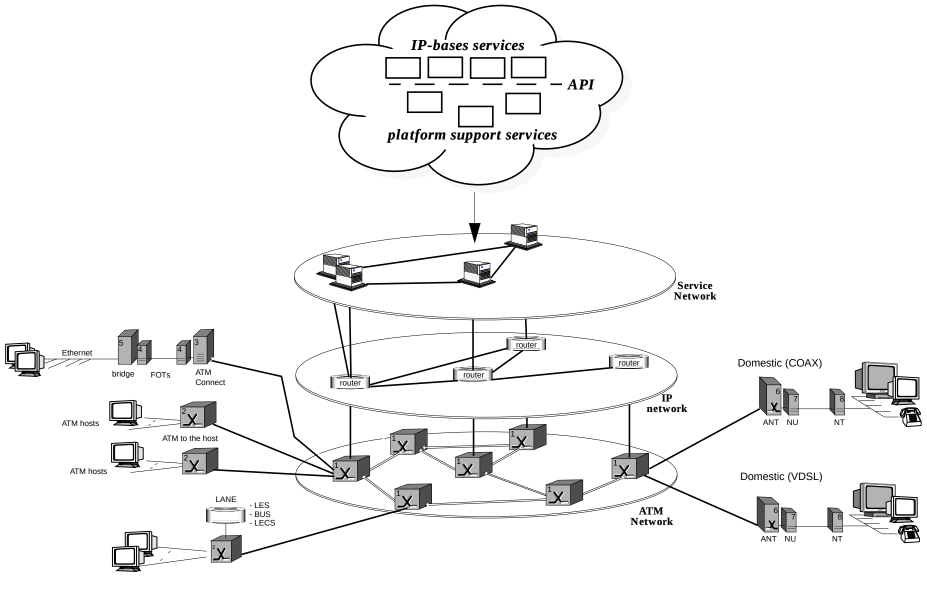


Figure 1 : *Public IntraNet business concept*

The IP network consists of a number of IP routers that interconnect customer LANs and service LANs with the transport network. Some of these connections are through application layer or bastion host firewalls, thereby ensuring the integrity of the network.

The ***Service Network*** provides a generic service platform and a set of services. It is targeting both the business and residential market and is adaptable to different national market needs. The Service Network is based on distributed object technology (CORBA), but also makes extensive use of both the JAVA and WWW technology.

The Service Network is composed of a number of access control and authorization servers (denoted ‘brokers’) that perform user authentication and service authorization, plus a number of dedicated application servers that supply selected services such as mail, news and IP-phone to the user.

2.2 Service Network

The ***Service Network Platform*** represents the core of the Public IntraNet. In here are found generic support for service brokering and execution control, being represented by a stack of support facilities for subscription, charging, resource handling, security and management. These support facilities are made available to the services through a set of APIs and toolkits.

The ***Service Network Applications*** being deployed on the service network platform can comply to and use these mechanisms to different degrees, thereby obtaining different levels of integration into the Public IntraNet. The platform includes a *Service Creation Environment* in which applications may be developed from scratch using the set of support services as provided by the APIs, or just deploying a 3rd.party application as it is using only a subset of the available support services.

The ***Service Creation Environment*** is based on the 3-tier model and the de facto standard CORBA 2.0 as specified by the OMG (Object management Group). CORBA provides the means for implementing a distributed object-environment, and includes IIOP for object interoperability and IDL (interface specification language) for language neutral interfaces.

2.3 The MMTS Platform

The Multi-Media Telephony System (MMTS) offered by Ericsson enables interpersonal communication between desktop users on IP networks as well as gateway functions to PSTN and ISDN networks. These communication services range from the very basic audio calls through more advanced audio, video and data conferences. All MMTS services are offered as part of a service framework which includes support functions for service control and user mobility in excess to that standardized by the H.323 standard [12].

The MMTS system can either be delivered as an integrated service in the Public IntraNet platform or as a stand-alone solution on a separate scaled down service platform. The same set of functions are, however, provided in both cases and includes:

- Service Subscription and Control
- User Database and Directory Service
- End-User Mobility
- Network, Service and Customer Management
- Security, Authentication and Access Control
- Charging Support
- Resource Management

These Service Platforms will be extended over time to form a natural base for building and integrating value added services to the H.323 communication services.

2.4 H.323’s Gatekeeper Entity

The gatekeeper has the role of providing H.323 communication services to the set of endpoints in the H.323 network. The basic services now being provided are according to the specification of the ITU-H.323 standard, version-1. This includes control functions for giving access to clients, gateways and multipoint control units (MCUs) as well as basic address translation and routing functions. The Ericsson MMTS gatekeeper extends this standard functionality with a set of additional functions for network- and service provider control.

Within the MMTS network concept, the gatekeeper is the key component as it enables call- and resource control functions for the service provider, i.e. the gatekeeper adds the previously listed platform functions to those of the H.323 standard which are in brief:

- Audio, Video and Data Phone-Calls
- Audio, Video and Data Conferencing
- Application sharing, shared whiteboard
- File Transfer and Communication Tools (e.g. talk)
- GW functions (H.320, PSTN)

These control functions offered by the gatekeeper and the service platform are crucial for service providers which want to control the use of network resources and charge for the use of these. The gatekeeper is also a natural entry point for inclusion and offering of network-centric functions, e.g. user mobility, voice mail and directory services.

Chapter 3

Multimedia Telephone System (MMTS)

3.1 General Overview

The Ericsson Multi-Media Telephone System (MMTS) allows end-users to place audio, video and/ or data calls over an IP network. Three different call-scenarios are offered, being PC-to-phone, phone-to-phone and PC-to-PC.

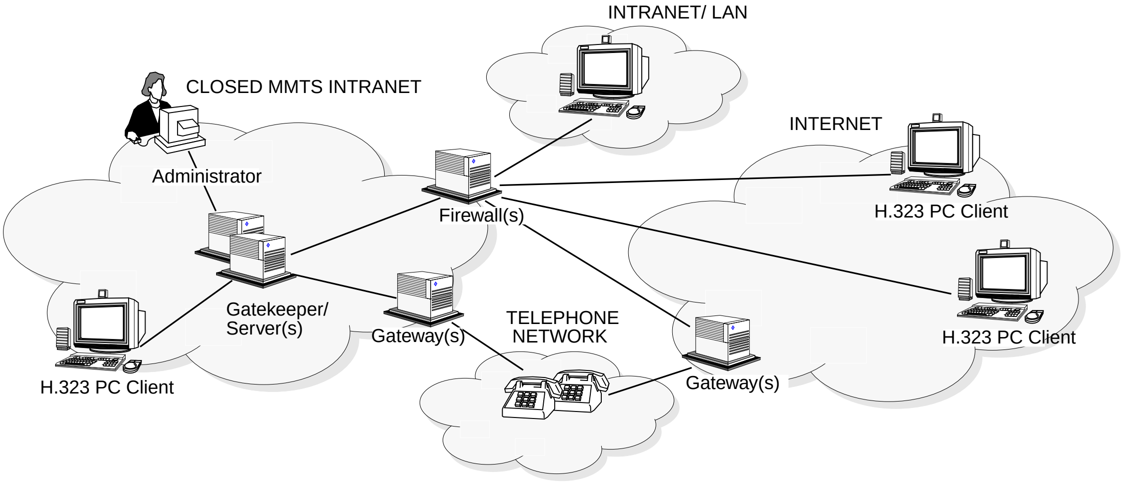


Figure 2 : *MMTS Overview*

3.1.1 Phone-to-Phone

In the phone to phone scenario, both the originating and terminating terminal/ phone are connected to the public switched network, but the call is carried by an IP network. Within this scenario, two different call (sub)scenarios are being supported:

1. The MMTS network is end-user aware and authenticates, authorizes and charges the end-users for their use of the service(s).
2. The MMTS network is not end-user aware, but is configured to trust all traffic originating through a set of gateways. The traffic is controlled and charged towards these gateways and the phone provider(s) associated with these.

3.1.2 PC-to-Phone

In the PC-to-phone scenario, a call is established between a party on the switched telephone network and another on an IP network. The call either originates from a standard telephone connected to the telephone network and terminates on the network user's H.323

terminal, or visa versa. If the call is to be carried over the internet, both the H.323 terminal application and the gateway need to support a low bit-rate codec.

3.1.3 PC-to-PC

In the PC-to-PC scenario both the originating and the terminating party are on the IP network and the connection is set up between the two H.323 terminals. As this provides for a better interface for the end-user, it is possible to exploit the more media-rich communication facilities provided by the MMTS solution including audio, video and data communication.

3.2 Functional Overview

3.2.1 MMTS User Functions

The provider of the MMTS network can offer a number of services to the subscribers and end-users of the MMTS H.323 network. These services includes:

H.323 client-to-H.323 client calls

The H.323 clients are PC based programs that support establishment and control of audio, video and data communication channels. This provides for the following set of alternative IP communications services:

- audio communication
- audio + video communication
- audio + data communication
- audio + video + data communication

H.323 client-to-H.320[1] phone calls

The PC based H.323 clients can communicate with PC based H.320 clients in the same way they communicate with other H.323 clients. The main differences is related to the perceived quality of the communication channels (video in particular).

H.323 client-to-GSTN[2] phone calls

The users of the H.323 network are able to originate and receive calls from subscribers/ terminals on the GSTN networks. The communication facilities will in these cases be restricted to audio-only.

[1] H.320: The ISDN based conferencing solution corresponding to H.323

[2] GSTN: Generic Switched Public Network.

GSTN phone-to-GSTN phone call

An end-user can actively select to route a call through an IP based H.323 network, typically for reasons of cost. To do this the user will dial a prefix (carrier access code) that selects the H.323 network as the carrier of the call.

Conference Calls

When conferencing equipment is included in the network, end-users can setup and participate in H.323 multiparty conferences, allowing 3 or more participants to take part in a call. The communication facilities (audio, video, data) offered to each participant are limited by the combined restrictions of their local equipment, their peers and the resources booked and provided by the conference equipment (note that the conference is not “downgraded” to the least common denominator of all it’s participants).

3.2.2 MMTS Provider Functions

3.2.2.1 Support Functions

Service Subscription

The service subscription functions provide support for subscribing to services and facilities within these. The subscription model offers a data model that provides the basis for other functions such as user mobility, charging and access control.

Database and Directory Functions

The system provides an internal database for the traffic user- and service data, i.e. containing the data required for traffic execution of services. This encompass the traffic view of user- and service profiles as well as routing data, charging configuration data, etc.

This database has an interface to — and is populated from an off-line database containing — higher level (e.g. business-level) data profiles of users, services, etc.

End-User Mobility

End users can place and receive calls from any terminal being connected to the telephone network after having registered their presence at this terminal. This means that network implements a database of all users and keeps track of these user’s location in the network.

Management

The management system provides management functions within the following domains:

- Network Management (including Element/ Server management)
- Service Management
- Customer Care System (interface and small-scale applications)

The functional areas being covered in these management domains are:

- Fault and Alarm Management
- Configuration Management
- Charging and Accounting Management
- Performance Management
- Security Management

Authentication and Access Control

The telephone network offers “identity based services” such as user mobility and service subscription/ user profiles. This requires that the user identities are established and trusted and that their credentials (rights) within the service domain are enforced. To support this, a set of authentication and authorization routines are being provided.

Charging

The charging functions include a set of support functions for generating charging events and for formatting and transferring the resulting charging records from the traffic system to some off-line system for post-processing and billing.

3.2.2.2 Communication Functions

Gatekeeper Functionality

The gatekeeper is the logical switch of the H.323 network, taking part in- and controlling the call- setup and call-control channels. This ensures that the gatekeeper can add a level of network intelligence to the functionality of the H.323 clients. Among the central functions are:

- H.323 network point-of-presence (including gateway and conference functions)
- user mobility
- supplementary services
- Quality of Service (QoS) control/ offering
- user authentication and access control
- service charging, fault surveillance and performance monitoring

Gateway Functions

Gateway functions provide access to and from other telephone network such as ISDN and PSTN. This is provided through gateway functions that convert the call, control and media streams to- and from the signalling standards being employed in the different networks.

Multiparty Conferencing

Multiparty conferencing (for 3 or more participants) is enabled by means of a Multipoint Control Unit (MCU). This is a functional entity (typically a server) that controls the multiparty session and mixes[3] the media-streams being sent to- and from the different participants in the conference.

Firewalling

Firewall functions perform screening functions on protocol layers for the H.323 protocol and other IP based services. This ensures that proper screening of traffic can be enforced on the boundaries of the different administrative domains, being e.g. the Internet or CPNs.

[3] Mixing media streams (e.g., audio) is performed by a Multipoint Processor, an H.323 entity which may optionally be implemented within an MCU.

Chapter 4

Assessment of MMTS Threat Scenarios

This chapter analyses the basic threats, vulnerabilities and impacts associated with H.323 multimedia communication over packet based networks. The goal is to establish a threat model to be used in the MMTS Security work of the IMiS Ericsson project.

The chapter focuses on the basic protocols, components and elements in this type of communication. Only a few fundamental application specific issues are addressed at this point, such as those associated with user logins.

4.1 General

The MMTS at focus here is part of an overall business concept, called the Public Intranet Service Network, studied at Ericsson. Ericsson plan to deploy various commercial IP based services, such as MMTS, over wide area networks, and will also connect and integrate the services into other telecom networks. The overall Public Intranet is studied in the Ericsson PARIS Project and described in more detail in [15].

The MMTS is based on the ITU-T H.323 family of protocols [12]. ITU-T has also addressed H.323 security in H.235, as described in [16], [17] and [18]. These documents are currently drafts and are subject to ITU-T standardization efforts. It is an important, early goal of the risk analysis to be able to influence these emerging standards.

4.2 Basic Preconditions

This chapter will primarily focus on the basic functioning of H.323 multimedia communication in typical environments. Some important application specific issues will be addressed, but the main focus will be on the basic protocols and components as they are used in common types of communication. This is partly because not enough information about the applications is available, and partly because we want to influence the ongoing standardization process as soon as possible. Therefore:

- Only the MMTS portion of the Public Intranet Service Network will be considered, although certain aspects of the overall picture might affect MMTS security.
- The threat assessment will focus on technical aspects and will ignore questions related to organizational measures and physical protection.
- The threat assessment will primarily be interested in confidentiality and integrity issues. Availability will only be of interest in certain denial-of-service attacks, and redundancy issues will not be addressed herein.

4.3 System Description

The material which follows offers a description of an MMTS context, along with information about H.323, H.235, client logins and call set-up within such a context. Descriptions of stream communication and call teardown are not included here.

4.3.1 Overall Description

This system description will focus on MMTS relevant components and functions.

The MMTS services described below are based on *IP over a high capacity transport network architecture*. In this description, the network itself is assumed to be owned by a public service provider. A service center and several customers are assumed connected to the network — a network having connections to other networks, such as PSTN and the Internet, through gateways. Customers may connect from

- terminals directly connected to routers in the IP network
- customer LANs connected to routers in the IP network
- network service providers connected to routers in the IP network
- directly connected to the underlying high capacity transport network.

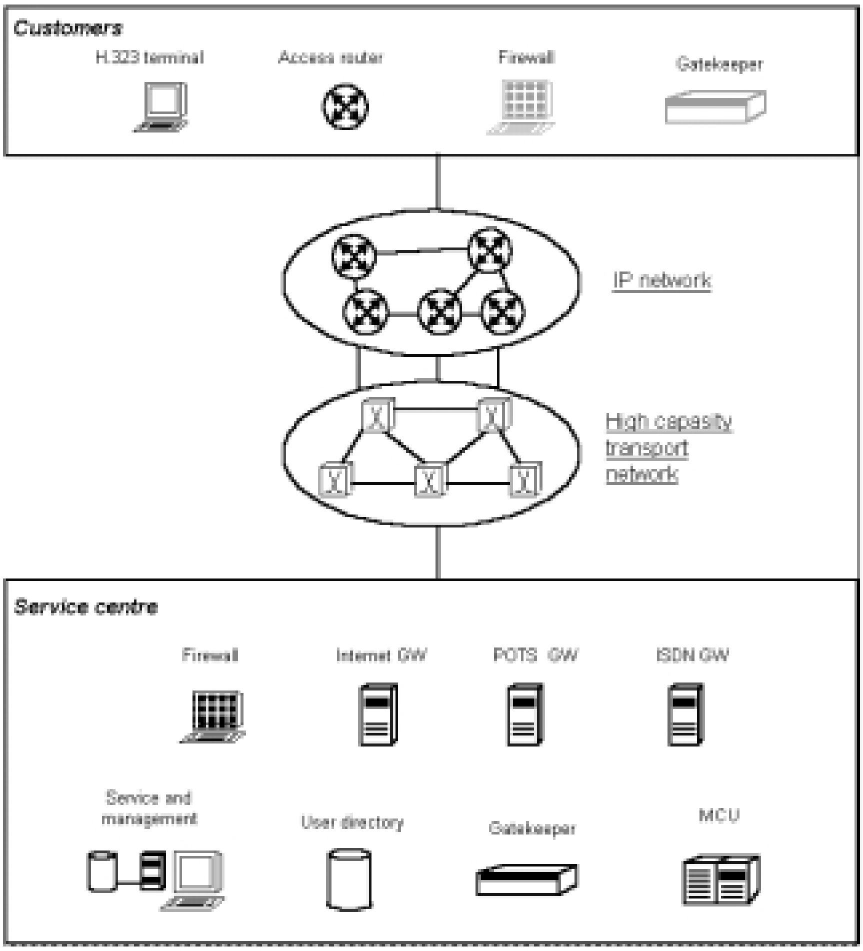


Figure 3 : *Overall system description*

The service centre includes one or more gatekeepers, MCUs and gateways in order to support multimedia communication between customer clients and between customer clients and external non-customer clients. The service centre also includes servers, databases and other components in order to handle access brokering, service brokering, service management and assure proper billing and subscription. The service center may also support Public Key Infrastructure (**PKI**) and yellow book services.

The customer LANs, as far as MMTS is concerned, consist of one or more multimedia terminals and optionally one or more local gatekeepers and, perhaps less likely, MCUs. Customer LANs connect through access routers and optionally, through firewalls. Customer terminals can be stationary or mobile.

Two or more users then communicate audio, video and/or data using the H.323 family of protocols in this environment.

The system solution can also provide PSTN to PSTN communication. In this case end users communicate with phones connected to nearby system gateways, so that end users - for instance - can communicate overseas for charges associated with local calls. The communication can be routed through a central gatekeeper.

4.3.2 Multimedia Components

The H.323 multimedia components, as described in detail in [12], are

- terminals
- gatekeepers
- MCUs
- gateways and
- firewalls.

One physical component may serve more than one of these component functions.

The users communicate using H.323 terminals, or - with the help of gateways - using other kinds of (external) terminals such as POTS, GSM or ISDN equipment. A H.323 terminal may be a powerful PC with audio and video I/O and codec capabilities, or it may be some kind of an IP phone. Audio support is mandatory; video and data is optional. In the PSTN to PSTN case, end-users use regular phones.

H.323 gatekeepers provide call control services to H.323 terminals, gateways and MCUs. They also provide address translation and control admission to the network, and they handle bandwidth requirements. As described in detail later, the public intranet gatekeepers will act as both access brokers and service brokers for clients. They will also gather charging information to be sent to a charging collector function in the service centre.

MCUs provide the capability for three or more terminals to participate in a multipoint conference. MCUs consist of a mandatory Multipoint Controller (MC) and an optional Multipoint Processor (MP). The MC provides conference control, and the MP provides central processing of audio, video and data streams in (centralized) conferences.

Gateways handle the connections to other types of network, and firewalls provide LAN access control to some degree.

4.3.3 Protocol Overview

As described in detail in [12], the following protocol families are used within, and/or strongly related to, H.323:

- H.225.0 RAS
- H.225.0 Call Signalling
- H.245 Channel Control
- RTP
- RTPC
- T.120

The underlying protocols used at the transport layer are TCP (for reliable transport) and UDP (for unreliable transport). At the network layer IP is used.

H.225.0 RAS messages are exchanged between a gatekeeper and a terminal/gateway/MCU to convey registration, admission, bandwidth change and status messages. H.225.0 RAS uses UDP and will be initiated on well known ports; usually UDP/1718 (for discovery) and UDP/1719.

H.225 Call Signalling is basically used to convey the call set-up and teardown messages between H.323 entities. Terminals may communicate call control messages directly (direct call) or through one or more gatekeepers (gatekeeper routed call). The call signalling uses TCP and will be initiated on a well known port; usually TCP/1720.

H.245 Channel Control is used for capability exchange and the set-up of communication channels for audio, video and data, and for the exchange of various control parameters. An ephemeral TCP port agreed upon in the call signalling is used. The H.245 control channel may also be routed directly between terminals or through one or more gatekeepers.

RTP over UDP is used for the communication of audio and video streams. H.323 also use RTPC for accurate control of these streams. The RTP and RTPC channels are uni-directional, so that four UDP connections are associated with each stream; one RTP and RTPC connection in each direction. The ports associated with RTP and RTPC UDP connections are required to be one number apart, with the RTP port being even and the RTPC being the next higher odd.

Note that the T.120 set of protocols, used for data streams, work somewhat independently. T.120 communication may be set up separately, and even prior to the call establishment described in this chapter. This is however not the preferred mode of operation.

