Performance Evaluations of IP Multiparty Multimedia Communications over GEO Satellite Networks

Lei Liang

Submitted for the Degree of Doctor of Philosophy from the University of Surrey

UniS

Centre for Communication Systems Research
School of Electronics and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, UK

November 2005

© Lei Liang 2005
Summary

With the rapid development of the Internet, new theories and technologies are blooming and boosting the associated applications. One group of important applications is the multiparty multimedia communications. Satellites, which have played an important role in telephony communication and TV broadcasting services, could also play an important role to provide multiparty multimedia communications with their global coverage and on-board processing ability over IP networks.

IP multiparty multimedia conferencing is one of these communication applications. With the support of satellites, one can provide an IP conferencing service globally to anywhere, even the place does not have the access to terrestrial networks. This thesis introduces the VoIP technologies that underpin IP conferencing services. It describes protocols, architectures, network entities, and network performance. In addition, this thesis presents how a multicast routing architecture was designed for an IP conferencing system with consideration to the new features introduced by an integrated GEO satellite network. An associated conferencing model will be presented as well to accompany this routing architecture. All of the technologies used in the design were implemented in a demonstrator. To test and evaluate the system, efforts have been put into the IP traffic measurement technologies and a measurement regime was developed to evaluate the system with consideration to multicast routing and the system architecture.

New relative QoS requirements of multiparty communications are identified in this thesis. A set of parameters to present these new requirements are proposed as derivations of the IPPM (IP Performance Metrics) end-to-end parameters. A new adaptive QoS optimisation algorithm is proposed that is based on the measurement of these new parameters to satisfy the relative QoS requirements of the multiparty multimedia communications. Simulations were carried out to verify this algorithm and the results prove that it can optimize the relative QoS for multiparty multimedia communications.
Acknowledgments

I would like to thank my parents for all their support on my study in Surrey.
I would like to thank my supervisor Dr. Zhili Sun and Dr. Haitham Cruickshank.
I would also like to thank the rest of members in the network group for their help.
I would specially acknowledge the sponsor from the EU IST project IP ConferEncing with Broadband multimedia ovER Geostationary Satellites (ICEBERGS).
# Contents

Summary................................................................................................................. ii

Contents...................................................................................................................... iv

List of Figures ............................................................................................................. ix

List of Tables ............................................................................................................... xi

Glossary of Terms ....................................................................................................... xii

Publication List .......................................................................................................... xvi

1 Introduction .............................................................................................................. 1

   1.1 Motivations........................................................................................................... 2

   1.2 Objectives ............................................................................................................ 3

   1.3 Main Contributions ............................................................................................. 4

   1.4 Thesis Outline ..................................................................................................... 5

2 Review of VoIP Technologies .................................................................................. 7

   2.1 Teleconferencing Protocol Suite - H.323 ............................................................ 8

       2.1.1 H.323 Entities ............................................................................................... 8

       2.1.2 H.323 Protocol Set ....................................................................................... 8

       2.1.3 H.323 Benefits ............................................................................................. 8

   2.2 Session Initiation Protocol (SIP) ......................................................................... 10

       2.2.1 SIP Entities .................................................................................................... 10

       2.2.2 SIP Messaging Syntax ............................................................................... 11

       2.2.3 SIP Message Headers .................................................................................. 12

       2.2.4 Redirect and Proxy Servers ........................................................................ 12

       2.2.5 SIP Procedure ............................................................................................ 13

       2.2.6 Integration with Existing Protocols ............................................................. 14

       2.2.7 SIP Benefits ................................................................................................ 14

       2.2.8 SIP vs. H.323 ............................................................................................. 15

   2.3 Real-time Transport Protocol (RTP) .................................................................. 16

   2.4 Real Time Control Protocol (RTCP) ................................................................. 19

   2.5 Relevance to Satellite Networks ......................................................................... 20

   2.6 Summary .............................................................................................................. 21
3 Multicast Routing Protocols ................................................................................................22
  3.1 Introduction to Multicast ...............................................................................................22
    3.1.1 Multicast Addressing ............................................................................................23
    3.1.2 Multicast Scope ....................................................................................................23
    3.1.3 Internet Group Management Protocol (IGMP) .....................................................24
  3.2 Multicast Routing Protocols ..........................................................................................25
    3.2.1 Intra-domain Multicast Routing Protocols ..........................................................25
    3.2.2 Distance Vector Multicast Routing Protocol (DVMRP) .........................................26
    3.2.3 Multicast Extensions to OSPF (MOSPF) ...............................................................27
    3.2.4 Core Based Trees Protocol (CBT) .......................................................................28
    3.2.5 Protocol Independent Multicast (PIM) ..................................................................28
    3.2.6 Inter-domain Multicast Routing Protocols ............................................................29
      3.2.6.1 Near-term Solutions ........................................................................................30
      3.2.6.2 Long-term solutions .......................................................................................33
      3.2.6.3 Source Specific Multicast (SSM) ....................................................................38
  3.3 Relevance to Satellite Networks ..................................................................................39
  3.4 Summary .......................................................................................................................40

4 Multimedia Conference Model over Satellite ...................................................................42
  4.1 End System Mixing Model ..........................................................................................43
  4.2 Large-Scale Multicast Conferences Model ..................................................................43
    4.2.1 Suitability of Large-Scale Multicast Conferences Model for IP Conference over Satellite ..............................................................44
    4.2.2 Inviting Users to Join in Large-Scale Multicast Conferences Model ....................44
    4.2.3 Scalability of Large-Scale Multicast Conferences Model ....................................45
    4.2.4 Location of Service Logic in Large-Scale Multicast Conferences Model ...............45
  4.3 Dial-In Conference Model ..........................................................................................45
    4.3.1 Suitability of Dial-In Conference Model for IP Conference over Satellite .............46
    4.3.2 Inviting Users to Join in Dial-In Conference Model ..............................................47
    4.3.3 Users Joining in Dial-In Conference Model ..........................................................48
    4.3.4 Scalability of Dial-In Conference Model ...............................................................48
    4.3.5 Location of Service Logic in Dial-In Conference Model .......................................48
  4.4 Ad-hoc Centralized Conferences .................................................................................49
  4.5 Dial-Out Conferences ...................................................................................................49
  4.6 Centralized Signalling with Distributed Media .............................................................49
    4.6.1 Suitability of Centralized Signalling with Distributed Media Conference Model for IP
List of Figures