Note that H.323 supports many modes of operation. Communication between terminals may for instance not involve gatekeepers. When gatekeepers *are* used, certain parts of the communication may be routed through the gatekeeper while other parts bypass. We will assume that gatekeepers are used, and that the call signalling and the H.245 control channel are routed through gatekeepers, while audio, video and data streams bypass gatekeepers for performance reasons.

Also only centralized conferences are analyzed initially. This means that a MCU administers multipoint conferences. It receives both call signalling and call control at the MC for central control and the audio, video and data streams at the MP for central processing.

4.3.4 Client Logins

How end-users log on to their workstations or network is not covered by H.323. We describe how this likely will be done in the MMTS part of the public intranet. We will focus on clients that are end-users using either a H.323 enabled PCs or a PSTN phones.

The objective of the client login procedures is to ensure proper identification and authentication (**I&A**), as well as authorization and access control. The login procedure will have to:

- Authenticate the end-user (or client) to the network (or the network services)
- Authenticate the network to the end-user
- Associate a proper user profile to the end-users
- Tie terminal identities, such as IP addresses or E.164 numbers, to specific end-users

PC clients will first authenticate the network, i.e. a public intranet gatekeeper, by requesting a X.509 certificate, using Transport Layer Security (**TLS**, see [19]) and a browser based GUI. The client will then log on using a login applet returned by the gatekeeper in the process. The method used for end-user authentication may vary; it could be based on user certificates, passwords or smart cards.

For user authentication and user identification, PSTN clients can use user-ID and pin-code. For such clients, however, it will not be possible to fully establish a trusted binding between the network and the end-user using A-numbers[1] alone. A-numbers can only identify fixed terminals.

The gatekeeper will, when the identity is properly verified, connect to a user profile database in order to assign the proper authorizations to the user (H.323 PC client) or terminal (PSTN phone) as illustrated in figure 4. Customers or network operators may operate customer databases, and the public intranet will probably have to mirror or duplicate certain parts or subsets of such databases as needed.

The gatekeeper discovery and registration process described in the next section, will probably be an automated part of the client login procedures.

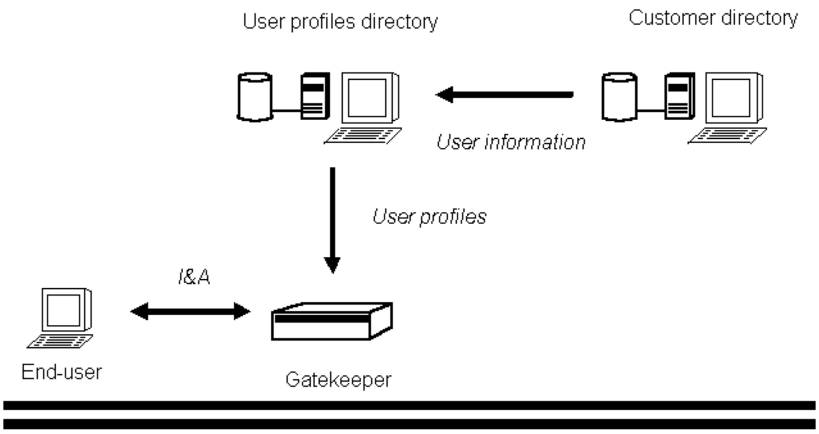


Figure 4 : *Client login and authorization assignment*

4.3.5 Call Set-up

The message sequence involved in call set-up between two endpoints[2], is shown in figure 6 (one gatekeeper) and figure 7 (two gatekeepers). In these scenarios both endpoints have registered at their gatekeeper prior to this; see figure 5.

The names and types of the messages are indicated together with the transportation protocol used (UDP or TCP) and the associated port number.

The gatekeeper discovery and registration process is simple. Basically the endpoint gets registered at a gatekeeper, and exchanges address information in the process. The gatekeeper will typically end up associating an alias address, such as an E.164 number or an email address, with the IP address of the endpoint.

[1] A-numbers are the numbers associated with specific terminals (e.g., telephone numbers).

[2] Endpoints are callable; in H.323, these are H.323 terminals, MCUs and gateways.

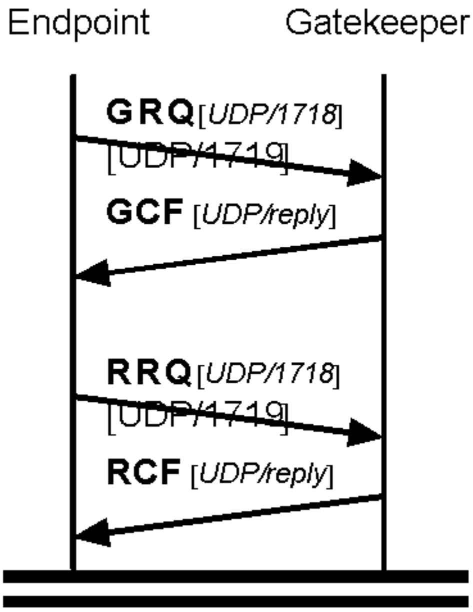


Figure 5 : *Gatekeeper discovery and registration*

Note that gatekeepers may communicate with the service centre in order to check which user is currently logged on with the corresponding IP address[3]. This is an important part of the broker function of the gatekeeper. It is highly likely that the gatekeeper discovery and registration process will be an automated part of the client login procedures. The gatekeeper may cache this type of user information for later use or retrieve the information as needed.

RAS channels are also used for making bandwidth change requests (BRQ/BCF/BRJ), for location requests (LRQ/LCF/LRJ) and for status information requests (IRQ/IRR). The latter may also be requested periodically by gatekeepers using interval specifications in ACF messages.

The subsequent call set-up, when both endpoints are registered to the same gatekeeper, is also straightforward. The calling endpoint asks the gatekeeper if it can make a call[4] and sends a set-up message to the gatekeeper containing the address (perhaps an alias) of the receiving endpoint. The gatekeeper forwards a set-up message to the receiver, and the receiving endpoint then basically asks the gatekeeper if it can answer the call and sends OK back to the caller through the gatekeeper in similar manner.

[3] As mentioned, this may not be possible with all types of client equipment.

[4] Note that endpoints also indicate bandwidth requirements to the gatekeeper in ARQ messages.

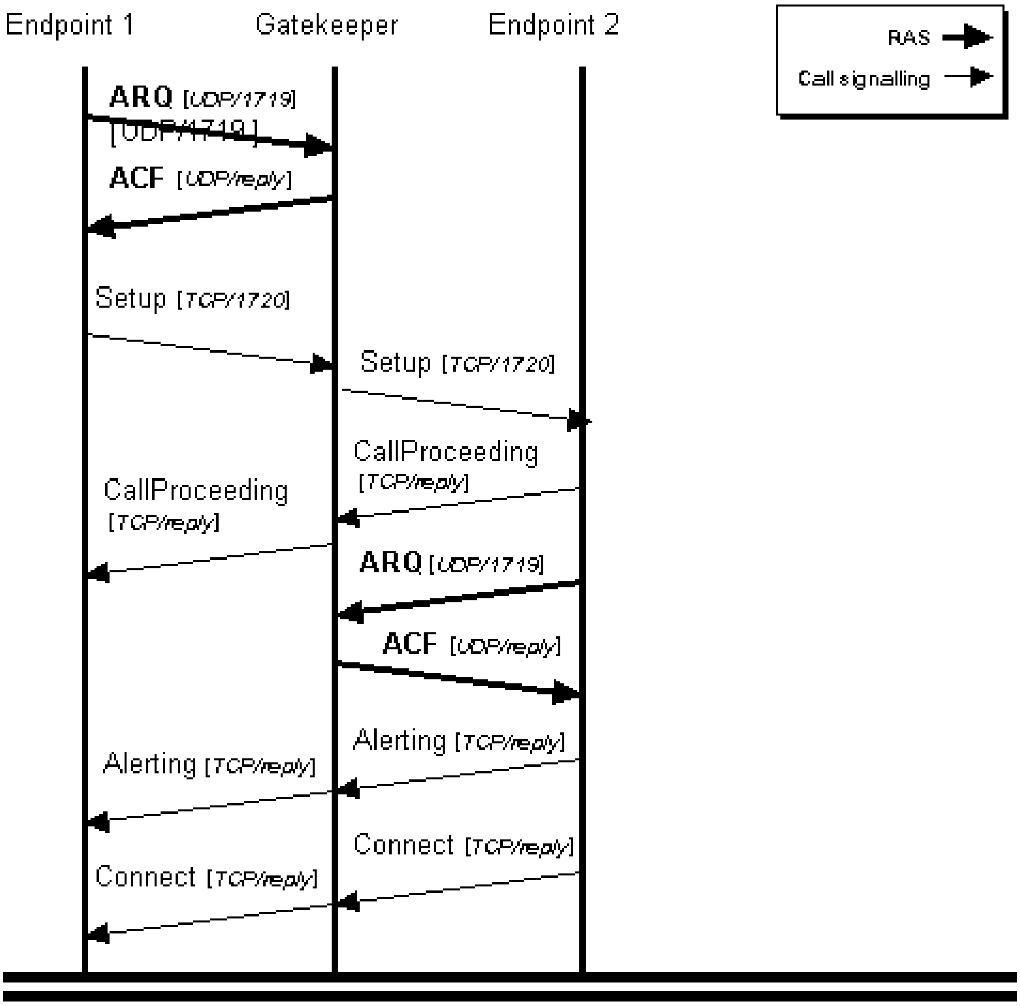


Figure 6 : *Both endpoints registered to same gatekeeper: Direct call signalling*

H.245 control ports are exchanged in the process[5], and the participants are ready to proceed by opening H.245 control channels.

When the endpoints are registered to different gatekeepers, the message sequence is slightly more complex. The calling endpoint will not notice any difference, and most of the extra complexity will be handled by its gatekeeper. What happens is that the receiving endpoint sees its gatekeeper requires routed call model in the ACF response. It then sends a facility message back (to the caller's gatekeeper), saying it should route the call to its gatekeeper, and indicates the gatekeeper address. The caller's gatekeeper terminates the first connection and sends a new set-up message through the receivers gatekeeper. The communication then proceeds in normal fashion, though with two intervening gatekeepers instead of one.

[5] Usually the receiving endpoint (and subsequently the gatekeeper) includes its H.245 TCP port in the 'connect' message, so that the calling endpoint (and subsequently the gatekeeper) can initiate a H.245 control channel connection.

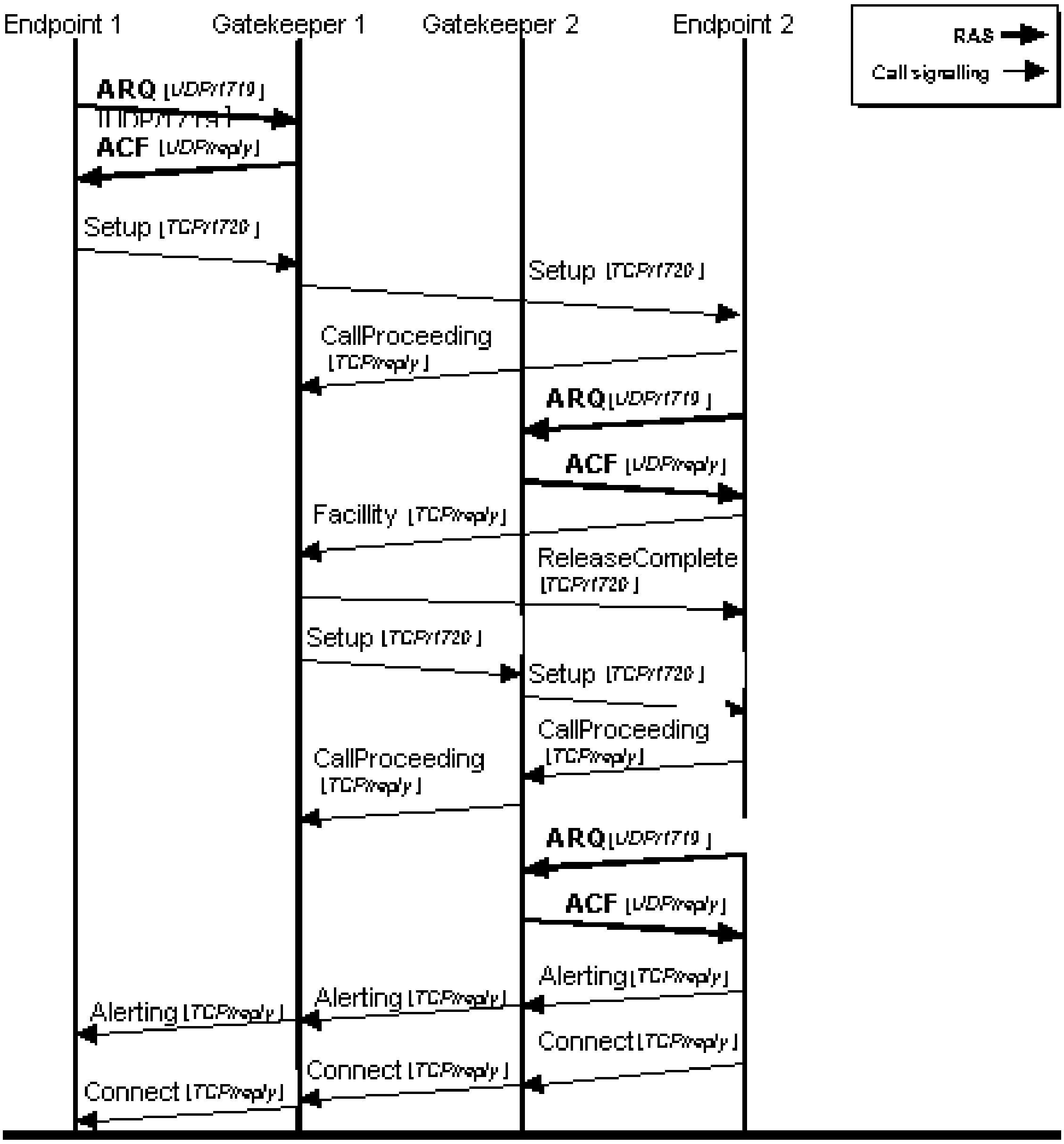


Figure 7 : *Both endpoints registered to different gatekeepers: Direct call signalling*

After the call set-up, the endpoints and gatekeeper(s) open up H.245 control channels on the exchanged ephemeral TCP ports between each other. These bi-directional channels can be routed through the gatekeeper(s) without further control signalling. Once the control channel is up, participants are free to terminate the corresponding call signalling channel. The signalling channels are how ever typically left up.

The endpoints exchange their audio, video and/or data capabilities on the H.245 control channel, before they agree to open up logical channels for the RTP multimedia communication. The UDP ports to be used for the RTP stream communication are among the things contained in the H.245 messages.

4.3.6 A Short Description of H.235

H.235 proposes to use TLS to set up a secure H.245 control channel. The communicating endpoints will then exchange (symmetric) crypto keys and associated parameters on this channel, so that the RTP multimedia streams subsequently can be properly protected. TLS options, such as the need for authentication, can be assigned during the call signalling. Call signalling channels may themselves be (a priori) TLS secured.

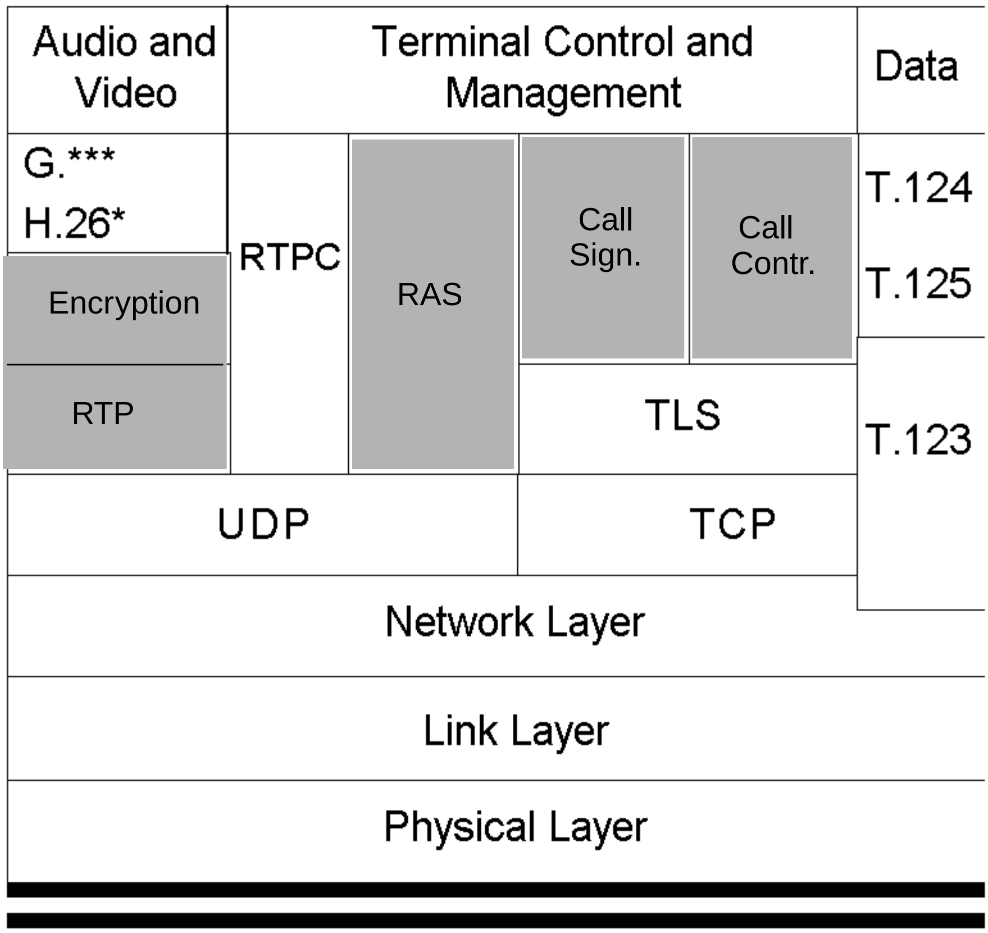


Figure 8 : *H.235 scope (shaded areas)*

The current version of H.235 uses certificates (only) for authentication. Certificates are used in TLS handshakes and can also be exchanged on demand over the H.245 control channel.

H.235 also proposes some degree of integrity protection of RAS messages, by encrypting certain message fields, for instance using a Diffie-Hellman key exchange scheme [20].

Note that H.235 does not address security of data applications, such as those based on T.120.

Note: In the latest draft of H.235, IPSEC [21] is also offered for the protection of the call signalling and the call control.

4.4 Threats

The assets at risk are generally the data, the services and the components associated with the MMTS solution described. The data consists of both user data and all types of management data. Some services can be described or characterized by a set of “Quality of Service” (**QoS**) parameters[6].

Both the types of threats and the source of such threats need to be considered.

4.4.1 Classes of Threats

4.4.1.1 Improper identification or authentication

Proper I&A is of fundamental importance for the overall security. Other security enforcing function, such as access control, will depend on it. This class of threat includes spoofing and masquerade attacks, all kinds of forgeries and the like. Note that many different kinds of identities are at risk, not only persons. Also organizations, roles, functions, addresses, phone numbers, terminals, servers etc. may be spoofed.

As mentioned, attackers may control parts of the network. In this situation it is not sufficient to authenticate parties when initiating communication alone; sessions may be hijacked later on. It seems necessary to use both proper I&A and encryption. This may not be possible in all situations. It may be impossible to authenticate external users sufficiently, and encryption may be ruled out for performance reasons in some cases.

It seems that public key technologies may be used extensively also for I&A purposes. Public key technology is very promising and offers several important functions that are needed in the public intranet. It should be noted how ever that public key protocols are vulnerable to certain known cipher text attacks, and it is not well known among regular users that they, for instance, may compromise information by signing or decrypting data presented to them, in some way, by attackers. Users may also fail to verify essential data (that perhaps could be poorly understood), such as certificate fingerprints, and they can end up trusting hostile servers and sites if proper routines are discarded.

It could be noted that voice recognition is commonly accepted as a valid authentication criteria, and even more so when also MMTS video is used. This may be inadequate for critical applications.

4.4.1.2 Unauthorized access

Protection from all kinds of unauthorized access is of key importance for the correct functioning and configuration of the MMTS system solution. The risk of unauthorized access to files, servers, databases, applications, services, routers, LANs etc. must be considered.

The MMTS solution is no doubt complex and includes a wide variety of services, technologies and platforms. It also involves several categories of people, such as third party service providers, network operators, non-customers, customers and administrators. This

[6] For further information concerning QoS issues, see chapters 8 and 9. Section 9.2.2 deals specifically with QoS specifications.

means that proper access control and authorization must be devised, and policed, at many levels: from the WAN and LAN level (i.e. firewalls, gateways and routers), to individual servers, databases, files and directories. And it is highly likely that configuration errors will be introduced at some point that attackers may be able to exploit. All though organizational measures will not be addressed in this chapter, we would like to emphasize the need for well documented policies, requirements and instructions for the secure configuration and operation of the MMTS system.

4.4.1.3 Inadequate traceability

Another important security function is the ability to track down events; especially in a system as complex as the one considered. This comes down to proper auditing and recording of events - and the processing of these, and it may also be necessary in some applications and services to have non-repudiation.

4.4.1.4 Unauthorized modification or insertion

For most business applications, the integrity of the data is more important even than the confidentiality of data. The integrity of all types of data being stored, operated and/or transmitted must be considered. The risk includes all kinds of unauthorized fabrication, insertion, reordering, deletion, alterations and/or replays of communication packets, and similar modifications to files, user or application data, management data, etc. The ability to detect modifications may also be of importance.