Figure 2-1: SIP signalling Initiation procedure ................................................................. 13
Figure 2-2: RTP header format ............................................................................................ 17
Figure 3-1: Basic multicast transmission module ............................................................... 23
Figure 4-1: Centralized signalling with decentralized media ............................................... 51
Figure 4-2: MCU architecture example ............................................................................... 54
Figure 4-3: SIP call example with proxy server using INVITE ............................................ 55
Figure 4-4: SIP call example with proxy server using REFER ............................................ 57
Figure 5-1: Star topology of an integration of satellite networks and terrestrial networks .......... 62
Figure 5-2: Multicast hierarchical example ......................................................................... 65
Figure 5-3: PIM-SM deployment of the IP conference over satellite network with satellite terminals provided with multicast router ................................................................. 67
Figure 5-4: Satellite enabled Multicast Router co-located with satellite terminal ................. 68
Figure 5-5: One satellite hop Inter-domain multicast scenario ............................................ 70
Figure 5-6: MBGP/MSDP deployment over the IP conferencing over satellite system with satellite terminals provided with Multicast Router ................................................................. 71
Figure 5-7: The procedure of a new terrestrial unicast sources joining a conference ............. 75
Figure 5-8: The procedure of a new satellite unicast sources joining a conference ............... 76
Figure 5-9: The procedure of a terrestrial unicast receiver joining a conference .................. 77
Figure 5-10: The procedure of a satellite unicast receiver joining a conference ................... 78
Figure 5-11: The procedure of a terrestrial unicast source leaving a conference ................... 79
Figure 5-12: The procedure of a terrestrial unicast receiver leaving a conference ............... 80
Figure 5-13: The procedure of a satellite unicast receiver leaving a conference ................... 81
Figure 6-1: Basic principle of passive measurement ............................................................ 84
Figure 6-2: IP conferencing over satellite demonstrator prototype general layout ................ 91
Figure 6-3: Remote nodes layout ......................................................................................... 92
Figure 6-4: IP conferencing over satellite demonstrator symbols’ table ............................... 92
Figure 6-5: I-FAN for unicast access mode ......................................................................... 93
Figure 6-6: I-FAN for multicast access mode ..................................................................... 93
Figure 6-7: One-way delay measurement scenarios for ICEBERGS ................................... 96
Figure 6-8: Pure terrestrial conference test scenario ........................................................... 105
Figure 6-9: One-way delay for pure terrestrial test scenario ............................................... 106
Figure 6-10: One-way jitter for pure terrestrial test scenario .............................................. 107
Figure 6-11: Two unicast-users conference test scenario ................................................... 109
Figure 6-12: Three-participant-conference scenario .................................................................111
Figure 6-13: Four-participant-conference test scenario ...............................................................113
Figure 7-1: One-to-group measurement scenario example .....................................................119
Figure 7-2: Group-to-one measurement scenario example ......................................................124
Figure 7-3: Class modification trigger function flow chart for Udperv agent ..........................137
Figure 7-4: Receiving packet function flow chart for Udperv agent ........................................138
Figure 7-5: Receiving packet function flow chart for Udprevient agent .................................138
Figure 7-6: Simulation configuration scenario ...........................................................................140
Figure 7-7: configuration of client and server ...........................................................................141
Figure 7-8: RDV simulation result for scenario 1 ......................................................................143
Figure 7-9: RDV simulation result for scenario 2 ......................................................................145
Figure 7-10: RDV simulation result for Scenario 3 .................................................................147
List of Tables

Table 2-1: SIP vs. H.323 ................................................................. 16
Table 2-2: RTP header extension format ........................................ 18
Table 6-1: End-to-end delay for TIPPHON Systems ......................... 98
Table 6-2: Call setup delay for unicast and multicast users ................ 103
Table 6-3: SIP registration and deregistration delay for unicast and multicast users .... 104
Table 6-4: Measurement result for unicast end user to unicast end user .......... 110
Table 6-5: Delay of MCUs in unicast end user to unicast end user scenario ...... 110
Table 6-6: Measurement result for three-participant-conference scenario ......... 112
Table 6-7: MCU delay in three-participant-conference scenario ............... 112
Table 6-8: Measurement result for four-participant-conference scenario .......... 114
Table 6-9: MCU delay in four-participant-conference scenario ............... 114
Table 7-1: Simulation results for scenario 3 ........................................ 146
Table 7-2: Priority manipulations in Scenario 3 ................................. 147
# Glossary of Terms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABR</td>
<td>Area Border Routers</td>
</tr>
<tr>
<td>AFI</td>
<td>Address Family Identifier</td>
</tr>
<tr>
<td>AMP</td>
<td>Active Measurement Program</td>
</tr>
<tr>
<td>AS</td>
<td>Autonomous System</td>
</tr>
<tr>
<td>ASM</td>
<td>Any Source Multicast</td>
</tr>
<tr>
<td>ATM</td>
<td>Asynchronous Transfer Mode</td>
</tr>
<tr>
<td>BD-R</td>
<td>Border Routers</td>
</tr>
<tr>
<td>BGMP</td>
<td>Border Gateway Multicast Protocol</td>
</tr>
<tr>
<td>CAIDA</td>
<td>Cooperative Association for Internet Data Analysis</td>
</tr>
<tr>
<td>CBT</td>
<td>Core Based Trees</td>
</tr>
<tr>
<td>CR-R</td>
<td>CoRe Router</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>DR</td>
<td>Designated Router</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processor</td>
</tr>
<tr>
<td>DVMRP</td>
<td>Distance-Vector Multicast Routing Protocol</td>
</tr>
<tr>
<td>ED-R</td>
<td>EDge Router</td>
</tr>
<tr>
<td>E-NIU</td>
<td>EuroSkyWay Network Interface Unit</td>
</tr>
<tr>
<td>ESW</td>
<td>EuroSkyWay</td>
</tr>
<tr>
<td>F-ISP</td>
<td>Federated Internet Service Providers</td>
</tr>
<tr>
<td>FPS</td>
<td>First Person Shooter</td>
</tr>
<tr>
<td>F-RP</td>
<td>Federated Rendezvous Point</td>
</tr>
<tr>
<td>GEO</td>
<td>Geostationary Earth Orbit</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hyper Text Transport Protocol</td>
</tr>
</tbody>
</table>
| ICEBERGS     | IP Conferencing with Broadband multimedia over Geostationary
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satellites</td>
<td>Internet Control Message Protocol</td>
</tr>
<tr>
<td>ICMP</td>
<td>Internet Draft</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>I-FAN</td>
<td>ICEBERGS Federated Access Network</td>
</tr>
<tr>
<td>IGMP</td>
<td>Internet Group Management Protocol</td>
</tr>
<tr>
<td>IPPM</td>
<td>IP Performance Metrics</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>ITU-T</td>
<td>International Telecommunication Union - Telecommunication Standardization Sector</td>
</tr>
<tr>
<td>JoBS</td>
<td>Joint Buffer management and Scheduling</td>
</tr>
<tr>
<td>JPEG</td>
<td>Joint Photographic Experts Group</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LEO</td>
<td>Low Earth Orbit</td>
</tr>
<tr>
<td>LSA</td>
<td>Link-State Advertisements</td>
</tr>
<tr>
<td>MABR</td>
<td>Multicast Area Border Routers</td>
</tr>
<tr>
<td>MASC</td>
<td>Multicast Address-Set Claim</td>
</tr>
<tr>
<td>MCU</td>
<td>Multipoint Control Unit</td>