4.4.1.5 Unauthorized disclosure

This covers all kinds of loss of privacy. It also includes traffic analysis issues and the disclosure of information that may facilitate certain kinds of attack.

4.4.1.6 Unauthorized reduction of availability / QoS

This includes all kinds of denial-of-service attacks. Performance may degrade because of mis-configuration, because someone is flooding packets, making repetitive resource requests, refusing to free resources, or in other ways, trying to clog the system. Attacks of this kind would clearly degrade the quality of service achieved during service delivery (e.g., degradation of audio quality). For real-time applications such as audio and video, it is furthermore of key importance to preserve synchronization.

4.4.1.7 Other threats

This includes threats from “trojan horses”, viruses etc. Such infected programs or data may have different strategies for reproduction and may result in one or more of the threats mentioned above.

4.4.2 Sources of Threats

A manifested threat, whether it is caused by accident or deliberate attack, may arise from one or more sources. The sources considered are the following:

- Humans; in our case humans can be categorized as:
  - third party service providers
  - network operators
  - customers
  - administrators
  - non-customers; this category includes hackers and competitors, who perhaps are of foremost concern.
- Software
- Hardware
- Other

In the MMTS context described in section 4.3, attackers may have full control of some parts of the network and may be able to launch full blown man-in-the-middle attacks.

4.5 Case Studies

We will discuss a number of use cases more carefully and analyze associated risks and vulnerabilities. Certain use cases have been postponed for later study; these are named in section 4.5.5. Here, the focus is more on the applications and services themselves. Currently the emphasis is upon the protocols and their use, as well as how well the emerging H.323 and H.235 standards are suited for the MMTS public intranet system solution. From a protocol perspective, most of these use cases below are not that very different.

We will, as mentioned, assume throughout that call signalling and H.245 control channels are routed through gatekeepers, while audio, video and data streams bypass gatekeepers for performance reasons.

4.5.1 Terminal-to-Terminal Communication

The basics of terminal-to-terminal communication, where two H.323 terminals communicate through one or two gatekeepers, is covered in section 4.3.5.

When H.235 TLS encryption is used to protect the H.225.0 call signalling and/or the H.245 call control, separate TLS sessions must be established between the terminals and the gatekeepers, and between the gatekeepers. The number of TLS sessions needed to establish an end-to-end encrypted, secure H.245 control channel in this case is g+1, where g equals the number of gatekeepers involved. The number g could be 0, 1, 2 or 3. If TLS encryption is needed also for the H.225.0 call signalling, twice that number of TLS sessions is required. Although TLS handshakes may be carried out in advance and TLS session parameters may be cached, this may result in extensive delays.

It may not be necessary to encrypt communication between a terminal and a local gatekeeper, for instance if they are connected to an adequately protected local network.

4.5.2 Multipoint Communication

The multipoint communication scenario that will be considered initially — illustrated in figure 9 — is a centralized multimedia conference between three or more user terminals. The terminals may be H.323 terminals, or - using gateways - PSTN or H.320 terminals. The central MCU is controlling and administering the conference by receiving all media streams. A gatekeeper-routed call model is assumed, as illustrated in figure 9. Local gatekeepers may optionally be used.

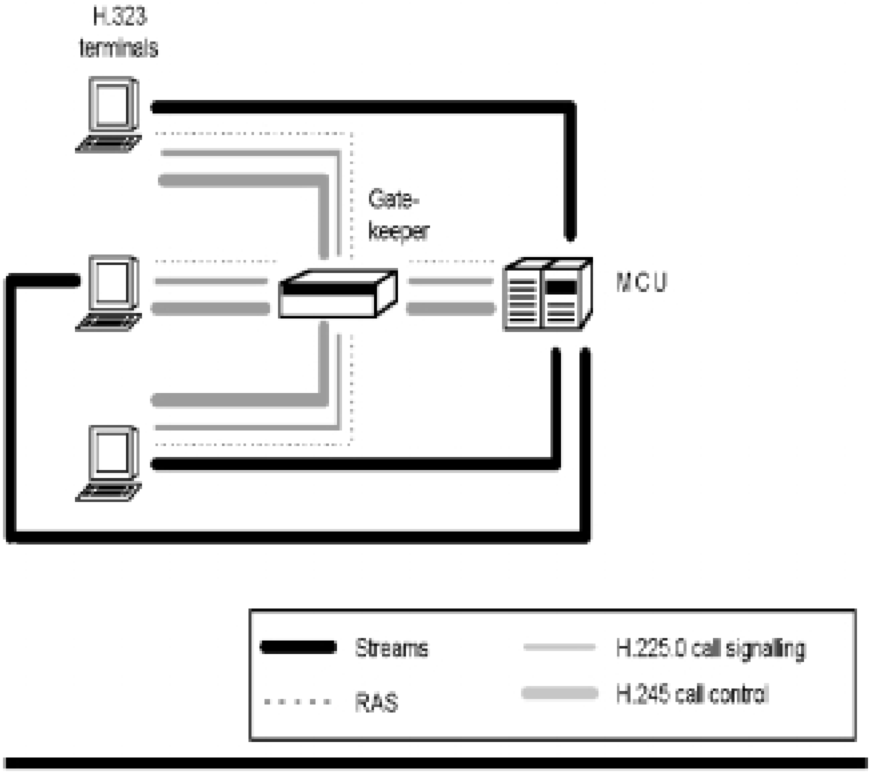


Figure 9 : *Centralized multimedia conference - gatekeeper routed call signalling and call control*

The endpoints register at the gatekeeper as described in earlier sections. The call set-up uses the usual set of protocols, but the exact sequence will depend on the type and nature of the conference. We will initially focus on scheduled or call-up multipoint conferences. In these cases an initiator orders a conference at the MCU, and the participants are called up by the MCU at the scheduled point in time. The initiator of the conference must supply necessary references of the participants to the MCU. Special information, such as a conference password, may be required in order to join the conference. Exactly how the initiator orders a conference is yet to be determined.

Again, if TLS encryption is used to protect the call signalling and/or the H.245 call control, separate TLS sessions must be established between the terminals and the gatekeepers, between the gatekeepers, and between gatekeeper(s) and MCU(s). A gatekeeper and MCU will often be connected to the same local network and it may not be necessary to encrypt gatekeeper-to-MCU communication.

4.5.3 Gateway Communication

Two types of gateways will be considered:

* H.323/PSTN (POTS, GSM, ISDN) gateway

* H.323/H.320 (ISDN) gateway

The H.323/PSTN may support ISDN audio, but not ISDN video. The latter requires use of a full H.323/H.320 gateway.

The gateways have the characteristics of a H.323 terminal on the public intranet side and the characteristics of PSTN or H.320 on the other. These gateways will perform stream transcodings, if necessary, and certain conversions of the control and set-up messages. A H.323/PSTN gateway must convert both to and from Q.931 call signalling on the PSTN side, as well as H.225.0 call signalling and H.245 call control on the H.323 side. Likewise, the H.323/H.320 gateway must perform conversion of both *Q.931-to-H.225.0* call signalling and *H.242/H.243-to-H.245* call control.

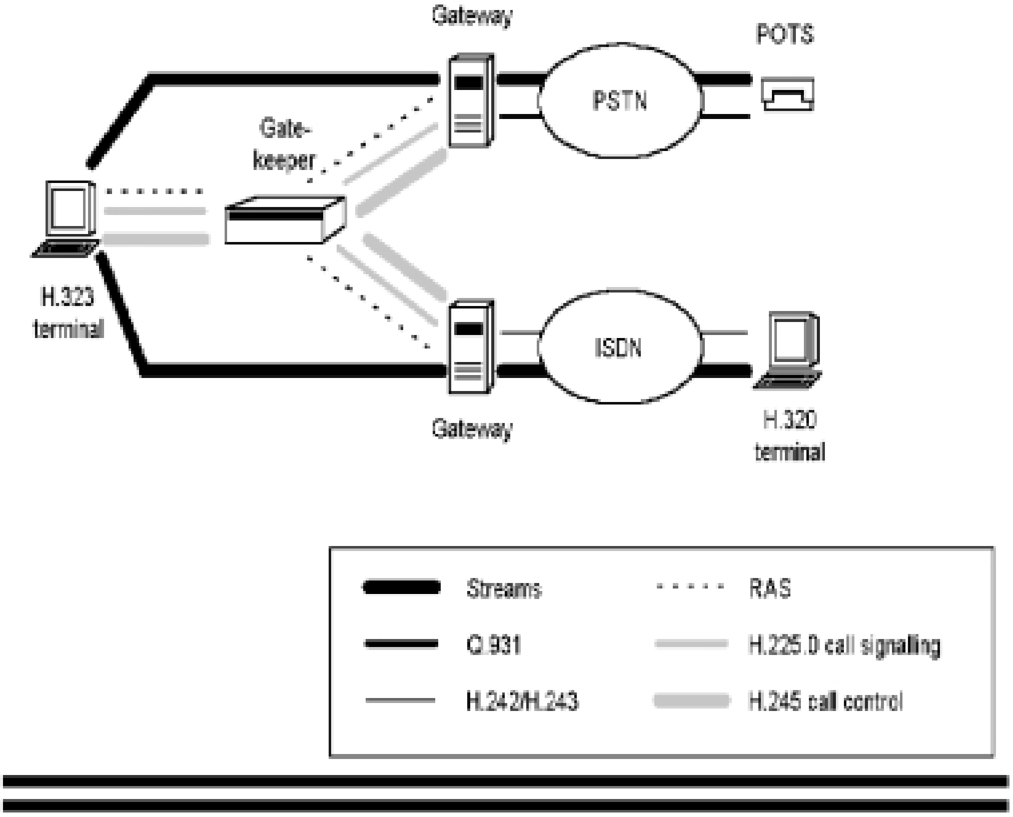


Figure 10 : *Gateway communication*

PSTN and H.320 identify terminals by E.164 numbers. The public intranet will therefore also allocate E.164 numbers to all its callable H.323 endpoints (see footnote 2). A gatekeeper maintains a table of IP-address-to-E.164 number associations of terminals within its zone.

When an external PSTN/H.320 user calls a public intranet user, she/he will be using the E.164 number of the receiver. The gateway will route the call to a public intranet gatekeeper, which in turn will find the IP address of the receiver in its tables and route the call accordingly.

When an internal public intranet user wants to call an external E.164 number (on PSTN or H.320), the gatekeeper will route the call to the proper gateway.

Gateways register at gatekeepers in normal fashion; the use of protocols, as illustrated in figure 10, is described in earlier sections.

TLS may not be supported at the PSTN and H.320 side, and the gateways must be able to establish TLS sessions with gatekeepers on their behalf, if TLS protection is to be used.

4.5.4 Firewall Communication

We will consider firewalls that protect customer LANs and central services, and also firewalls that act as gateways to external IP networks, most notably the Internet. All firewalls must register at a gatekeeper.

There are a number of types of firewalls described in the literature, with different characteristics and levels of security. The most common, generic types are perhaps:

- packet filtering firewalls (or routers)
- circuit-level proxy firewalls
- application-level proxy firewalls.

An H.323 firewall is required to handle several simultaneous connections, ephemeral ports, UDP and more, and it seems that only application-level proxies can provide the required functionality and security.

An H.323 application proxy can make use of the following in order to know which packets to deny or allow through. The call signalling is initiated on well-known TCP ports, so a proxy could allow TCP packets associated with these ports. The proxy can subsequently extract the TCP ports to be used for H.245 control channels from the call signalling messages, and allow packets to these TCP ports as needed. The proxy can similarly extract UDP ports to be used for the stream communications from the call control messages, and allow packets to these UDP ports as needed.

The firewall must also handle the RAS communication, which is initiated on well known UDP ports. The connectionless nature of UDP, how ever, makes this more difficult, although RAS UDP ports are indicated in RAS messages.

The firewall can also base its allow/deny policy on addresses, and may perhaps communicate address information with a gatekeeper in the process.

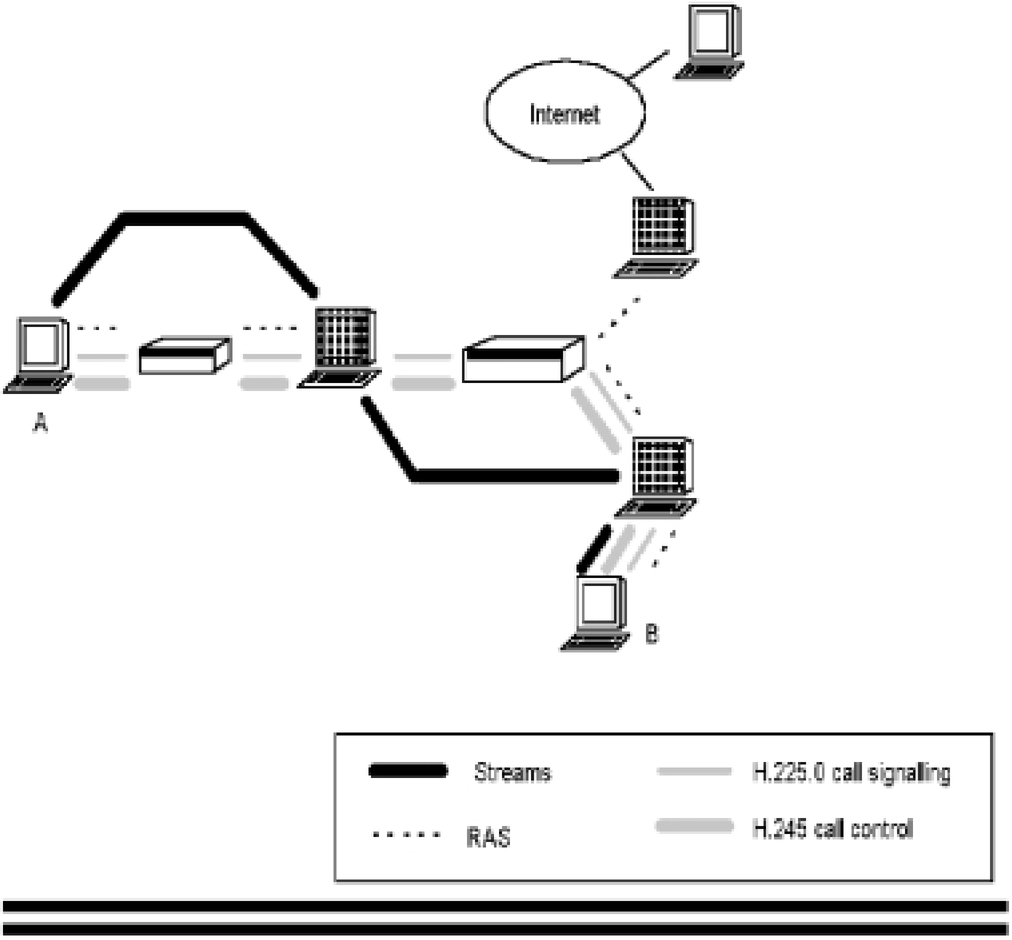


Figure 11 : *Firewall communication*

If TLS is to be used, this latter kind of firewall must terminate TLS connections and be able to establish TLS-connections on behalf of others. An end-to-end encryption of the call control channel from terminal A to terminal B in figure 8, would for instance take 5 TLS sessions. If the TLS also shall protect the call signalling, five more TLS session are needed. Note that a firewall and a gatekeeper may be connected to the same local network, and it may not be necessary to encrypt every firewall-to-gatekeeper link.

4.5.5 Other use cases

Certain use cases have been postponed for later study. These include:

- Client Login Procedures
- PSTN to PSTN Communication
- Billing and Subscription

4.6 General Assessment

As mentioned earlier, the protocols used in the use cases above are no that different. This creates a situation in which the use cases presented can be assessed in a similar manner.

H.235 with full authentication and encryption is *sufficient* to protect the communication services offered with respect to confidentiality, integrity and I&A, with a few possible exceptions. First, the RAS channels aren't encrypted, so that H.235 is vulnerable to traffic analysis of the RAS channel. It is, for instance, easy to monitor *who is calling who*, and this may be unfortunate in certain scenarios and business-critical applications (e.g., finance and law). Note that RAS is UDP based, so that TLS protection is excluded here. Secondly, due to US export restrictions, cryptographic algorithms or key lengths offered by subsystems, client software etc. may not be strong enough. These issues are discussed at length elsewhere [22] and will not be elaborated upon here.

Proper authentication is essential to the security. As mentioned earlier, H.235 authentication is based on X.509v3 certificates. The features offered by such certificates are very interesting and technology is promising. But we would like to emphasize that there are a number of risks associated with such certificates, should end-users understand their features poorly. Certificates may be spoofed if users are unaware, and clients may end up trusting false authorities if proper controls are neglected. In addition, public key technology itself is inherently vulnerable to known cipher text attacks, and encrypted messages may be compromised if end-users are ignorant. This may be a problem for large scale solutions, such as the public intranet, with many users having different levels of understanding. It is of key importance to develop good user interfaces in order to ensure correct user operations.

The H.235 set-up procedures, as described in section 4.3, are quite elaborate. Adding H.235 security to H.323 communication will slow down and complicate the set-up, and there is a risk in some applications that users may want to disable the security altogether to avoid the extra overhead.

Our comments drafted for the H.235, see Appendix B, sum up to what extent H.235 suits the public intranet system solution. *Basically, H.235 lacks an underlying trust model and ignores that the network or the service centre may perform security functions on behalf of clients.* Foremost, it is felt that an option to let the network authenticate end-users is needed within the standard.

With respect to authentication and encryption, the public intranet no doubt requires a complex system solution. The H.323 and H.235 standards themselves are fairly complex, and the total solution to these two security problems also require several other components and technologies for full completeness. It seems that there are technical solutions available to secure the various elements of the public intranet properly, such as the client login, the use of smartcards, user profile databases, certificates and so on.

Still, the complexity has security implications. Configuration may not be a straightforward task, and fully secure operation of the system may not be possible for a long period of time. A total security solution is partly based on technologies which haven't been tried out much outside research laboratories, nor with a large number of real end users[7]. In certain

[7] Public key technology and the use of X.509v3 certificates is one such example.

such cases, the operating procedures for wide-scale use are not properly developed, which leaves unanswered the question as to how well some types of solutions really scale. The point is a fundamental issue for the public intranet.

Chapter 5

Intelligent Answering Service (IAS)

Flexible, personalized communication is rapidly becoming more and more important, in business as well as personal life. Many people — especially those involved in business professions — are developing an increasing need to be able to control communication both to and from themselves throughout the day. In telecommunication terms, support for personal mobility and call completion are among the kinds of features that are being called for.

This chapter provides a general description of an **Intelligent Answering Service (IAS)** — a service concept which is founded upon *Timetable-Based Handling of Incoming H.323 Calls*. As described here, IAS is conceived as a non-standardized Supplementary Service for MMTS subscribers. It basically provides alternative handling of Incoming Calls based on the current time and a user-configurable time table.

Whereas this chapter offers a general, non-technical presentation of IAS, the following two chapters present:

- an description of the IAS user interface which has been conceived — the interface by which subscribers could configure their IAS service, and
- preliminary work upon an architectural approach for an H.323 MMTS Supplementary Service execution environment — the kind of environment required for provision and execution of services such as IAS.

5.1 Overview

In thinking about the purpose and use of IAS, the primary target users originally identified for IAS were "...business professionals requiring the capacity of being contacted — when desired — wherever they may find themselves." This group was selected based upon the communication needs existing (and continuing to develop) within this group. The selection was also based upon the fact that a large number of mobile, business professionals are already familiar with and have skills using different kinds of portable equipment (e.g., mobile phones, laptop and palmtop computers). Lastly, this target group represents an economically strong sector within the market, a group not greatly hindered by small expenses.

As a service, IAS provides subscribers with the possibility of specifying how incoming calls should be handled. In telecommunication terms, the primary functionalities delivered to the subscriber through IAS are that of *call completion* and *personal mobility.*

In this regard, it is interesting to compare IAS with UPT (Universal Personal Communication) [6]. UPT makes it possible for telecom operators to offer services such as Telenor's Alfanumber. Telenor's Alfanumber offers a personal, geographically-independent number which one can use as a *single* "external" number for all of one's telecommunication devices — whether these be one of any number of telephones at work or home, “beepers”, mobile phones, faxes, etc. The manner in which the Alfanumber is used is that the subscriber provides the telecom operator with:

- a list of numbers to associate with the Alfanumber and
- the order in which those numbers should be tried, as long as the incoming call is not answered by the callee (i.e., the subscriber).

The subscriber is also free to rearrange the order of these numbers, as well as to add or remove numbers from the list at any time.

This UPT-based service offers a certain degree of personal mobility. However, it does not guarantee call completion; that is, if the caller does not successfully reach the callee at *any* of the numbers provided, the caller fails in his attempt to contact the callee. It is precisely this issue that the IAS concept aims to address.

In the IAS service concept, the subscriber specifies how incoming calls should be handled. Like UPT, the subscriber can configure the IAS service with an ordered set of “contact points” to attempt, should the service be activated and the incoming call go unanswered. *Unlike* UPT, however, IAS offers the subscriber the possibility to associate such sets to different temporal intervals throughout the day.

Thus, IAS offers:

- a subscriber-configurable set of logical tuples (i.e., “time-based contact points”): <*from* time$_x$ *until* time$_y$, *try* <alias1, alias2,...>>[1],
- where an ‘alias’ can be:
  - an H.323 alias or
  - an email address

The IAS service is invoked when it is activated (i.e., “turned on”) and either:

- the subscriber (i.e., callee) is not logged in XOR
- the subscriber doesn’t answer the call.