</tr>
<tr>
<td>MEO</td>
<td>Medium Earth Orbit</td>
</tr>
<tr>
<td>MG</td>
<td>Media Gateway</td>
</tr>
<tr>
<td>MGC</td>
<td>Media Gateway controller</td>
</tr>
<tr>
<td>MGCP</td>
<td>Media Gateway Control Protocol</td>
</tr>
<tr>
<td>MIGP</td>
<td>Multicast Interior Gateway Protocol</td>
</tr>
<tr>
<td>MLD</td>
<td>Multicast Listener Discovery</td>
</tr>
<tr>
<td>MMUSIC</td>
<td>Multiparty Multimedia Session Control</td>
</tr>
<tr>
<td>MOSPF</td>
<td>Multicast Extensions to Open Shortest Path First</td>
</tr>
<tr>
<td>MPEG</td>
<td>Motion Picture Expert Group</td>
</tr>
<tr>
<td>MRIB</td>
<td>Multicast Routing Information Bases</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>NCC</td>
<td>Network Control Centre</td>
</tr>
<tr>
<td>NCP</td>
<td>Network Connection Provider</td>
</tr>
<tr>
<td>NGI</td>
<td>Next-Generation Internet</td>
</tr>
<tr>
<td>NLANR</td>
<td>National Laboratory for Applied Network Research</td>
</tr>
<tr>
<td>NLRI</td>
<td>Network Layer Reachability Information</td>
</tr>
<tr>
<td>NOC</td>
<td>Network Operation Centre</td>
</tr>
<tr>
<td>NTP</td>
<td>Network Time Protocol</td>
</tr>
<tr>
<td>OBP</td>
<td>On Board Processing</td>
</tr>
<tr>
<td>OSPF</td>
<td>Open Shortest Path First</td>
</tr>
<tr>
<td>PIM-DM</td>
<td>Protocol-Independent Multicast Dense Mode</td>
</tr>
<tr>
<td>PingER</td>
<td>Ping End-to-end Reporting</td>
</tr>
<tr>
<td>PMA</td>
<td>Passive Measurement and Analysis</td>
</tr>
<tr>
<td>PIM-SM</td>
<td>Sparse Mode</td>
</tr>
<tr>
<td>PSTN</td>
<td>Public Switched Telephony Network</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RA</td>
<td>Routing Arbiter</td>
</tr>
<tr>
<td>RAMA</td>
<td>Root Addressed Multicast Architecture</td>
</tr>
<tr>
<td>RDV</td>
<td>Relative Delay Variation</td>
</tr>
<tr>
<td>RFC</td>
<td>Request For Comments</td>
</tr>
<tr>
<td>RIB</td>
<td>Routing Information Bases</td>
</tr>
<tr>
<td>RIPE</td>
<td>Réseaux IP Européens</td>
</tr>
<tr>
<td>RMON</td>
<td>ReMOTe Monitoring</td>
</tr>
<tr>
<td>RP</td>
<td>Rendezvous Point</td>
</tr>
<tr>
<td>RPT</td>
<td>Rendezvous Point Tree</td>
</tr>
<tr>
<td>RTCP</td>
<td>Real-Time Control Protocol</td>
</tr>
<tr>
<td>RTP</td>
<td>Real-time Transport Protocol</td>
</tr>
<tr>
<td>RTT</td>
<td>Round Trip Time</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SA</td>
<td>Source Active</td>
</tr>
<tr>
<td>SAP</td>
<td>Session Announcement Protocol</td>
</tr>
<tr>
<td>SDP</td>
<td>Session Description Protocol</td>
</tr>
<tr>
<td>SIP</td>
<td>Session Initiation Protocol</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreements</td>
</tr>
<tr>
<td>SMP</td>
<td>Satellite Multicast Router</td>
</tr>
<tr>
<td>SMTP</td>
<td>Simple Mail Transfer Protocol</td>
</tr>
<tr>
<td>SNMP</td>
<td>Simple Network Management Protocol</td>
</tr>
<tr>
<td>SPT</td>
<td>Shortest Path Tree</td>
</tr>
<tr>
<td>SRP</td>
<td>Satellite Rendezvous Point</td>
</tr>
<tr>
<td>SSM</td>
<td>Source Specific Multicast</td>
</tr>
<tr>
<td>Sub-AFI</td>
<td>Subsequent Address Family Identifier</td>
</tr>
<tr>
<td>TIPHON</td>
<td>Telecommunications and Internet Protocol Harmonization over Networks</td>
</tr>
<tr>
<td>TTL</td>
<td>Time-To-Live</td>
</tr>
<tr>
<td>UA</td>
<td>User Agent</td>
</tr>
<tr>
<td>UAS</td>
<td>User Agent Server</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>URIB</td>
<td>Unicast Routing Information Bases</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over IP</td>
</tr>
<tr>
<td>WIDE</td>
<td>Widely Integrated Distributed Environment</td>
</tr>
<tr>
<td>WWW</td>
<td>World Wide Web</td>
</tr>
</tbody>
</table>
Publication List

Journal Papers


Conference Papers


Internet Draft

1 Introduction

This chapter will give a brief introduction on my Ph.D research area. Motivations of the research will be presented as well as the objectives. The main research results are listed. The thesis content is outlined.

Internet has grown tremendously in recent decades. The culmination of optic, mobile wireless technology and the advent of the World Wide Web (WWW) have made the Internet accessible to a large population. As a result, real-time multimedia applications emerged such as IP telephony, multiparty conferencing and online gaming, which demand higher bandwidth and high-quality services.

However, the real-time communication based on the Voice over IP (VoIP) technologies still has its own issues. One of the most important problems is Quality of Service (QoS) that the service provider can provide users when the multimedia transmission carrier transfers from circuit switched networks to IP based networks. Due to the dynamic nature of the IP network, information are packetized and transmitted from sources to destinations via routes in a network where they might be dropped and delayed for variable reasons, such as buffering, congestion, hardware function failure and so on. The retransmission technologies and connection-oriented protocols work well for pure data communications but can turn out to be the barrier for real-time communications with QoS support. New technologies were boosted by the QoS requirements of real-time applications in the network, i.e. QoS routing algorithms, real time transport protocols, resource reservation protocols, service differentiation algorithms, variable codecs and adaptive jitter buffers. All of these technologies worked together and tried to provide acceptable QoS to the VoIP users and hope to be competitive to the tradition Public Switched Telephony Network (PSTN) systems.

Though the VoIP technologies are still struggling to fight the PSTN in terms of providing competitive QoS, they can provide many value-added services that the PSTN does not have or cannot provide easily. One of those services is the multimedia multiparty communications including visual and audio signals among a group of people. Two typical applications are multimedia conferencing and online gaming. These services boosted group communication technologies such as multicast and the relevant QoS algorithms. However, the multicast technologies are not that mature and facing many difficulties with deployment in the public Internet. Many possible solutions were proposed, such as Any Source Multicast (ASM) and Source Specific Multicast (SSM). Furthermore, inter-domain multicast is facing more difficult
problems. Different domains have various policies and administration strategies that make them black boxes to each other. Users in the same multicast group will not easily obtain source information to establish the multicast tree if only using the ASM intra-domain multicast technologies.

The physical network support is another key factor for the fast-developed real-time applications. The current Internet architecture appears unable to meet the needs of such new classes of users who are widely distributed around the world. In addition to the terrestrial transmission technologies, satellite networks can play an essential part of the Next-Generation Internet (NGI) due to their ability to deliver high data rates and their global coverage. Satellite links such as GEO (Geostationary Earth Orbit) and LEO (Low Earth Orbit) satellite systems become very attractive for their global coverage. Moreover, satellite communication systems have very flexible bandwidth-on-demand capabilities, which is capable of meeting Quality-of-Service requirements to multimedia services. However, integration of satellite links into existing Internet infrastructure, providing end-to-end QoS support, is a challenging task due to satellite propagation delay and high bit-error-rate.

1.1 Motivations

A satellite network has advantages to provide multiparty multimedia communications services globally due to its wide coverage and high data rates. However, to seamlessly integrate the satellite networks and terrestrial networks, one has to consider the satellite network requirements that are different from terrestrial ones. It is important to investigate these requirements from a system point of view and satisfy them to enable multimedia conferencing service over GEO satellite. It will include proposing a suitable conference model based on Session Initiation Protocol (SIP) and building up the routing architecture for delivering the service. IP network traffic measurement technologies will have to be used to verify and evaluate the designed conferencing system over satellite.

Traffic measurement can provide valuable input to the study to identify existing problems, guide network optimisation, and aid in the prediction of potential performance problems, such as unsatisfied delay and drop ratio.

Traditional measurement technologies can be introduced into the multiparty multimedia communication systems. The issue is that all parameters defined in standard organizations are for one-to-one services. They can not efficiently and accurately represent the performance of a group of users in multiparty multimedia communication services. It is necessary to propose a set of new parameters derived from the existing ones for multiparty multimedia communications. These parameters will also be able to describe the relative QoS requirement of the multiparty multimedia
communications. A relative QoS optimization algorithm will be proposed to satisfy the relative QoS requirement based on a dynamic traffic class assigning scheme using the proposed parameters. This algorithm does not introduce more entities into the network, unlike existing ones.