With a service of this kind, it is possible to achieve an extremely high degree of personal mobility, as well as to guarantee some form of call completion. To a great extent, such personal mobility is enabled through *the characteristics of terminal mobility* — characteristics which are an inherent part of the H.323 standard. More explicitly, when a terminal is connected — even temporarily — to an H.323 network (or a network which includes an appropriate gateway to an H.323 network), one or more gatekeepers within the H.323 network store information as to how to direct incoming calls to that *terminal*. This property of H.323 achieves one kind of terminal mobility.

To achieve *personal mobility* in this context, Ericsson (among others) is developing approaches and techniques by which to associate a person with a terminal. With such an association, calls can be directed to the right *person*, regardless of which terminal they are using when logged on.

[1] In this project, we limited the cardinality of the set of aliases to a maximum of two, in order to retain focus upon more relevant issues. This “limitation” is *not* the result of any technical hinders.

Concerning call completion[2], the purpose in having IAS allow the specification of an email address is the following: Should a subscriber have the IAS service activated and an incoming call go unanswered, the caller can then be prompted as to whether he wishes to compose a multimedia message to be sent to the callee (subscriber). Here, the IAS service can seamlessly move from one communication modality (i.e., the caller's attempt to establish a *synchronous* contact with the callee), to another communication modality (i.e., the opportunity for the caller to compose an *asynchronous* multimedia message to be sent to the callee).

Of course, the kind of multimedia message a caller can compose will be limited by the characteristics of the caller's terminal. If the caller is using a PC as a terminal during the call attempt, the caller may have the opportunity to record and send an audiovisual message, along with other documents, images, film clips, etc. With only a common telephone, the caller could be offered the possibility of sending e.g., a voice mail to the caller.

What should be understood from the discussion above is that IAS subscribers have the possibility of controlling both *when* they are contacted, *whether* they are contacted and *how* they are contacted (i.e., whether they are contacted in a synchronous or asynchronous manner).

An example below will help illustrate the manner in which the IAS service is used by a subscriber, and how the service operates.

5.2 Use and Operation of IAS

The scenario which follows exemplifies the use and operation of IAS. It is based upon a hypothetical person named Bill, whose current work schedule and daily habits are described briefly below in storybook fashion. Figure 12 aims to illustrate the topology of Bill's travels throughout one of his work-days.

[2] Perhaps "contact completion" is a more accurate term in this context.

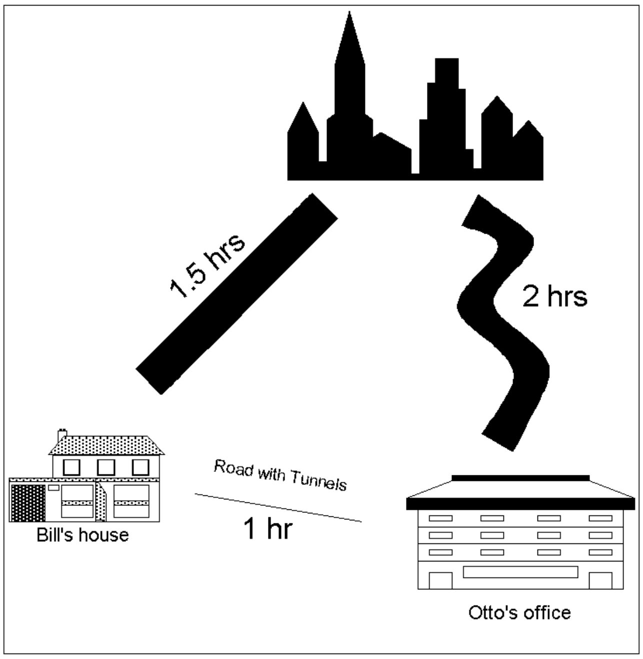


Figure 12 : *The topology of Bill's travels*

5.2.1 The life of Bill

Bill is a clever guy, and he works upon a number of different kinds of tasks for his employer. Some of these tasks last for less than a day, while others can stretch out as long as two weeks or more. When faced with longer tasks, Bill tries to create a daily schedule which is as regular as possible. For the most part, however, BIll feels like he's always on the run.

Since Bill works on so many different kinds of tasks, people are always trying to get in touch with him throughout the day. This is one of the reasons Bill tries to keep some regular kind of office hours. Still, he's often called out of the office on unexpected business.

At home, Bill also tries to keep a somewhat regular schedule. For instance, he usually leaves the house every day sometime between 7:30 and 8:30, and tries to be home sometime around 17:00. Part of Bill's "regular" schedule is that he drives in directly to work each day, to pick up any messages left for him there.

Due to his hectic life, Bill has developed the habit of turning on his GSM phone when preparing to drive to work. On occasion, he receives calls which mean that he must drive directly to some customer, rather than heading for the office.

Bill usually arrives at work some time between 8:45 and 10:00. He has a habit of logging into his H.323 terminal each day as soon as he gets in. Each day, he has a regular meeting with the boss from 11:00 to 12:00, in order to discuss business priorities. Bill *never* wants to be disturbed while in that meeting.

After the meeting with the boss, Bill leaves for his afternoon work together with Otto; his work with Otto is just one of his ongoing, prioritized tasks. Since the road to Otto's office cuts through some mountainous terrain, Bill doesn't bother with his GSM phone on the way; the chances of getting a call through are always poor.

Bill usually arrives at Otto's place around 14:00. Bill and Otto work together at Otto's H.323 terminal using Otto's account and password; they usually work together from about 14:00 -16:00.

Some of Bill's closest colleagues know that Bill is at Otto's place using Otto's terminal. For this reason, Otto sometimes gets emails in his mailbox which are actually intended for Bill. If Bill has already left Otto's place, Otto always forwards the emails on to Bill.

On the hour's drive home from Otto's office, Bill always relaxes. He usually turns on some music and turns off his GSM phone. He arrives home about 17:00 each day, sometimes doing some work there during the evening at his H.323 terminal.

5.2.2 Bill's IAS configuration

Given his set of prioritized tasks, along with the times and places Bill has decided to carry out his work, Bill has chosen to configure his IAS service as illustrated in figure 13.

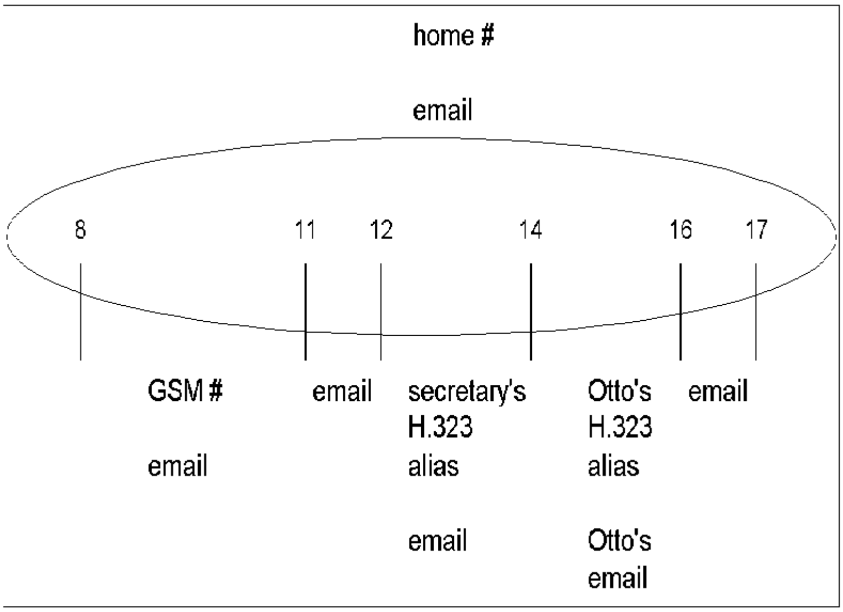


Figure 13 : *Bill's IAS configuration for the day*

5.2.3 Effects of the configuration

It is important to remember that IAS only affects calls made to a subscriber's H.323 number when the subscriber has the service activated. When the service is not activated, incoming calls to the subscriber behave in the standard fashion[3].

When activated, Bill's IAS configuration can be understood to suit his situation in the following way:

[3] See sections 4.3.4 and 4.3.5, for further details.

- from 8-11 am, Bill is either preparing to leave for work, on the way to the office or perhaps even at the office
    - if he’s arrived and has logged in at the office[4], calls made to his H.323 number ring first at the terminal he’s logged into;
    - if he fails to answer this call, or simply hasn’t arrived and logged in yet, his IAS service redirects the call to his GSM number;
    - if he fails to answer this call at the GSM number, IAS behaves so as to prompt the caller with the opportunity to create a multimedia message for Bill.
- from 11-12, Bill is with the boss and doesn’t want *any* interruptions — he doesn’t even wish that the secretary potentially misjudge the urgency of a call and thereby interrupt the meeting; once again:
    - if Bill is logged into and associated with a terminal — calls made to Bill’s H.323 number ring at that terminal:
    - if he fails to answer this call, Bill’s IAS service behaves so as to prompt the caller with the opportunity to create a multimedia message for Bill.

It is left as an exercise to the reader:

- to further study Bill’s daily schedule and IAS configuration
- to discern the service’s behavior at different times and conditions during his day and
- to understand why Bill’s IAS configuration suits his current work situation.

5.3 IAS Configuration Characteristics

Perhaps the most immediate observation about IAS configuration is that it is based upon a rotating 24-hour schedule, rather than a weekly or monthly “calendar”. This decision was based upon judgements about the intended target group; in particular, it is judged that the majority of the intended target group have work-days and schedules which are highly dynamic in nature. In other words, such persons may find that determining a complete schedule more than three or four days in advance is not entirely possible.

The decision about basing the service upon a 24-hour timetable — along with other judgements about the conditions of the intended target group — has implied several other IAS configuration requirements. These include:

- the service must be *directly* configurable by the subscriber (unlike UPT)
- configuration and re-configuration of IAS settings must be simple and fast.

To achieve this latter point, the IAS configuration interface offers user-creation of “presets” or “templates”, along with rapid template selection. Use of template overrides has also been conceived as a manner by which to achieve rapid configuration. These topics are the subject of the chapter which follows.

[4] Here, “logged in” intends to mean *logged into and associated with* a terminal which is connected — even temporarily — to an H.323 network (or a network which includes an appropriate gateway to an H.323 network),

5.4 Other Issues

As will be described in chapter 7, realization of the IAS service requires a gatekeeper-routed call control model. Other issues concerning IAS behavior and functionality have been identified and, as yet, remain open. These include:

- the possibility for a subscriber to define groups and group-roles/relations (e.g., such that calls can be redirected to *any* secretary within a secretarial pool)
- when a call is being redirected to someone other than the initially called subscriber, the caller should be notified as to nature of the redirection (i.e., notified of either the name of the person and/or their role/relation with respect to the original subscriber)
- third-party privacy: ensuring that a subscriber cannot redirect calls to *anyone* they wish
- device scaling: depending on the device, a subset of the configuration user interface should be available
- refining seamlessness between call conferencing and message modalities, (e.g., including the definition of “companion functions” for message composition, such as functions for scanning and reading email; initiating calls directly from a message reader, etc.)

Chapter 6

User Interface Specification for Intelligent Answering Service

6.1 Introduction

The purpose of IAS has been explained in chapter 5. The focus of this section is to present results from the project's user interface design activity. The purpose of this activity has been to specify a simple, fast and easy-to-use user interface concept for Service Configuration in general, and the Intelligent Answering Service (IAS) in particular. For population and use of settings within this user interface, a service-independent data configuration interface is also required[1]; *that* interface is not discussed in this section.

The user interface design has been developed with a *desktop PC* in mind as terminal type, and a *mouse* for interaction control. However, the concepts should be easily transferable to smaller devices, such as mobile telephones and PDAs, with other kinds of interaction controls.

The approach taken is *task-oriented*, where only the user and the task at hand guide the design of the interface [9]. Such an approach is useful when the task is limited and clearly defined, which is the case for configuration of IAS. This is in contrast to the traditional *object-action* approach (i.e., choose an object, then perform an action on that object), an approach which forces the user to concentrate too much on how to manipulate the user interface itself [7]. For a general introduction to state-of-the-art user interface design, see [10].

The concepts have been developed using brainstorming and *think tank* methodology, and have been worked out gradually within a small team. The prototyping method was large blackboard drawings, continuously refined. No outside evaluation and testing of the ideas has yet been performed.

Due to time constraints, no attempt to integrate IAS and traditional *calendar tools* has yet been made. However, this should be investigated, as the similarities are striking. Other corresponding solutions and functions which affect IAS (e.g., from PABX, GSM) should also be considered.

The IAS user interface is one subpart within a wider conceptual interface framework for integrated H.323 communication services. Other “peer services” to IAS could include Call Making, Call Reception, Message Sending, Message Receiving etc. Therefore, the interface framework is first described in section 6.2, and the details of the IAS user interface are presented in the following sections. Of these descriptions, section 6.3 lists some of the design premises chosen for IAS.

[1] A service-independent data configuration interface is required between the Client and a persistent store for such information; see section 7.5 for further details.

Sections 6.4 and 6.5 contain the actual user interface specification, divided into the *graphical presentation* (traditionally referred to as the user interface), and *system services* (or functionality). This division was adopted, in modified form, from certain usability engineering principles employed at Ericsson. Lastly, section 6.6 discusses some ideas for implementation of the concepts described.

6.2 Service Interface Framework

A complete conceptual interface framework for H.323 communication services will possibly include services such as Call Making, Call Reception, Message Sending, Message Receiving etc. At present, these parts of the conceptual framework are not integrated with respect to common user interface and interoperability. A suggestion for such an integrated user interface framework is discussed below.

The framework is logically divided into three main parts: *Service Selection* (upper part), *Service Configuration* (middle part) and *Service Activation/Deactivation* (lower part), as illustrated in figure 14 below. Service Selection and Service Activation/Deactivation are directly coupled, and those two will always be present in the interface. They will never change appearance, apart from colors and “inner” icons. Service Configuration is the more dynamic part, and its appearance is *service-specific*, i.e. dependent upon the service selected.

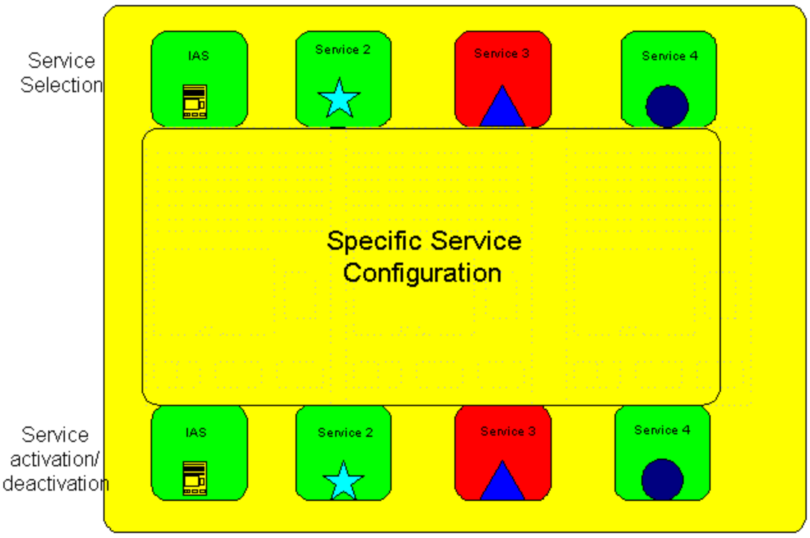


Figure 14 : *Framework: division of user interface*

The main idea with respect to graphical presentation is to use clear labelling, and distinguishable background images for each service in the framework [8]. The user should never be in doubt as to which service is being configured. Also, a change in the service activation status should lead to an explicit change in colors (e.g., green = activated, red = deactivated). The user should never be in doubt as to whether a service is activated or not. This is another main feature.

Regarding functionality, one service selection button, and one service activation/deactivation button, will be found for each service. These buttons will be placed directly opposite one another in the interface. This is done to enable easy service access, since activation and deactivation are regarded as the two most fundamental operations belonging to a service. Though a two-button solution is perhaps not optimal, it is the best suggestion which has been agreed upon within the design team. It has been judged that a single-button approach — to be used for both service selection *and* activation/deactivation — is too complex and not preferable.

In Service Selection, IAS will be only one of the possible services to select. Likewise, in Service Activation/Deactivation the on/off state of IAS will be only one of many.

Turning a service on or off is simply done by pressing the corresponding service button in the Service Activation/Deactivation part. This can be done at any time during user interaction, independent of whether the service is being configured or not. The result is not only a change in color (and possibly appearance) of that button; a change also occurs in the background image in the Service Configuration area (when that service is the one presently selected for configuration).

Configuring a service is done by first pressing the corresponding service button in the Service Selection part. A service-specific interface will then be present in the central part of the framework (i.e., the Service Configuration area). An example of this kind is given for the IAS service (see section 6.4). When a service is selected, the color and background image within the Service Configuration area will be the same as that of the associated service selection button. This design characteristic will enhance service recognition.

In addition to service configuration, retrieval of Service Statistics will typically be another generic function to be performed for each service in the framework. However, the user interface for such a function has not yet been considered.

Some future extensions to the framework were also discussed during the interface design activity. One of these was that the framework should be designed to allow selection of any one of a *very* large number of services — not only limited to four, as might be assumed from figure 14 above. Such a requirement could necessitate the introduction of some form of *scrolling* facility. Alternatively, services could be grouped according to some set of higher-level abstractions (e.g., “personal mobility”), and made available through a *shallow*, yet nested selection scheme. In the greatest extreme, the framework could consist of only *one* service having different features, perhaps ordered in sublayers.

The framework should also allow for special services designed by the *user* himself, not only by service creators and providers such as Ericsson.

6.3 Design Premises

When working upon the functionality of IAS and its interface design, a number of design premises were established. These premises were not based in technical limitations nor user studies; instead, they were established in order that these activities retain focus upon more relevant issues. Based on the functionality desired for IAS, the initial design assumptions included:

- The basis for service configuration is a 24 hour *timetable*, with non-overlapping *intervals* (or *timetable intervals*) of length n_i hours, where n_i=1,2,..etc. Thus, the maximum number of allowed intervals is 24 (i.e., n_i=1 for i=1,..., 24).
- Intervals crossing midnight are allowed.
- Intervals either contain content, or they do not. Intervals which do not contain content are called *empty intervals* (or *default intervals*).
- The *content* within an interval consists of a *time interval* and *contact content*. Recalling from section 5.1, this content can be logically represented in the form: <*from* time_x *until* time_y, *try* <alias1, alias2>>. Here, the individual aliases in the tuple are sometimes called *contact aliases*.
- The *contact content* of any timetable interval is either an H.323 alias, a Message Application Identifier or both. Note that whenever a Message Application Identifier is included as a contact alias, it appears lattermost amongst those aliases.
- An H.323 alias is either *dynamic* (i.e., belonging to a user: H.323 address, UPT number, etc.), or *static* (i.e., belonging to an endpoint: E.164 number, IP address, etc.) For the time being only an E.164 number should be allowed to be specified[2].
- A Message Application Identifier is an *email* address, specifying the address to which an asynchronous multimedia message should be directed.
- *Default contact content* is defined for empty (default) intervals; this default content is of Message Application Identifier type.
- Using only two contact aliases, four contact content *combinations* of H.323 alias and Message Application Identifier are possible[3], of which three are valid. Either both are specified (with the Message Application Identifier type appearing lattermost), or one of the two, or none (invalid).
- It is the *user*, not the service manager (i.e., Gatekeeper), who is responsible for ensuring that the content of a timetable interval is meaningful.

Additionally, a number of potential *hinders* to actual use were identified, which one should be aware of when designing the interface. These include:

- interfaces which are too complex are not used
- users may lack discipline when configuring the service
- employing a time axis which is too brief (e.g., only 24 hours) may hinder actual use of the service
- recurring situations for the user (i.e., special treatment of weekdays and/or task days (e.g., travel)) must be accommodated for by the service.

[2] This was one interface design restriction, established solely in order to enable better focus.

[3] This limitation to four combinations was the result of another restriction for simplicity: that is, only two contact aliases were allowed in the logical tuple, see section 5.1 for further details.

6.4 IAS' Graphical Interface

The representation of IAS' timetable is the most important issue when it comes to the service's graphical presentation.

Before the Configuration functionality is presented, some logical definitions are necessary for easier comprehension:

- An *ordinate* is a generic object. A *populated ordinate* has associated with it a user-defined name, a time interval and valid contact content (i.e., a valid set of contact aliases). In other words, a populated ordinate can logically be defined as <name, <*from* time_x *until* time_y, *try* <alias1, alias2>>>.
- Each populated ordinate can have multiple *instances*, i.e., there is a 1:n relationship between populated ordinates and their instances.
- An *empty* ordinate has no specified time interval and no contact content.
- A *block* (or *interval block*) is a graphical representation of (any kind of) ordinate.
- A *preset* is a generic object, associated with a specific timetable configuration; it consists of a name, and zero or more ordinate instances (note: an *empty* preset is a preset having zero ordinate instances).
- The current timetable is seen as one preset instance in a special, *current*, state.

The most immediate problem when trying to develop a graphical representation for the timetable has been how to achieve quick visible overview of both ordinate count and ordinate duration. Several ideas were discussed, which either considered a circular or a linear representation of time. The originally proposed and often used "24 hour clock" was abandoned because of its non-existing equivalent in real life, and because unused intervals took up too much space.

Instead, it was decided to represent each ordinate instance by equally-sized "blocks" along a linear time axis — an axis which displays the next 24 hours ahead in time. However, as each block would span a time period of arbitrary length, the time axis only indicates the relative position in time between the blocks (ordinates). A default interval is presented as an empty interval, and not represented by any block at all. The chosen representation is shown in figure 15. Intentionally, as can be seen, no strict metaphor for timetable representation was considered: it only resembles a "shelf of hats".

This interface characteristic facilitates quick identification of ordinate count, at the expense of ordinate duration. To help indicate the latter, an arrow of length proportional to the ordinate duration is suggested added to each block. Whether this is a good solution remains to be seen.

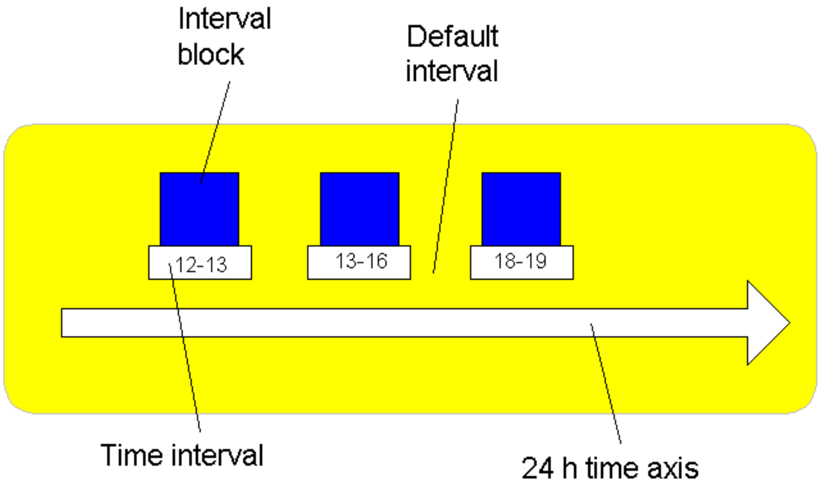


Figure 15 : *Timetable representation*

Given this timetable representation, the specific user interface for IAS configuration is now discussed. Logically, the interface is divided into four parts: a *status* area, an *ordinate* area, a *preset* area, and an *editing* area. This division is illustrated in figure 16. The colors and symbols used should not be considered as final.

At all times, the upper middle status area shows the current timetable configuration (using the representation illustrated above). At the left-most end of the time axis, a “live” clock displays the present time. As time passes, the blocks in the current timetable drift (or slide) from right to left.

The ordinate area at the left is for user-populated ordinates, each representing some logical information tuple of the form: <name, <*from* time$_x$ *until* time$_y$, *try* <alias1, alias2>>>. For each such ordinate, the user provides a descriptive name (e.g., Secretary), and a time period indicating the beginning and end of the ordinate. Additionally, each ordinate contains valid contact content information, which is however only shown when editing. The ordinate area also contains an empty (default) ordinate, having an unspecified time period and content. This default ordinate is used when creating new ordinates.

The preset area at the right is for user-specified presets, representing different timetable configurations. Each preset has a descriptive name (e.g., Travel), and displays a small-scale image of the timetable, containing only the number of specified ordinates (blocks). There is also an empty (default) preset, which is used when creating new presets (and when clearing the current timetable configuration).

The editing area at the lower middle is where all editing of the current timetable, presets and ordinates take place. By default, the area is empty. The complete editing semantics are described in section 6.5.1, along with all the other configuration operations allowed in the service.

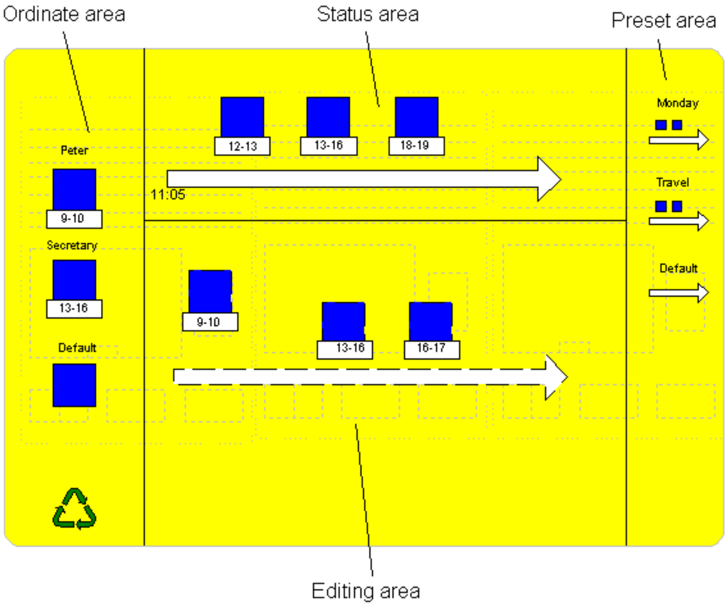


Figure 16 : *IAS: division of user interface*

In addition to these four areas, the IAS user interface contains e.g. a trash can (recycle symbol) in the lower left corner, which is used for deletion of presets and ordinates. The decision as to whether there should be an initial collection of ordinates and presets at start-up and what these should eventually contain, has not been finalized.

6.5 Functionality: system services

As already indicated in section 6.2, the generic functions associated with a typical H.323 service are:

- Configuration
- Activation/deactivation
- Statistics

How these functions, particularly Configuration, are realised in the IAS user interface is described in this section.

6.5.1 Configuration

As the work with the IAS functionality is still only on a conceptual level, the complete configuration semantics have yet to be finalized. However, *drag-and-drop* functionality is seen as the fundamental user interaction form, since it is believed to be the most natural approach for this service configuration task. Thus, there will be no pull-down menus nor no pop-up menus, which are considered to be more complex to manipulate.

The suggested configuration operations are presented below. Again, figure 16 is used as reference.

Operations on Current Timetable:

- **Replace current timetable with preset**: Grab preset (from preset area) with mouse, drag it to status area, and drop it; this sets the current timetable.
- **Clear current timetable**: Grab empty (default) preset with mouse, drag it to status area, and drop it. This operation is similar to “replace current timetable” though here, an *empty* preset is grabbed and dropped.
- **Edit current timetable**: Grab current timetable (from status area) with mouse, drag it to editing area, and drop it. Modify, add or remove ordinate instances (see Operations on Preset Content below). Grab edited timetable (from editing area), drag it back to status area, and drop it; this sets the current timetable.

Operations on Presets:

- **Edit existing preset**: Grab preset (from preset area) with mouse, drag it to editing area, and drop it. Modify, add or remove ordinate instances (see Operations on Preset Content below). Grab edited preset (from editing area), drag it back to preset area and drop it. The edited preset will automatically replace the previously saved one, i.e. analog to *save*.
- **Create new preset**: Grab empty (default) preset with mouse, drag it to editing area, and drop it. Specify preset name, and modify, add or remove ordinate instances (see Operations on Preset Content below). Grab edited preset (from editing area), drag it back to preset area, and drop it. This new preset will automatically be saved, i.e. analog to *save as*.
- **Delete existing preset**: Grab preset (from preset area) with mouse, drag it to trash can, and drop it. *NB! Deletion of empty preset is not allowed.*

Operations on Preset Content (carried out in editing area):

- **Specify preset name** (only allowed when creating new presets): Touch text field “above” preset with mouse, write name in that field, and release it.
- **Modify ordinate instance** (in preset): Touch instance with mouse, modify contact content or time interval (see Operations on Ordinate content below), and release instance.
- **Add ordinate instance** (to preset): Grab ordinate (from ordinate area) with mouse, drag it to editing area, and drop it. Modify ordinate instance (see above).
- **Remove ordinate instance** (from preset): Grab instance with mouse, drag it to trash can, and drop it.

Operations on Ordinates:

- **Edit existing ordinate**: Grab existing ordinate (from ordinate area) with mouse, drag it to editing area, and drop it. Modify contact content or time interval (see Operations on Ordinate Content below). Grab edited ordinate (from editing area), drag it back to ordinate area, and drop it. The edited ordinate will automatically replace the previously saved one, i.e. analog to *save*.
- **Create new ordinate**: Grab empty (default) ordinate with mouse, drag it to editing area, and drop it. Specify ordinate name, and modify contact content or time interval (see Operations on Ordinate Content below). Grab edited ordinate

(from editing area), drag it back to ordinate area, and drop it. The new ordinate will automatically be saved, i.e. analog to *save as*.

- **Delete existing ordinate**: Grab existing ordinate (from ordinate area) with mouse, drag it to trash can, and drop it. *NB! Deletion of empty ordinate is not allowed.*

Operations on Ordinate Content (carried out in editing area, see figure 17):

- **Specify ordinate name** (only allowed when creating new ordinates): Touch text field "above" ordinate with mouse, write name in that field, and release it.
- **Modify contact content**: Edit the two text fields (for H.323 alias and Message Application Identifier, respectively), which will appear as ordinate is dropped or touched in editing area. The exact form is not yet specified.
- **Modify time interval**: Modify beginning of interval by touching left up/down button, and modify end of interval by touching right up/down button.

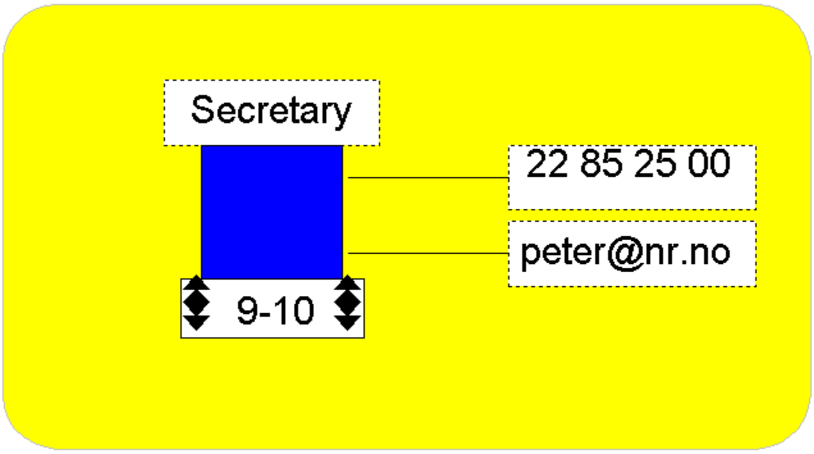


Figure 17 : *Operations on Ordinate content*

Many details regarding the operations have not yet been finalized. Also, a number of functionality issues are still outstanding. These include:

- modifying the default content for unspecified, empty intervals (note: having a separate area for editing of default value is one alternative)
- for a given service, whether the service configuration functionality offered in activated vs. deactivated mode should be identical (note: having the current timetable grayed out in deactivated mode is one alternative)
- changing the preset name or ordinate name once it is set
- when to commit changes made in the user's service data: per data element vs. "all-at-once"
- how to commit changes made in the user's service data: implicit vs. explicit commit

- logically coupling the presets with the concept of personal roles / situations e.g., “business trip”, “day off”, etc. (note: the name assigned to the preset by the user is one kind of logical coupling)
- logically distinguishing between an *update* (editing of preset) and a *“one-timer”* (editing of current timetable) when configuring
- detecting and avoiding time interval conflicts / inconsistencies when modifying ordinate instances: both within an ordinate and between ordinates
- establishing general error handling mechanisms for configuration.

6.5.2 Activation / deactivation

This functionality is part of the wider framework for H.323 communication services, and is described in section 6.2. As previously explained, activation/deactivation of IAS can be performed independently of service selection, and carried out at any time during user interaction.

6.5.3 Statistics

This functionality has not yet been investigated, neither in regard to what kind of statistics should be computed, where they are computed (client vs. server), nor what the user interface should look like.

6.6 Implementation

The implementation of IAS should be based on Java Applets within a web browser environment. This will make the service easily available on different platforms, and from different terminal types. It is suggested that a visual programming tool (e.g., Symantec Cafe, etc.) be employed for rapid prototyping, although implementation of the drag-and-drop functionality will require more explicit Java programming than is traditionally offered by such tools.

Since interface prototyping activities fell outside of the scope of the project’s budget, further details regarding interface implementation remain yet to be considered.

Chapter 7

Architectural Approach for an H.323 MMTS Supplementary Service Execution Environment

7.1 Introduction

In this context a Supplementary Service is defined as an application independent feature which adds value to the MMTS. Chapter 5 described a supplementary service (IAS) — a service which is independent of the H.323 application which runs on the client. This is in accordance with the definition given in [11] which states:

> *"A supplementary service modifies or supplements a basic telecommunication service. Consequently, it cannot be offered to a customer as a stand-alone service. It must be offered together with or in association with a basic telecommunication service."*

The architecture for an H.323 *Supplementary Service Execution Environment* (**SSEE**) is based upon a structure consisting of two layers: the *Call* layer (**C-layer**) and the *Model/Controller* layer (**M/C-layer**).

The C-layer is closely related to the H.323 control plane, as defined in [12] and realized within the H.323 entities *Gatekeeper* (**GK**) and terminals. The M/C-layer is an abstraction of the details encapsulated in the C-layer, providing the logic of the supplementary services with a *Supplementary Service Access Point* (**SSAP**). The two layers interact through a *Supplementary Service Independent Protocol* (**SSIP**), and the supplementary services themselves utilize the M/C-layer. The GK (which is an optional entity in H.323 systems) is mandatory in this architecture, and should operate in *Gatekeeper Routed Call Signaling* mode as specified in [12].

Figure 18 gives an informal view of the two layers and the H.323 entities' positioning.

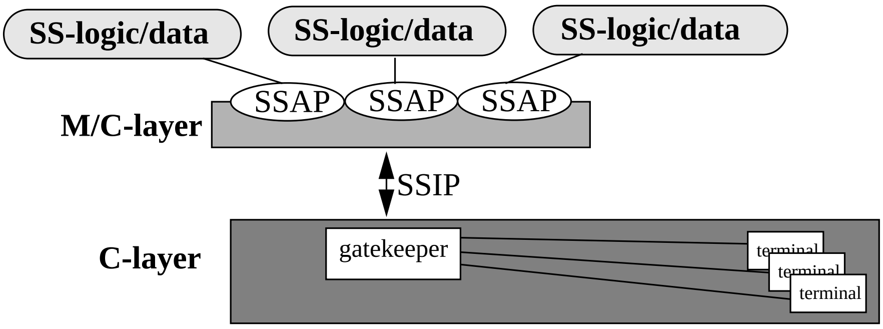


Figure 18 : *Informal layered view*

An important objective for the architecture is that it should handle both standardized supplementary services and possible proprietary supplementary services.

The architecture is based on the assumption that the design-and-deploy frequency of new services will be quite high.

As indicated in this chapter's title, the following description is an *approach* to the design and implementation of an architecture. Thus, there remain unresolved and open issues which are not covered in this report.

7.2 C-Layer

The C-layer contains the protocol machinery for the RAS channel and call signaling channel as defined in [12] and [13]. Thus, this layer is capable of performing the procedures for call admission and call setup and release and is therefore complete with regard to basic call processing.

In addition, this layer should be capable of detecting whether a supplementary service should be invoked, based on factors such as the state of the call, signaling events, signal information elements and the profile of the calling or the called user. This mechanism is here called *Triggering of Supplementary Service* (**ToSS**). Such a characteristic implies that signaling might be intercepted in the C-layer, and that further processing of the call is controlled by the logic and data of a supplementary service (via the M/C-layer, using a provided SSAP and the SSIP). The C-layer will act upon the operations sent in the SSIP by extending the basic call processing with the transitions necessary to perform the task of the supplementary service.

To accomplish the ToSS and the subsequent change of call control, a *Triangular Call Model* (**TCM**) is established as shown in figure 19.

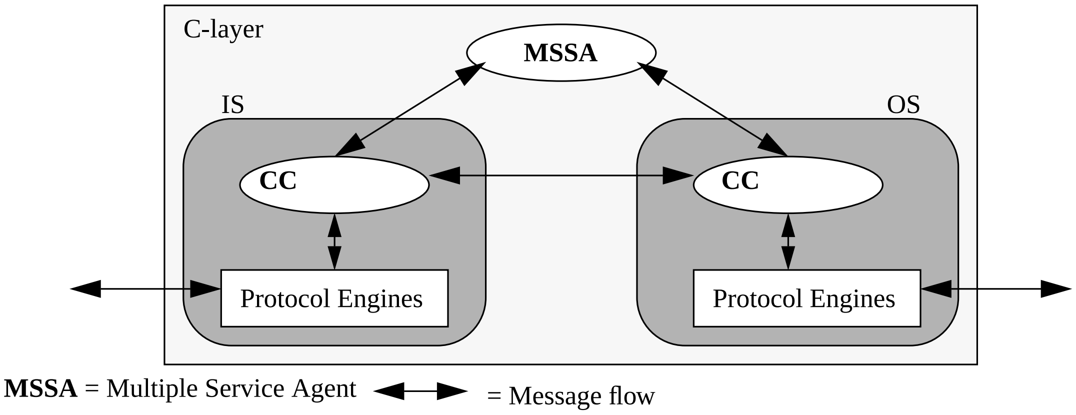


Figure 19 : *Informal C-layer model*

The TCM is based on viewing a call as having two halves: one *incoming side* (**IS**) which represents the source of the call, and one *outgoing side* (**OS**) which represents the destination of the call. Each *Half Call* (**HC**) will be handled by a *Call Control* (**CC**) that interacts with the protocol engines, the other side's CC and possibly a *Multiple Supplementary*

Service Agent (**MSSA**). Each CC is an *Extended Finite State Machine* (**EFSM**) which acts upon the primitives received.

The MSSA will be a passive yet resident part of the call model, as long as no supplementary services are involved in the call (i.e. it will not influence the performance of the basic call processing). If ToSS occurs, the MSSA will act as an agent for the triggered supplementary service (i.e. it will effectuate the operations received over the SSIP during the supplementary service session). The MSSA should be able to handle multiple supplementary service sessions simultaneously as indicated in the name, which again implies that the MSSA should handle possible interactions between supplementary services simultaneously involved in the call.

The presence of the OS depends on the state of the call; that is, a ToSS might occur during the admission phase (RAS signaling) of the IS where no OS is yet established.

7.2.1 Points of Invocation

The ToSS implies that a well defined set of possible points in the basic call processing is used for initial attachments of supplementary service execution. These points are named *Points of Invocation* (**POI**). There will be POIs for both IS and OS of a call; in other words, both originating and terminating supplementary services can be handled.

When a POI is encountered in the basic call processing, the POIs status will be checked as to whether it is **armed**. A POI could be armed based upon a user profile (i.e. a subscribed user has activated one or more supplementary services) or based upon the general availability of such services (i.e, certain kinds of services may not be available through certain connections). Figure 20 illustrates the POI feature in the C-layer.

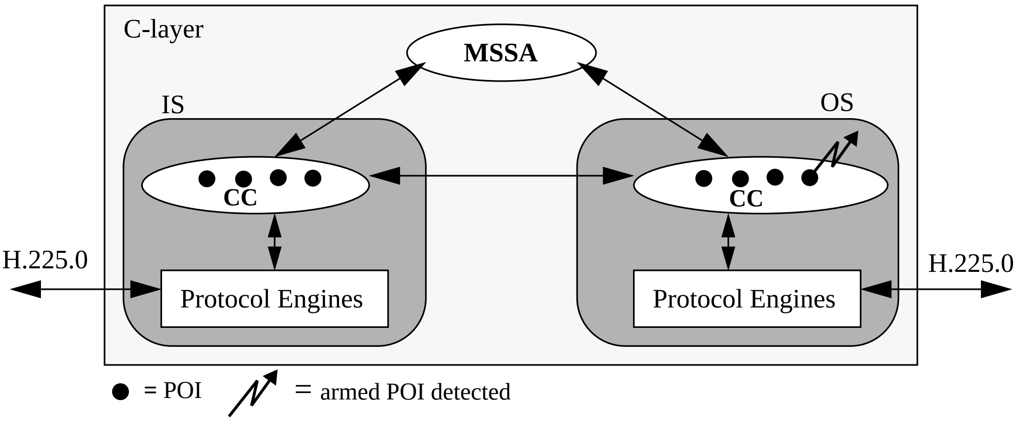


Figure 20 : *POIs in the C-layer*

For each POI, there will be a 1-1 relationship with a possible originating or terminating supplementary service. Thus, a subscriber of a supplementary service like *Call Transfer* (**CT**) [14] will have the CT attached to a POI that relates to the reception of the invoking component for CT in the Facility message.

When an armed POI is encountered, the MSSA will be involved in the call (i.e. the normal message flow between the CCs will be routed via the MSSA). Figure 21 illustrates this mechanism without going into the specific messages involved prior to the POI encounter.

The set of messages between the CC and MSSA should be a superset of the set of messages between the CCs.

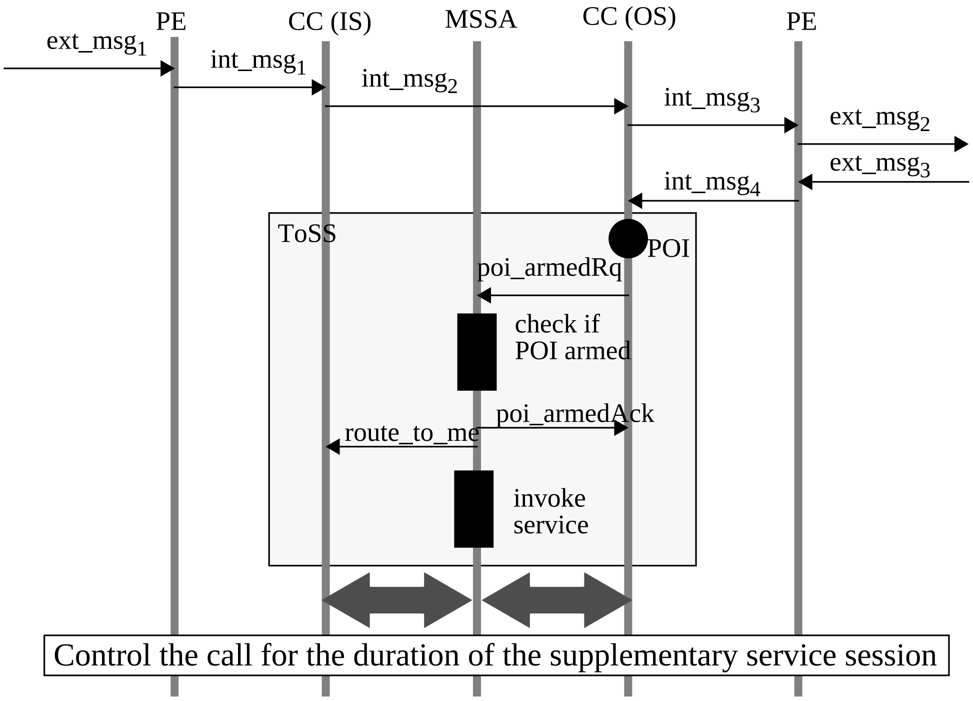


Figure 21 : *ToSS with IS and OS*

The MSSA-CC flow should be the same as the CC-CC flow with regard to the H.323 control plane. In addition, a message sequence such as that indicated in the shaded frame (ToSS part) in figure 21 must exist. This sequence checks whether the POI is armed, and if it is, the MSSA requests OS and IS to start routing messages via the MSSA. Note that figure 21 illustrates ToSS for OS (i.e. a terminating supplementary service). The same mechanism should apply for IS (i.e. originating supplementary services).

A POI can be more formally defined when viewing the CC as an EFSM. Two different categories of POIs have been identified:

- If in a state **S** message **M** is received
- If in a state **S** message **M** is received with one or more information elements $\mathbf{IE_1}$,$\mathbf{IE_2}$,...,$\mathbf{IE_k}$ set to values $\mathbf{v_1}$,$\mathbf{v_2}$,...,$\mathbf{v_k}$ within some value range of the $\mathbf{IE_j}$ type.

Thus, for the CC EFSM there will be a set of states {$\mathbf{S_1}$,$\mathbf{S_2}$,...} where for each $\mathbf{S}_i$ in this set, there will be at least one input message **M** that either with or without information elements, will be a POI.

In addition to the POIs, there will also be **PORs** (Point of Return). A POR is a point in the MSSA where a supplementary service session ends. When a POR is met for a supplementary service session, the MSSA must check whether this is the last supplementary service involved in the call. If it is, the normal flow between the CCs must be re-established. This is visualized in figure 22.

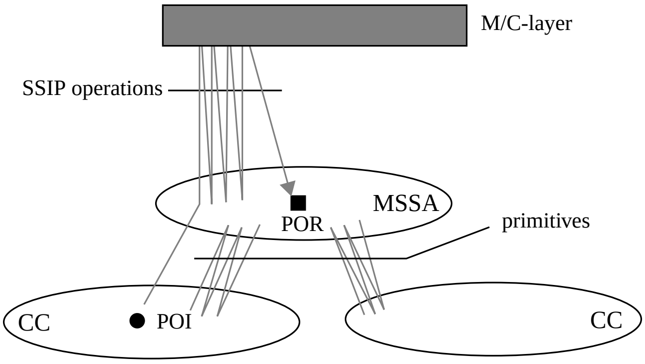


Figure 22 : *POI and POR for a supplementary service*

7.2.2 The MSSA

As previously described, the MSSA acts on behalf of supplementary service logic via the M/C-layer and the SSIP, and should be able to handle simultaneous supplementary service sessions. This implies that the internals of the MSSA must be able to handle potential interactions between the different service logics involved. This is an area of complexity and is out of the scope of this chapter, thus an simplified solution is sketched.

In figure 23, the MSSA internals are illustrated. Several *Supplementary Service Agents* (**SSA**) may coexist at the same time. To exemplify such a scenario, one could think of a sequence of ToSS for the IAS.

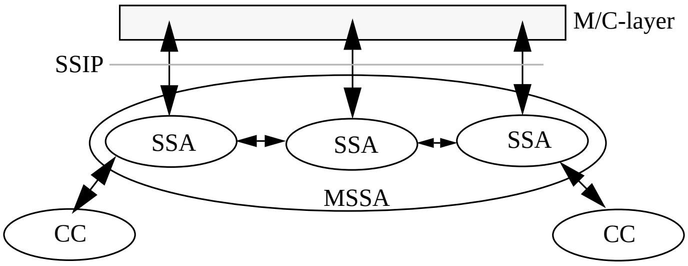


Figure 23 : *MSSA internals*

7.3 Supplementary Service Independent Protocol

As previously described, the C-layer and the M/C-layer interact through the SSIP. The intention of having a SSIP is to keep the C-layer as stable and call-related as possible. In the worst case, new supplementary services should only imply a *very* limited degree of C-layer redesign; such a situation could arise, for instance, when new POIs are being introduced.

Therefore, the C-layer should have no knowledge of the possible supplementary services deployed, their logic nor the data related to the supplementary services. This puts requirements upon the SSIP to be as generic and flexible as possible. Another important aspect is how to provide enough information through the SSIP so as to uniquely identify the correct service from a ToSS.

By using the POI identifier (which has an inherent 1-1 relationship with an armed supplementary service) and the calling or called user identifier, a supplementary service can be unambiguously identified.

As given in figure 21, a supplementary service will be invoked when an armed POI is encountered. This implies that the MSSA uses the SSIP for:

- Supplementary Service POI attachment check; i.e. to issue an SSIP request intended to check whether a supplementary service is armed for the POI
- Invocation and Execution of Supplementary Service; i.e. to establish an SSIP dialogue for the supplementary service session.

Given the POI identifier, the context for the supplementary service session is set; that is, the POI (which is a well-defined point in the CC) will determine the set of allowed operations that could be sent over the SSIP for this dialogue. Figure 24 illustrates two established dialogues for two supplementary services.

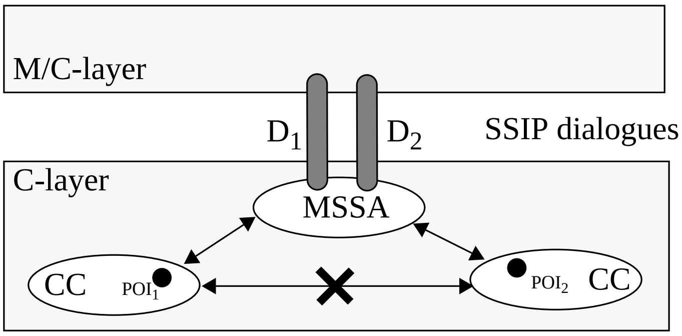


Figure 24 : *SSIP dialog*

One dialogue (D_1) originates from an encountered and armed POI (POI_1) in the CC of the IS, while the other dialogue (D_2) originates from an encountered and armed POI (POI_2) in the CC of the OS. Note also that the message flow between the two CCs is temporarily intercepted.

7.4 M/C-layer

In figure 25 the internals of the M/C-layer are shown in an informal manner. The Model part of the M/C-layer is shaded, while the Control part is the SSAP.

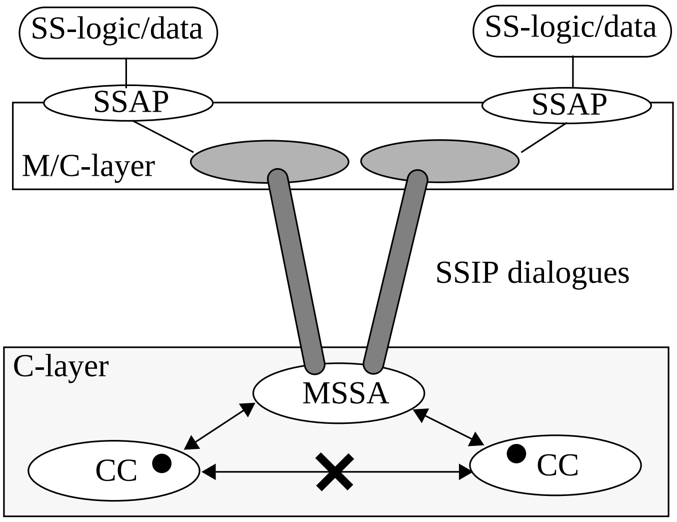


Figure 25 : *Informal view of the M/C-layer*

7.5 Service Logic and Data

Generally speaking, many supplementary services will rely upon information and data about each individual subscriber. Such data is employed during the execution of supplementary services, in order that their logical behavior correspond to that desired by the user. Data of this kind is kept within a persistent store, and accessed by the subscriber when he chooses to (re-)configure the service. Chapter 6 above describes a user-interface for configuration of IAS service data.

7.6 Open Issues

The articulation work upon the architecture for an MMTS Supplementary Service Execution Environment is far from complete; in fact, no standardization work has been done for such a service architecture. A sample of some of the issues requiring investigation are included below:

- design of a service-independent interface for exchange of service data to and from the client
- the GK is not transparent; H.245 signaling must be routed through the GK
- integration and interoperation with MCU functionality
- terminal-to-M/C-layer functionality
- dynamic extension the Call Control EFSM
- resolution and management of interactions amongst supplementary services.

Chapter 8

QoS Support and Adaptation

The purpose of this chapter is to look into the subject of adaptation — from a very general perspective — and to present its relationship to QoS support. The aim in doing so is to create a broad view upon the different issues involved, and to help provide insight into the range and characteristics of the many possible approaches which exist.

8.1 Adaptation: a General Description

Gecsei [25] describes *adaptation* in distributed multimedia systems (**DMS**). There, he emphasizes the goal of achieving user-acceptable performance of applications in the face of unexpected QoS variations:

> *In the context of DMS, adaptation is a complex process involving a number of system components. Despite the multitude of approaches, the overall objective of adaptation is the same: to extend the range of conditions over which a program performs acceptably.*
>
> *(Gecsei [25], p. 59)*

Furthermore, he describes the adaptation process in the following generalized manner:

> *Dynamically, adaptation in a DMS works through a set of mechanisms whose goal is to maintain the operating point within an acceptance region. When the operating point moves outside this region, a controller initiates some corrective action called adaptation, which brings the operating point back within the acceptance region. The controller does this by monitoring the value(s) of the observed parameters and by executing an adaptation algorithm.*
>
> *(Gecsei [25], p. 60)*

To simplify a complex constellation of issues and problems regarding QoS support and adaptation, Gecsei presents the DMS as stratified into three components: *user*, *application* and *system*. In the latter component are found databases, the network / communication infrastructure and the end-system's operating environment. Further, he identifies six important elements of adaptation — elements whose nature and placement in the system help clarify the different types of adaptivity:

> *1 Placement of the controller... This may reside with the user,.. the application,.. or in different parts of the system.*
>
> *2 The DMS component that adapts... Adaptation may occur in the system (for example, when switching to a different protocol...), in the application (switching from color to black-and-white display), or by the user (accepting telephone quality audio).*
>
> *3 Observed parameters... [which] may come from any DMS component. We should also consider the observation method, such as polling, monitoring, exception notification, on demand, or user intervention.*

> 4 *Adaptation algorithm. These rules describe conditions under which adaptation is triggered and the result... This may be embedded in the QoS negotiation procedures... The algorithm must have knowledge of the desired system behavior (the acceptance region...).*
>
> 5 *Time frame. Adaptation can be activated statically... or dynamically. User-centered adaptations tend to be more static than those reacting to changing network conditions.*
>
> 6 *The user's role. The user may be unaware of the system's efforts to maintain QoS, the application may notify him that adaptation is about to take place, or the user may directly control the timing and kind of action to take.*
>
> *(Gecsei [25], p. 60-61)*

8.2 Assessing Approaches to Adaptation

In addressing the problem of making users' working environments adaptive to their needs, as well as the constraints of their local equipment and communication infrastructure, the six elements listed in section 8.1 above give an indication as to the size and dimensions of the solution space. It is useful to assess some of the extremes of this space, and to consider some of the advantages and disadvantages associated with them.

8.2.1 Stringent resource reservation

At one extreme of the spectrum, QoS architectures[1] which achieve apriori resource reservation with hard guarantees can be devised. In this kind of approach, an admission control[2] mechanism can be considered to be initialized with a user-based QoS specification[3]. Thereafter, the system can determine whether or not the resources required to satisfy the specification are available. If so, the required resources can be reserved and pre-allocated, and the relevant end-to-end association(s) can be established. If not, the system returns a negative admission reply to the application and/or user, perhaps with some indications as to what could be achieved instead.

The primary advantage to a stringent resource reservation approach is that the user can obtain strong (and in some cases, absolute) guarantees with respect to the performance and behavior of the applications he employs.

Perhaps the greatest disadvantage with this kind of approach demands a certain amount of homogeneity in the system infrastructure. For instance, if network resources are being reserved through the use of RSVP and IPv6, then there must exist at least one path through the network which includes RSVP/IPv6 implementations on each router along the path.

In addition, it must be ensured that the application-level QoS control protocols[4] employ a common semantics. The same also applies to all middleware-level QoS control protocols

[1] QoS architectures are defined and discussed more thoroughly in chapter 9.

[2] Admission control concerns the decision as to whether a new request can be accommodated in the system. The decision depends upon the resource requirements arising from the requested QoS and the resources available in the system. For further clarification, see section 9.2.3.1.

[3] A QoS specification captures application-level QoS requirements and management policies. For further clarification, see section 9.2.2.

at each end of the association, whenever QoS control signalling is used end-to-end below the application layer. In other words, end-to-end QoS control is only possible when the application(s) and middleware used on each end is *compliant*. Here, ‘compliant’ does not mean that the QoS architectures used at each end of an association must be identical; instead, it means that the semantics of end-to-end QoS control signalling must be preserved.

8.2.2 User-controlled adaptation

Another one of the extremes in the solution space for adaptation is that which is primarily user-controlled. The idea here is that in order to satisfy the user’s QoS preferences, neither the system nor application perform any action upon their own initiative; it is the user alone who acts so as to inform the application (and less directly, the system) as to what steps to take in order to improve and/or degrade QoS[5].

The advantage to this approach is that the user is in full control of the application and system mechanisms used to modify the QoS experienced. The user is also free to adapt by accepting degraded QoS, see points 2 and 6 in section 8.1.

This “advantage” can also be strongly argued to be the greatest disadvantage, as it violates the principle of transparency mentioned later in section 9.2.1. In the face of an extremely heavy network load, this kind of solution would likely yield extremely long response times, and the only means for improving the situation would be through explicit user intervention. In such cases, the need for user intervention could possibly be distracting to the user’s task at hand.

8.2.3 Strict application-control

Yet another extreme point in the solution space for adaptation is one wherein adaptation is accomplished solely within the application. One extreme example could be an application which hides its efforts at maintaining QoS until some threshold is reached. At that time, the application could automatically modify its user interface and functionality, without allowing for modification-intervention by the user. The application might even inform the user of the upcoming interface modification, prior to actually performing that modification.

One advantage to this approach is that the user never needs to bother with helping the application decide what to do when trying to maintain QoS, nor help it decide what to do when the preferred QoS cannot be maintained. In certain work-situations, this could be exactly what is best for the user.

Other users might find this “pre-programmed” behavior to be a disadvantage, since they might feel that they don’t have any control over the way the application behaves and performs: what they want the system to do is simply not what is does !

[4] That is, the QoS control protocols between the client and the server (e.g., between a video player and a video server) *and/or* the control protocols between peer applications (e.g., between two teleconferencing applications).

[5] Here, there is an implicit assumption that the application and/or system can factually influence the QoS being experienced by the user.

With regard to the creation of “pre-programmed” QoS behavior, it is found that Aurrecoechea, et. al. [23], Campbell, et. al. [24] and Gecsei [25] explicitly address this issue. The former identify it as the “QoS management policy” (see section 9.2.2), which is one part of the overall QoS specification supplied by the user. The latter author touches upon it as part of the “adaptation algorithm”, see point 4 in section 8.1.

Concerning strict application-control in order to achieve adaptation, perhaps the most important factor is the degree the user can influence / tailor the application’s pre-programmed QoS behavior, both prior to as well as within a session.

8.3 QoS Support through Hybrid Forms of Adaptation

As illustrated in sections 8.2.1 to 8.2.3 above, a wide range of possibilities exist by which to design adaptation into (distributed) multimedia systems. Gecsei [25] observes that “...the emerging answer seems to be that resource reservation and [application] adaptivity *both* offer valid and complementary methods of DMS design.”

It is logical to conclude that the design solutions for QoS support which appear will be hybrid in nature, exploiting the best characteristics from amongst the different forms of adaptation which can be devised.

Chapter 9

QoS Architectures and a Generalized QoS Framework

The primary purpose of this chapter is to present the motivation for QoS architectures, as well as a Generalized QoS Framework. The aim in doing so is to provide more insight into system-related aspects and issues for supporting QoS via adaptation. This chapter also briefly contrasts resource-oriented and functionally-oriented approaches to QoS support.

The text included in sections 9.1 and 9.2 below (i.e., *"Motivation for QoS Architectures"*, and *"Elements of a Generalized QoS Framework"*, respectively) is copied directly from original work by Aurrecoechea, Campbell and Hauw [23] [24], with the expressed written consent of those authors. The purpose in including this material in its original form is three-fold:

- it offers a well-developed conceptual framework and technical vocabulary,
- from a systems perspective, it more specifically articulates the general principles and elements offered by Gecsei [25] (see section 8.1), and
- it allows the readers immediate access to the original material.

9.1 Motivation for QoS Architectures

Meeting quality of service (QoS) guarantees in distributed multimedia systems is fundamentally an end-to-end issue, that is, from application-to-application. For example, consider the remote playout of a sequence of audio and video: in the distributed system platform, quality of service assurances should apply to the complete flow of media; from the remote server, across the network to the points of delivery. This generally requires end-to-end admission testing and resource reservation in the first instance, followed by careful coordination of disk and thread scheduling in the end-system, packet/cell scheduling and flow control in the network, and finally active monitoring and maintenance of the delivered quality of service. Furthermore, it is also essential that all end-to-end elements of distributed systems architecture work together in unison to achieve the desired application level behaviour.

To date, most of the developments in the provision of quality of service support have occurred in the context of individual architectural layers. Much less progress has been made in addressing the issue of an overall QoS architecture for multimedia communications. There has been, however, considerable progress in the separate areas of Open Distributed Processing (ODP), end system and network support for quality of service. In end-systems, most of the progress has been made in the specific areas of scheduling, flow synchronisation and transport support. In networks, research has focused on providing suitable traffic models and service disciplines, as well as appropriate admission control and resource reservation protocols. Many current network architectures, however, address quality of service from a providers point of view and analyse network performance, failing to comprehensively address the quality needs of applications. Until recently there has been little work on quality of service support in distributed systems platforms. What work there is has been mainly been carried out in the context of the Open Distributed Processing.

The current state of QoS provision in architectural frameworks can be summarized as follows [28]:

i) incompleteness: current interfaces (e.g., application programming interfaces such as Berkeley Sockets) are generally not QoS configurable and provide only a small subset of the facilities needed for control and management of multimedia flows;

ii) lack of mechanisms to support QoS guarantees: research is needed in distributed control, monitoring and maintenance QoS mechanisms so that contracted levels of service can be predictable and assured; and

iii) lack of overall framework: it is necessary to develop an overall architectural framework to build on and reconcile the existing notion of quality of service at different systems levels and among different network architectures.

In recognition of the above limitations, a number of research teams have proposed a systems architectural approach to QoS provision; we refer to these models as QoS architectures in this paper. The intention of QoS architecture research is to define a set of quality of service configurable interfaces that formalize quality of service in the end-system and network, providing a framework for the integration of quality of service control and management mechanisms.

The following section presents a generalized QoS framework and terminology[1] for distributed multimedia applications operating over multimedia networks with quality of service guarantees. The QoS framework is based on a set of principles that govern the behavior of QoS architectures.

9.2 Elements of a Generalized QoS Framework

In what follows, we describe a set of elements used in building quality of service into distributed multimedia systems.These include principles which govern the construction of a generalised QoS framework, QoS specification which captures application level quality of service requirements, and QoS mechanisms which realise desired end-to-end QoS behaviour.

9.2.1 QoS Principles

Five principles motivate the design of a generalised QoS framework:

i) *the integration principle* states that quality of service must be configurable, predictable and maintainable over all architectural layers to meet end-to-end quality of service [29]. Flows[2] traverse resource modules (e.g.,CPU, memory, devices, network, etc.) at each layer from source media devices, down through the source pro-

[1] Where appropriate we have adopted the standard terminology of the ISO QoS Working Group [56]

tocol stack, across the network, up through the receiver protocol stack to the playout devices. Each resource module traversed must provide QoS configurability (based on a QoS specification), resource guarantees (provided by QoS control mechanisms) and maintenance of ongoing flows;

ii) *the separation principle* states that media transfer, control and management are functionally distinct architectural activities [30]. The principle states that these tasks should be separated in architectural frameworks; one aspect of separation is the distinction between signalling and media-transfer; flows (which are isochronous in nature) generally require a wide variety of high bandwidth, low latency, non-assured services with some form of jitter correction; on the other hand, signalling (which is full duplex and asynchronous in nature) generally requires low bandwidth, assured-type services with no jitter constraint;

iii) *the transparency principle* states that applications should be shielded from the complexity of underlying QoS specification and QoS management such as QoS monitoring and maintenance. An important aspect of transparency is the QoS-based API [31] at which desired quality of service levels are stated (see QoS management policy in section 9.2.2). The benefits of transparency are three-fold: it reduces the need to embed quality of service functionality in applications; it hides the detail of underlying service specification from the application; and it delegates the complexity of handling QoS management activities to the underlying framework;

iv) *the asynchronous resource management principle* [30] guides the division of functionality between architectural modules and pertains to the modeling of control and management mechanisms; it is necessitated by, and is a direct reflection of fundamental time constraints that operate in parallel between activities (e.g.,scheduling, flow control, routing, QoS management, etc.) in distributed communications environments; the “state” of the distributed communication system is structured according to these different time scales. The communication system ‘operating point’ is arrived at via asynchronous algorithms that operate and exchange control data periodically among each other; and

v) *the performance principle* subsumes a number of widely agreed rules for QoS-driven communications implementation that guide the division of functionality in structuring communication protocols for high performance in accordance with Saltzer’s systems design principles [32], avoidance of multiplexing [33], recommendations for structuring communications protocols such as application layer framing and integrated layer processing [34], and the use of hardware assists for protocol processing [35] [36].

9.2.2 QoS Specification

QoS specification is concerned with capturing application level quality of service requirements and management policies. QoS specification is generally different at each system

[2] The notion of a flow is an important abstraction which underpins the development of QoS frameworks. Flows characterize the production, transmission and eventual consumption of a single media source (viz. audio, video, data) as integrated activities governed by single statements of end-to-end QoS. Flows are simplex in nature and can be either unicast or multicast. Flows generally require end-to-end admission control and resource reservation, and support heterogeneous QoS demands

layer and is used to configure and maintain QoS mechanisms resident at each layer. For example, at the distributed system platform level QoS specification is primarily user-oriented rather than system-oriented. Lower-level considerations such as tightness of synchronisation of multiple related flows, or the rate and burst size of flows, or the details of thread scheduling should all be hidden at this level. QoS specification is therefore declarative in nature: users specify what is required rather than how this is to be achieved by underlying QoS mechanisms. Quality of service specification encompasses the following:

- *flow synchronisation specification*, which characterises the degree (i.e., tightness) of synchronisation between multiple related flows [37]. For example, simultaneously recorded video perspectives must be played in precise frame by frame synchrony so that relevant features may be simultaneously observed. On the other hand, lip synchronisation in multimedia flows does not need to be absolutely precise when the main information channel is auditory and video is only used to enhance the sense of presence;
- *flow performance specification*, which characterises the user's flow performance requirements [38]; the ability to guarantee traffic throughput rates, delay, jitter and loss rates, is particularly important for multimedia communications. These performance-based metrics are likely to vary from one application to another; to be able to commit necessary end-system and network resources a QoS framework must have prior knowledge of the expected traffic characteristics associated with each flow before resource guarantees can be met;
- *level of service*, which specifies the degree of end-to-end resource commitment required (e.g, deterministic [39], predictive [40] and best effort). While the flow performance specification permits the user to express the required performance metrics in a quantitative manner, level of service allows these requirements to be refined in a qualitative way as to allow a distinction to be made between hard, firm and soft performance guarantees. Level of service expresses a degree of certainty that the QoS levels requested at flow establishment or re-negotiation will actually be honored;
- *QoS management policy*, which captures the degree of QoS adaptation (continuous or discrete) that the flow can tolerate and scaling actions to be taken in the event of violations in the contracted QoS [41]. By trading off temporal and spatial quality to available bandwidth, or manipulating the playout time of continuous media in response to variation in delay, audio and video flows can be presented at the playout device with minimal perceptual distortion. The QoS management policy also includes user-level selection of QoS indications in the case of violations in the requested quality of service, and periodic bandwidth, delay, jitter and loss notification (i.e., QoS signals) which are suitable for adaptive applications [51]; and
- *Cost of Service*, which specifies the price the user is willing to incur for the level of service; cost of service is a very important factor when considering QoS specification. If there is no notion of cost of service involved in QoS specification, there is no reason for the user to select anything other than maximum level of service [42].

9.2.3 QoS Mechanisms

Quality of service mechanisms are selected according to user supplied QoS specification, resource availability and resource management policy. In resource management, QoS mechanisms are categorized as either static or dynamic in nature: static resource management deals with flow establishment and end-to-end QoS re-negotiation phases (which we describe as QoS provision), and dynamic resource management deals with the media-transfer phase (which we describe as QoS control and management). The distinction between the former and latter is due to the different time scales on which they operate and is a direct consequence of the asynchronous resource management principle; control distinguishes itself from management in that it operates on a faster timescale.

9.2.3.1 QoS Provision Mechanisms

QoS provision is comprised of three components:

> i) *QoS mapping* performs the function of automatic translation between representations of QoS at different system levels (i.e., operating system, transport layer, network, etc.) and thus relieves the user of the necessity of thinking in terms of lower level specification. For example, the transport level QoS specification may express flow requirements in terms of level of service, average and peak bandwidth, jitter, loss and delay constraints. For admission testing and resource allocation purposes this representation must be translated to something more meaningful to the end-system scheduler. For example, one function of QoS mapping is to derives the period, quantum (i.e., unit of work), and deadline times of the threads associated the from transport level flows [43].
>
> ii) *admission testing* is responsible for comparing the resource requirement arising from the requested QoS against the available resources in the system. The decision as to whether a new request can be accommodated generally depend on system-wide resource management policies and simple resource availability. Once admission testing has been successfully completed on a particular resource module, local resources are reserved immediately and then committed later if the end-to-end admission control test (i.e., accumulation of hop by hop tests) is successful.
>
> iii) *resource reservation* protocols arrange for the allocation of suitable end-system and network resources according to the user QoS specification. In doing so, the resource reservation protocol interacts with QoS-based routing to establish a path through the network in the first instance; then, based on QoS mapping and admission control at each local resource module traversed (e.g. CPU, memory, I/O devices, switches, routers, etc.)end-to-end resources are allocated. The end result is that QoS control and management mechanisms such as network-level cell scheduler and transport-level flow monitors are configured appropriately;

9.2.3.2 QoS Control Mechanisms

QoS control mechanisms operate on timescales close to media transfer speeds. They provide real-time traffic control of flows based on requested levels of QoS established during the QoS provision phase. This is achieved by providing suitable traffic control mechanisms and arranging for time-constrained buffer management and communication protocol operation. The fundamental traffic control building blocks include the following:

- *flow shaping* regulates flows based on user supplied flow performance specifications. Flow shaping can be based on a simple fixed rate throughput (i.e., peak rate) or some form of statistical representation (i.e., sustainable rate and burstiness) the required bandwidth. The benefit of shaping traffic is that it allows the QoS framework to commit sufficient end-to-end resources and to configure flow schedulers to regulate traffic through the end-systems and network. It has been mathematically proven that the combination of traffic shaping at the edge of the network and scheduling in the network can provide hard performance guarantees. Parekh [44] has shown that if a source is shaped by a token bucket with leaky bucket rate control and scheduled by the weighted fair queueing service discipline [45], it is possible to achieve strong guarantees on delay;
- *flow scheduling* manages the forwarding of flows (chunks of media based on application layer framing) in the end-system [46][47][48] and network (packets and/or cells) in an integrated manner [49]. Flows are generally scheduled independently in the end-systems but may be aggregated and scheduled in unison in the network.This is dependent of the level of service and the scheduling scheme adopted;
- *flow policing* can be viewed as the dual of monitoring: the latter — usually associated with QoS management — observes whether QoS contracted by a provider is being maintained whereas the former observes whether the QoS contracted by a user is being adhered to. Policing is often only appropriate where administrative and charging boundaries are being crossed, for example, at a user-to-network interface [50]. A good flow shaping scheme at the source allows the policing mechanism to easily detect misbehaving flows. The action taken by the policing function can range from accepting violations and merely notifying the user, through to shaping the incoming traffic to an acceptable QoS level. We consider that policing flows in the end-system or network should be a function of the end-system or network level scheduling QoS mechanism;
- *flow control* includes both open-loop and closed loop schemes: open loop flow control is used widely in telephony and allows the sender to inject data into the network at the agreed levels given that resources have been allocated in advance; closed loop flow control requires the sender to adjust its rate based on feedback from the receiver [51] or network [57]. Applications using closed loop flow control based protocols must be able to adapt to fluctuations in the available resources. Fortunately, many multimedia applications are adaptive [52][53] and can operate in such environments. Alternatively, multimedia applications which cannot adjust to changes in the delivered QoS are more suited to open loop schemes where bandwidth, delay and loss can be deterministically guaranteed for the duration of the session; and
- *flow synchronisation* is required to control the event ordering and precise timings of multimedia interactions.Lip-sync is the most commonly cited form of multimedia synchronisation (synchronisation of video and audio flows at a playout device); other synchronisation scenarios reported include: event synchronisation with and without user interaction, continuous synchronisation other than lip-sync, continuous synchronisation for disparate sources and sinks. All place fundamental QoS requirements on flow synchronisation protocols [54]. Dynamic QoS management associ-

ated with flow synchronisation is mainly concerned with the 'tightness' of synchronisation between flows.

9.2.3.3 QoS Management Mechanisms

To maintain agreed levels of QoS it is often not sufficient to just commit resources; in addition, QoS management is frequently required to ensure that the contracted QoS is sustained. QoS management of flows is functionally similar to QoS control. However, it operates on a slower time scale; that is, over longer monitoring and control intervals [55]. QoS management mechanisms include the following:

- *QoS monitoring* allows each level of the system to track the ongoing QoS levels achieved by the lower layer.It often plays an integral part in a QoS maintenance feedback loop which maintains the quality of service being achieved by the monitored resource modules. Monitoring algorithms operate over different timescales.For example, they can run as part of the scheduler (as a QoS control mechanism) to measure individual performance of ongoing flows. In this case measured statistics can be used to control packet scheduling and for admission control. Alternatively they can operate as part of a transport level feedback mechanism [58] [59];
- *QoS maintenance* compares the monitored quality of service against the expected performance and then exerts tuning operation (i.e., fine or coarse grain resource adjustments) on resource modules to sustain the delivered QoS. Fine grain resource adjustment counters QoS degradation by adjusting local resource modules(e.g., loss via the buffer manager, queueing delays via the flow scheduler and throughput via the flow regulator [29]);
- *QoS degradation* issues a QoS indication to the user when it determines that the lower layers have failed to maintain the QoS of the flow and nothing further can be done by the QoS maintenance mechanism. In response to such an indication the user can choose either to adapt to the available level of QoS or scale to a reduced level of service (i.e., end-to-end renegotiation);
- *QoS signalling* allows the user to specify the interval over which one or more QoS parameters (e.g., delay, jitter, bandwidth, loss, synchronisation) can be monitored and the user informed of the delivered performance via a QoS signal. Both single and multiple QoS signals can be selected depending the user requested QoS management policy; and
- *QoS scalability* comprises *QoS filtering* (which manipulates flows as they progress through the communications system) and *QoS adaptation* (which scales flows at the end-systems only) mechanisms. Many continuous media applications exhibit robustness in adapting to fluctuations in end-to-end quality of service. Based on the user supplied QoS management policy, QoS adaptation in the end-systems can take remedial actions to scale flows appropriately. Resolving heterogeneous quality of service issues is a particularly acute problem in the case of multicast flows. Here individual receivers may have differing capabilities to consume audio-visual flows; QoS filtering helps to bridge this heterogeneity gap while simultaneously meeting individual receivers' quality of service requirements.

9.3 Focus of the Generalized QoS Framework

In the introduction to the Generalized QoS Framework presented above, Aurrecoechea, et. al. write [23][3]:

> *...a number of research teams have proposed a systems architectural approach to QoS provision; we refer to these models as QoS architectures... The intention of QoS architecture research is to define a set of quality of service configurable interfaces that formalize quality of service in the end-system and network, providing a framework for the integration of quality of service control and management mechanisms... The QoS framework is based on a set of principles that govern the behavior of QoS architectures.*

This statement clearly indicates that the Generalized QoS Framework presented above reflects a systems architectural perspective, with focus upon end-systems and networks. What is less well-addressed and articulated in the framework are the aspects of QoS which involve the user, such as the user's role and interaction of the user with the application.

Before closing this chapter, an alternative to resource-oriented QoS support is presented.

9.4 Resource- vs. Function-Oriented QoS Support

QoS architectures often focus upon *resources*. That is, many QoS architectures focus upon static and dynamic resource reservation and allocation — within both end-systems and network — based upon a QoS specification originating from the user. Among other things, these resources include buffers, bandwidth, devices, memory, CPU, etc.

In contrast, another approach toward addressing QoS requirements is based upon real-time protocol configuration. Here, the focus is upon *functionality*, rather than resources. The philosophy behind this kind of approach is to employ the most streamlined protocol configuration possible, while still satisfying the functional requirements inherent in the user's QoS specification. In terms of protocol functionality, a streamlined protocol configuration carries little "extra baggage", thereby improving end-to-end performance.

The Da Capo system [26], [27] uses dynamic protocol configuration for QoS support, It includes a mapping from protocol functionality down to protocol mechanisms, and from these mechanisms down to [protocol] modules. Given a QoS specification as input, Da Capo analyzes the specification's functional requirements, then employs weighting functions and heuristic rules in order to rapidly create a (near) optimal configuration of protocol modules which satisfies those requirements. Da Capo performs these actions in real-time, and is therefore capable of performing protocol *re*-configuration, when conditions call for it.

It should be clear that both resource- and function-oriented approaches to QoS support have the same aim: that is, to render users' working environments more flexible, tailorable and robust.

[3] See also section 9.1.

Chapter 10

Focal Areas for Future Work

This chapter first summarizes some overall considerations which should be made when considering further work within an IMiS-Ericsson context. Thereafter, some recommendations are made as to which areas of work should receive attention, and certain project goals, objectives and rationale for such work are suggested.

10.1 Exploiting Possibilities for Synergy

Within this first IMiS-Ericsson project, efforts were made to exploit common areas of ongoing work within the other IMiS projects (i.e., IMiS-Kernel, IMiS-Ericsson II, IMiS-Veritas and ENNCE). The IMiS Forum meetings and IMiS Reference Group meeting helped to provide and reap the effects of synergy amongst these projects. It is strongly recommended that any new IMiS-Ericsson project context (i.e., “IMIS-Ericsson II”) *continue to exploit* the common areas of focus amongst these projects. The primary areas of common work amongst these projects are:

- achieving seamlessness across
  - services
  - network types
  - terminal types
  - work and collaboration contexts: and,
- issues and technology related to QoS.

It is therefore recommended that IMIS-Ericsson II include work related to these areas. It is also recommended that IMIS-Ericsson II exploit the experimental network being developed within the IMiS-Kernel project. Figure 26 provides a very-high-level illustration of the current state of that network.

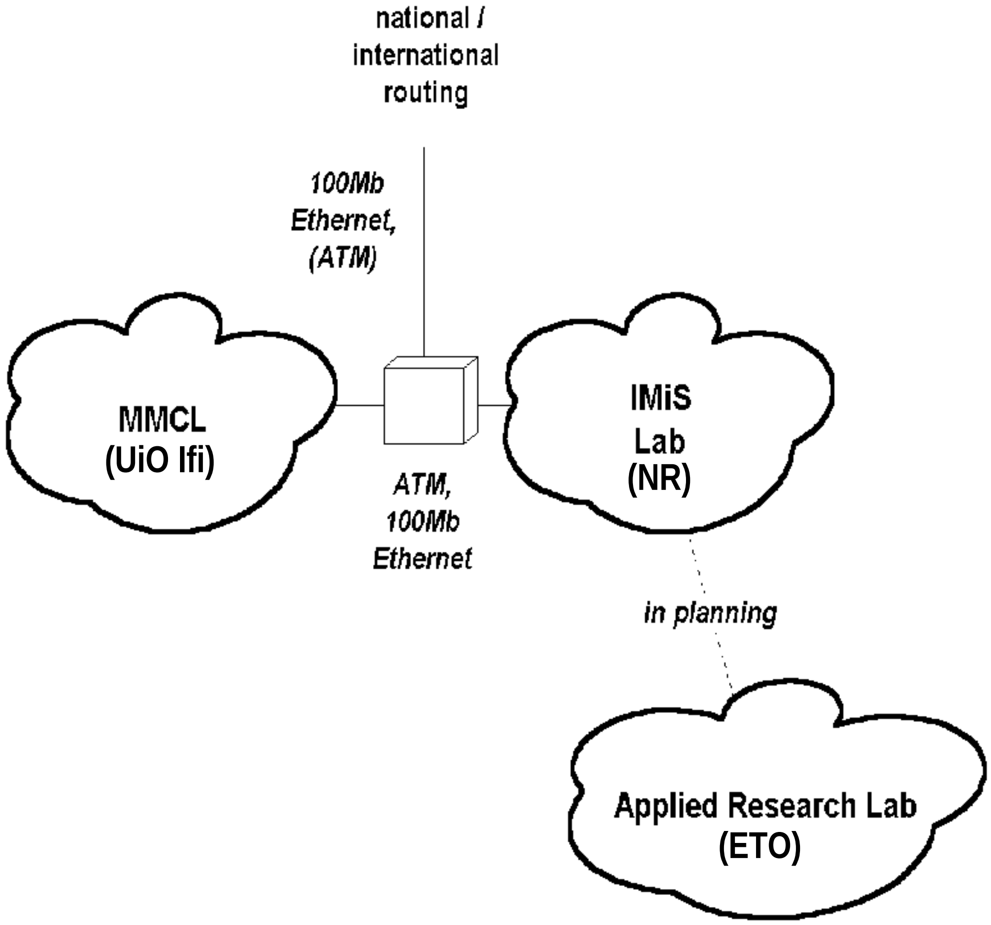


Figure 26 : *IMiS-Kernel experimental network*

IMiS Kernel's infrastructural context consists of the IMiS Lab (at NR) and the Multimedia Communication Lab (MMCL, at the University of Oslo's Institute for informatics (UiO Ifi). The funding for NR's part in this infrastructure has been supplied almost exclusively through the IMiS-Kernel project (with a particularly large investment from UNINETT). The funding for the MMCL at UiO Ifi has been partially supplied through the ENNCE Project. As seen in the figure, a communication connection between NR and ETO's Applied Research Lab is soon to be realized.

Much of the shared switching and routing equipment within this experimental network was purchased as part of a Coordinated infrastructural Development phase within the IMiS-Kernel and ENNCE projects. **On March 23, 1998, this experimental network was officially opened as the core element of "Internet 2 in Norway".**

10.2 Characterization of IMiS-Ericsson II

10.2.1 Suggested goals and objectives

It is suggested that an eventual IMiS-Ericsson II project have the following primary goals:

- The project should contribute to ETO’s efforts in developing a more intelligent network
- The project should produce an (H.323-based) prototype for collaborative multimedia communication, to improve communication
- The project should carry out and disseminate relevant research results, in particular to create an application framework for collaborative multimedia which maximally exploits network intelligence and services.

Some of the long-term research objectives for the project should be:

- The development of an application framework and object model for collaborative multimedia on a packet-switched network
- Employing user-studies in order to identify communication and coordination needs for a selected pilot group within some organization (e.g., ETO)
- Employing user-participatory design and evaluation during the creation of an H.323-based client aimed to support the pilot group
- Designing of a user-interface for the client
- Exploration of QoS issues and resource reservation schemes for H.323-based networks (together with IMiS-Kernel)
- Dissemination of progress and results

Some of the technical objectives for the project should be:

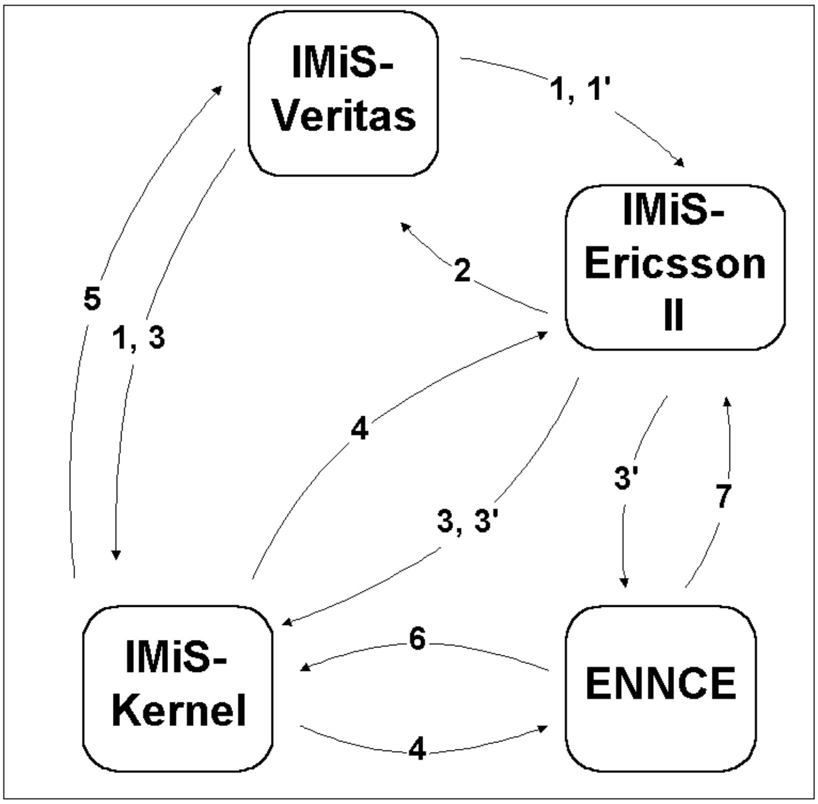
- Implementation of an H.323-based client prototype for supporting real-time electronic personal communication
- Implementation and testing of application-level security features within the prototype
- Establishment of an H.323 zone at NR
- Extension of the H.323-based client prototype with:
    - facilities for supporting asynchronous EPC (e.g., multimedia messaging)
    - enhanced communication and interaction facilities, through the use of specialized application and network services
- To demonstrate an integrated, prototype application which transcends the collaborative seams between time and space

10.2.2 Rationale

A project having the character described above should be specified in close cooperation and coordination with IMiS-Kernel. Given proper planning, it would be possible to develop:

- an IPv6-based H.323 client, for concrete investigation of QoS upon an IPv6 experimental network, and
- an application framework for creation and experimentation with a variety of interoperable IPv4/IPv6/H.323 multimedia applications.

The rationale here is that when these two kinds of developments were “in hand”, it is judged both NR and ETO would find themselves in an excellent position to continue exploration into areas of applied research which lay in the forefront of communication issues and technology. These opportunities can be further understood by studying figure 27, which illustrates the some of the possibilities for synergy amongst the IMiS Project Family.



1 real users; factual work contexts; functional requirements

1’ user participatory design and evaluation.; requirements for ETO’s network services

2 support for collaborative multimedia

3 infrastructural requirements; QoS requirements

3’ H.323-based MEDIATE client

4 IPv4/IPv6 infrastructure; resource reservation testbed; IPv4/IPv6 MEDIATE client

5 support for personal mobility; auto-configuration; service location

6 reference model for addressing QoS requirements

7 UI requirements for enabling users’ specification/observation/management of QoS

Figure 27 : *Possibilities for synergy amongst the IMiS Projects*

References

[1] Alvestrand, H., Børseth, H., Lovett, H., Ølnes, J., Final report for IMiS feasibility project: Potential for a main program with focus on mechanisms for and use of multimedia applications in seamless networks, NR Note IMEDIA/04/97, Oslo, January 1997.

[2] Børseth, H., Holmes, P., Johansen, B., Lindsjørn, Y., Lovett, H., Lous, J., Løbersli, F., Villa, B., IMiS Pilot Project — Final Report: Infrastructure for Multimedia Applications in Seamless Networks, SINTEF Report STF40 F97020, February 1997 (Restricted).

[3] WWW Homepage for IMIS-Veritas:
http://www.nr.no/imis/veritas/

[4] WWW Homepage for IMIS-Kernel:
http://www.nr.no/imis/imis-k/

[5] WWW Homepage for ENNCE:

[6] ITU-T Recommendation F.850 - Principles of Universal Personal Communication.

[7] Gritzman, M., Kluge, A., Lovett, H., “Task Orientation in User Interface Design”, in Nordby, K., Helmersen, P., Gilmore, D.J., Arnesen, S.A. (eds.) *Human Computer Interaction*, Interact ‘95, pp. 97-102. Chapman and Hall, London, Glasgow, 1995.

[8] Kristof, R. & Satran, A., *Interactivity by Design - Creating & Communicating with New Media*, Adobe Press, 1995.

[9] Norman, D. A., “Why Interfaces Don’t Work”, in Brena Laurel (ed.) *The Art of Human-Computer Interface Design*, Addison-Wesley Publishing Company, pp. 209-219, 1990.

[10] Schneiderman, B., *Designing the User Interface*, Addison Wesley, 1998.

[11] ITU-T Recommendation I.210 - Principles of telecommunication services supported by an ISDN and the means to describe them.

[12] Draft ITU-T Recommendation H.323V2 - Packed Based Multimedia Communication System, March 27, 1997.

[13] Draft ITU-T Recommendation H.225.0, Version 2 - Call Signaling Protocols and Media Stream Packetization for Packet Based Multimedia Communication Systems.

[14] Draft ITU-T Recommendation H.450.2 - Call Transfer Supplementary Service for H.323.

[15] *Public IntraNet Service Network*, ETX/TX/BP-97:009, 1997-08-20.

[16] *Security and Encryption for H Series multimedia terminals*, Draft ITU-T Recommendation H.235, March 1997.

[17] *Modifications to H.235 for Firewall Operation and Certificate Usability*, APC-1181, May 30th, 1997.

[18] *H.235: Message Integrity for the H.225.0 RAS channel*, Version 1.1, APC-1176, May 26th, 1997.

[19] "The TLS Protocol Version 1.0", T. Dierks and C. Allen, Internet Draft, May 21, 1997.

[20] "New Directions in Cryptography", W. Diffie and M. Hellman, IEEE Transactions on Information Technology, V. IT-22, n. 6, June 1977, pp. 74-84.

[21] "IP Security Document Roadmap", R. Thayer, N. Doraswamy and R. Glenn, Internet Draft, July 1997.

[22] "Applied Cryptography", Bruce Schneier, Second Edition, Wiley 1996, Chapter 7 and chapter 25.

[23] Aurrecoechea, C., Campbell, A., Hauw, L., "A Survey of Quality of Service Architectures", Technical Report, Lancaster University, ftp://ftp.comp.lancs.ac.uk/mpg/MPG-95-18.ps.Z, 1995.

[24] Campbell, A., Aurrecoechea, C. and L. Hauw, ``A Review of QoS Architectures,'' ACM Multimedia Systems Journal, 1996.

[25] Gecsei, J., "Adaptation in Distributed Multimedia Systems", IEEE Multimedia, April-June 1997, Vol. 4, No. 2 , pp. 58-66.

[26] Plagemann, T.: A Framework for Dynamic Protocol Configuration, Dissertation at Swiss Federal Institute of Technology (Diss. ETH No. 10830), Switzerland, September 1994 (also available at vdf Hochschulverlag AG an der ETH Zurich, Switzerland, 1996, ISBN 3 7281 2334 X).

[27] Plagemann, T., Plattner, B., Vogt, M., Walter, T.: Modules as Building Blocks for Protocol Configuration, Proceedings International Conference on Network Protocols, ICNP'93, San Francisco, USA, October 1993, pp. 106-115.

[28] Hutchison, D., Coulson G., Campbell, A., and G. Blair , "Quality of Service Management in Distributed Systems", M. Slomaned., Network and Distributed Systems Management, Addison Wesley, chapter 11, 1994.

[29] Campbell, A., Coulson, G., García, F., Hutchison, D., and H. Leopold, "Integrated Quality of Service for Multimedia Communications", Proc. IEEE INFOCOM'93, pp. 732-739, San Francisco, USA, April 1993.

[30] Lazar, A.A., "A Real-time Control, Management, and Information Transport Architecture for Broadband Networks", Proc.International Zurich Seminar on Digital Communications, pp. 281-295, 1992.

[31] Bansal, V., Siracusa, R.J, Hearn, J. P., Ramamurthy and D. Raychaudhuri, "Adaptive QoS-based API for Networking", FifthInternational Workshop on Network and Operating System Support for Digital Audio and Video, Durham, New Hampshire,April, 1995.

[32] Saltzer, J., Reed, D., and D. Clark, "End-to-end Arguments in Systems Design", ACM Transactions on Computer Systems,Vol. 2., No. 4., 1984.

[33] Tennenhouse, D.L., “Layered Multiplexing Considered Harmful”, Protocols for High-Speed Networks, Elsevier SciencePublishers (North-Holland), 1990.

[34] Clark, D., and D.L. Tennenhouse, “Architectural Consideration for a New Generation of Protocols”, Proc. ACM SIGCOMM’90, Philadelphia, 1984.

[35] Chesson, G., “XTP / PE Overview”, Proc. 13th Conference on Local Computer Networks, Pladisson Plaza Hotel, Minneapolis,Minnesota, 1988.

[36] Zitterbart, M., Stiller, B., and A Tantawy,”A Model for Flexible High-Performance Communication Subsystems”, IEEEJSAC, May 1992.

[37] Little, T.D.C, and A. Ghafoor, “Synchronisation Properties and Storage Models for Multimedia Objects”, IEEE Journal onSelected Areas on Communications, Vol. 8, No. 3, pp. 229-238, April 1990.

[38] Partridge, C., “A Proposed Flow Specification”, Internet Request for Comments, no. 1363, Network Information Center, SRIInternational, Menlo Park, CA, September 1990.

[39] Ferrari, D. and Verma D. C., “A Scheme for Real-Time Channel Establishment in Wide-Area Networks”, IEEE JSAC, 8(3),368-77, 1990.

[40] Clark, D.D., Shenker S., and L. Zhang, “Supporting Real-Time Applications in an Integrated Services Packet Network:Architecture and Mechanism”, Proc. ACM SIGCOMM’92, pp. 14-26, Baltimore, USA, August, 1992.

[41] Campbell, A., Coulson G. and D. Hutchison, “Supporting Adaptive Flows in a Quality of Service Architecture”, MultimediaSystems Journal, November, 1995.

[42] Kelly, F. P., “On Tariffs, Policing and Admission Control for Multiservice Networks”, Proc. Multiservice Networks ‘93,Cosener’s House, Abingdon, July 1993, and Internal Report, Statistical Laboratory, University of Cambridge, England, 1993.

[43] Coulson, G., Campbell, A and P. Robin, “Design of a QoS Controlled ATM Based Communication System in Chorus”, IEEEJournal of Selected Areas in Communications (JSAC), Special Issue on ATM LANs: Implementation and Experiences withEmerging Technology, May 1995.

[44] Parekh, A. and R. G. Gallager, “A Generalised Processor Sharing Approach to Flow Control in Integrated Service Networks- The Multiple Node Case”, Proc. IEEE INFOCOM’93, pp.521-530, San Francisco, USA, April 1993.

[45] Keshav, S., “On the Efficient Implementation of Fair Queueing”, Internetworking: Research and Experiences, Vol. 2, pp 157-173, 1991.

[46] C. Liu, J. Layland, “Scheduling Algorithms for Multiprogramming in Hard Real Time Environment”, Journal of the ACM,1973.

[47] Stankovic et al., “Implications of Classical Scheduling Results for Real-Time Systems”, IEEE Computer, Special Issue onScheduling and Real-Time Systems, June 1995.

[48] Tokuda H. and T. Kitayama, “Dynamic QOS Control Based on Real-Time Threads”, Proc. Fourth International Workshop onNetwork and Operating System Support for Digital Audio and Video, Lancaster University, Lancaster LA1 4YR, UK, 1993.

[49] H. Zhang, S. Keshav, “Comparison of Rate-Based Service Disciplines”, ACM SIGCOMM, 1991.

[50] ATM Forum, ATM User-Network Interface Specifications, Version 3.0, Prentice-Hall, 1993.

[51] Shenker, S., Clark, D., and L. Zhang, (1993) “A Scheduling Service Model and a Scheduling Architecture for an IntegratedService Packet Network”, Working Draft available via anonymous ftp from parcftp.xerox.com: /transient/service-model.ps.Z.

[52] Jacobson, V., (1994) “VAT: Visual Audio Tool”, vat manual pages, Feb 1993.

[53] Kanakia, H., Mishra, P., and A. Reibman, (1993) “An Adaptive Congestion Control Scheme for Real Time Packet VideoTransport”, Proc. ACM SIGCOMM ‘93, San Francisco, USA, October 1993.

[54] Escobar, J., Deutsch, D. and C. Partridge, “Flow Synchronisation Protoco”, IEEE GLOBECOM’92, Orlando, Fl., December1992.

[55] Pacifici, G., and R. Stadler, “An Architecture for Performance Management of Multimedia Networks”, Proc. IFIP/IEEEInternational Symposium on Integrated Network Management, Santa Barbara, May 1995.

[56] ISO-QoS, “Quality of Service Basic Framework - Qutline”, ISO/IEC JTC1/SC21/WG1 N1145, International StandardsOrganisation, UK, 1994.

[57] Jain, R., “Congestion Control and Traffic Management in ATM Networks: Recebt Advances and a Survey”, ComputerNetworks and ISDN Systems (to appear).

[58] Campbell A., Coulson G., Garcia F. and Hutchison D., “A Continuous Media Transport and Orchestration Service”, Proc.ACM SIGCOMM ‘92, Baltimore, Maryland, USA, 99-110, 1992.

[59] Schmit, C., “QoS-Monitoring - a Generic Approach”, Technical Report, University of Karlsruhe, Institue of Telematics,Germany, 1995.

Appendix A

List of Seminars and Talks

A list of the seminars and talks either initiated, given and/or attended by members of the IMiS-Ericsson project follows below.

Title: **“H.323, MMTS and Ericsson’s IP-based Service Platform”**
Given by: Werner Erikssen (ETO)
Date: 12 September 97
Meeting type: Closed
Project(s) represented: Imis-E, Imis-K

Title: **“Applying Intelligent Network architecture concepts to an H.323-based architecture for supplementary services”**
Given by: Anders Frøyhaug (ETO)
Date: 12 September 97
Meeting type: Closed
Project(s) represented: Imis-E, Imis-K

Title: **“Ericsson’s H.323 call model and its relation to the Gatekeeper”**
Given by: Espen Skjæran (ETO)
Date: 18 September 97
Meeting type: Closed
Project(s) represented: Imis-E, Imis-K

Title: **“Quality of Service and Dynamic Protocol Configuration within Da CaPo”**
Given by: Thomas Plagemann (UniK, Kjeller)
Date: 2 October 97
Meeting type: Open
Project(s) represented: Imis-E, Imis-K and ENNCE

Title: **“Personal mobility, UPT and Timetable Service”**
Given by: Paul Fjuk (ETO)
Date: 3 October 97
Meeting type: Open to IMiS Projects
Project(s) represented: Imis-E, Imis-K, Imis-V

Title: **“SSL: Secure Sockets Layer”**
Given by: Jan-Roger Sandbakken (NR)
Date: 6 October 97
Meeting type: Open
Project(s) represented: Imis-E, Imis-K, Imis-V, ENNCE

Title: **“The Imis project family: Presentation and IdeaBox” (First Meeting of the Imis Forum)**
Given by: NR, ETO, DnV
Date: 16 October 97
Meeting type: Open to IMiS Projects
Project(s) represented: Imis-E, Imis-K, Imis-V, ENNCE

Title: **“H.323, MMTS and the Gatekeeper”**
Given by: Werner Erikssen (ETO)
Date: 23 October 97
Meeting type: Open to IMiS Projects
Project(s) represented: Imis-E, Imis-K, Imis-V

Title: **“How to enable ubiquitous accessibility without unfortunate sessions”**
Given by: Fredrik Ljungberg (University of Gothenburg)
Date: 30 October 97
Meeting type: Open
Project(s) represented: Imis-E, Imis-K, Imis-V

Title: **“IPv6, RSVP and ATM”**
Given by: Lars Aarhus, Jannicke Riisnæs (NR)
Date: 5 December 97
Meeting type: Open
Project(s) represented: Imis-E, Imis-K, Imis-V, ENNCE

Title: **“Java Media Framework**
Given by: Bent Johansen (NR)
Date: 12 January 98
Meeting type: Open
Project(s) represented: Imis-E, Imis-K, Imis-V, ENNCE

Title: **“Second Meeting of the Imis Forum”**
Given by: NR
Date: 12 February 98
Meeting type: Open to IMiS Projects and ENNCE
Project(s) represented: Imis-E, Imis-K, Imis-V, ENNCE

Title: **“First Meeting of the IMIS Reference Group”**
Hosted by: NR, ETO, DnV
Date: 12 February 98
Meeting type: Open to invited members of IMiS Projects and ENNCE
Project(s) represented: Imis-E, Imis-K, Imis-V, ENNCE
Organizations represented: NR, UiO IfI, UiT, UniK, UNINETT, ETO, DnV

Title: **“The IMiS Ericsson Project”**
Given by: Peter Holmes (NR)
Date: 23 February 98
Meeting type: Open
Project(s) represented: Imis-E, Imis-K, Imis-V, ENNCE

Title: **“Type Checking and Binding of Stream Interfaces”**
Given by: Frank Eliasson (University of Tromsø)
Date: 18 March 98
Meeting type: Open
Project(s) represented: Imis-E, Imis-K, Imis-V, ENNCE

Appendix B

Input drafted for the H.235 standardization effort

The material in this Appendix was drafted for the January 1998 standardization meeting concerning ITU-T Recommendation H.235. Unfortunately, this draft was not finalized in time, and was therefore not submitted to that meeting. The material is included in this report in order that the H.235 draft input created in this project be documented. As the H.235 standard continues to mature, the opinions and assessments below can be reviewed once again, in order to judge whether the standard has taken these kinds of issues into account.

The underlying trust model is not clearly defined

We would like H.235 to explicitly discuss the trust model(s) assumed in the document. The draft discusses to some extent how H.323 components may be trusted to handle encryption keys, but the draft does *not* address the concept of trust models as such. Which entities in the system should be trusted, and for what purposes. This has important implications on the authentication schemes. To what extent, for instance, may H.323 components such as gateways and gatekeepers (i.e. the network) be trusted to authenticate users on behalf of others. These issues should perhaps be discussed in a separate chapter.

There should be a larger focus on network issues

Certain network issues are neglected in the current draft. It may for instance be necessary to have users authenticate them to the network (i.e. gatekeepers or gateways in H.323 multimedia service network) and not only end to end between users. End to end authentication using certificates may for instance be impractical when their certificate trust hierarchies differ. Cross certification, when several levels are involved, is difficult to manage. The complexity of the client will increase, and the number of certificates that must be exchanged will multiply, which as severe implications on the set-up time.

User-to-network authentication may also be required in order to achieve proper charging and for handling of traffic that cross different administrative domains (e.g. different network operators and service providers).

These authentication aspects must be resolved for phone-to-phone (gateway-to-gateway) communication, phone-to-PC communication (gateway-to-PC), PC-to-PC communication and also for communication between different administrative domains of the multimedia service network.

We do not require ready solutions in H.235 on these matters, but it is very important that the alternatives are not ruled out.

Client Authentication Issues

The draft does not fully discuss client authentication issues. It does not discuss how user identities and terminal identities relate. Authenticity discussions may at several places be interpreted to address both *user authentication* and *end-terminal authentication*, and a separate discussion on this is missing. Nor does the draft talk about how user identities and terminal identities, such as IP addresses, may be matched up.

The client authentication issues need to be clarified in terms of:

- when to authenticate the H.323 client SW (PC-client)
- when to authenticate the end-user
- mechanisms to employ in each case.

Chapter 6.2 refers to “authorization certificates”. It is unclear if this is an unlucky misspelling, or if it is the intention to refer to SDSI/SPKI extensions.

Chapter 6.2.1 discuss digital signatures and authentication. Please clarify that this does not relate to use of signature certificates.

Authentication of clients mediated/ bridged by MCU

It is stated that clients and MCU authenticate each other using certificates and that further client-to-client authentication is based on certificates provided by the MCU.

Problem-1

Unless it can be ensured that all participants share the same single CA for the session, it may be extremely hard for the CA to give a client requesting authentication of a ‘peer’ a certificate it understands (even if we presume that the MCU stores all certificates of all participants, which is expected to be unrealistic in a scaled network).

Problem-2

A solution where the CA shall accept only one CA pr. session may introduce scaling and distribution problems related to:

- finding a CA that all clients in the session may accept when the session is being set up one participant at a time
- on-demand (unscheduled) adding of new legs to the conference where the new participant only is accepted if it knows the conference CA.

Alternative-1

The MCU accepts only one CA (pr. session) which all clients needs to support (alternatively, the MCU is associated with a single CA).

Alternative-2

The MCU accepts any client certificate of which it knows the CA and respond to authentication requests by cross-signing/ building the required certificate chain (policy-wise dubious and bad performance).

Alternative-3

As the MCU is sufficiently trusted to manipulate the media streams, it should inherently be trusted to authenticate the other participants (i.e. it could fake any participant or media streams if it wanted to). It is therefore suggested to add simple H.245 primitives (non X.509 related) for this authentication.

RAS confidentiality issues are not discussed

The document should give a brief discussion about confidentiality requirements/ issues of the RAS channel (cf. traffic analysis: who calls who at what time). In some environments this can prove to be critical (e.g. finance, law,...). The document may also list/ recommend the different candidate solutions for this (e.g. IPsec).

An introduction is missing

The scope of the document is not clearly stated. A figure like the one showing the protocol stack with the scope H.235 indicated, should be included at the beginning of the document. Also it should be discussed why other types of communication than audio and video is excluded.

Several technical decisions are made without discussion

Symmetric, Kerberose-like solutions are currently excluded. Any discussions or arguments on this are missing.

The document is somewhat biased towards Diffie-Hellmann solutions. Please ensure that corresponding RSA schemes are not being excluded as this is what currently is available with smartcard technologies.

Many of the definitions in chapter 4 should be revised