1.2 Objectives

Multiparty multimedia communication services based on VoIP technologies are playing a more and more important role in our lives. The word “multimedia” in this thesis includes voice, video, data and the mixture of them. Since the works described in this thesis are all focus at the IP layer of the networks rather than the contents delivered on it, all content types will be called multimedia or data for convenience. At the same time, these exciting services also address new network issues and introduce new network requirements. The main objective of this research is to enable an IP conferencing service over GEO satellite by proposing a hybrid routing architecture and a suitable conference model with consideration of the satellite network requirement; to evaluate the designed system using IP traffic measurement technologies; and to study the QoS requirements the multiparty multimedia communications introduced to the network and address and solve the issues by proposing a group of new network performance parameters and a relative QoS algorithm to satisfy the requirement. The word “QoS” means the absolute objective quality of the networks that can be described using appropriate network performance parameters, such as packet delay, packet jitter and packet loss. The perceived QoS is out of the scope of this thesis. The objectives of the thesis include:

i) To study VoIP technologies and understand the QoS requirements of these technologies to support real-time communications. These technologies will include multicast routing and real-time transport protocols, and signalling technologies, which include H.323 and SIP (Session Initiation Protocol). The QoS requirements for real-time multiparty multimedia communications will be addressed in terms of network performance parameters. To study why these parameters are so important for the real-time multiparty multimedia communications.

ii) To propose the network architecture, including the routing architecture and the conference model, with performance evaluation for the provision of IP-based multicast multiparty multimedia conferencing services over an On Board Processing (OBP) satellite. Research includes how a GEO satellite hop can affect the conferencing system topology, how to seamlessly integrate this satellite hop into a multicast enabled terrestrial network that can provide multimedia conferencing service, and propose a SIP based conference model that can efficiently enable the conferencing service over a GEO satellite network.
iii) To study the existing measurement technologies and identify the relative QoS requirements for multiparty multimedia communications. Research is carried out on proposing relevant QoS parameters. Using these new parameters, a QoS algorithm based on dynamic measurements will be proposed to optimise the network to satisfy the requirements.

1.3 Main Contributions

The main contributions of this thesis are summarized here:

i) Addressed the network requirements of general satellite networks to support multiparty multimedia IP conferencing services. The research focused on the star topology of the satellite networks and pointed out the network requirements to support multicast over satellite including the interworking requirements between the satellite network and the terrestrial networks.

ii) Proposed a conference model based on SIP to support multiparty multimedia IP conferencing service over a GEO satellite network. This work was carried out within the IST project ICEBERGS (IP ConferEncing with Broadband multimedia ovBR Geostationary Satellites). My contributions to the model include the SIP signalling procedure design and proposal to use MCUs to enable both unicast and multicast user to access the conference service. This model was designed to cooperate with a unicast/multicast hybrid routing architecture in order to manage IP conferencing services. It avoids any multimedia data being transported over more than one satellite hop.

iii) Proposed a hybrid routing architecture including both unicast and multicast technologies to support the proposed conference model to enable multiparty multimedia IP conferencing service over a GEO satellite network within the ICEBERGS. My specific contribution to the architecture include the proposal of enable both unicast and multicast routing, the one-satellite-hop hieratical multicast design, and the conference procedures design. This hybrid routing architecture provided the ability to enable both unicast and multicast end users to join the conferencing services, which makes it more suitable for commercial implementation in the current stage while multicast is not widely deployed in the Internet. It is also easy for the future transfer to a full multicast architecture. The multicast support will make the usage of the satellite bandwidth more efficient.

iv) Evaluated the proposed designs of multiparty multimedia conferencing service over satellite on a demonstrator with real satellite connections. A detailed test plan has been developed including a functionality test and an objective test. Measurement parameters
and methodologies were designed and carried out on the demonstrator. The evaluation results showed the designed IP conferencing over satellite system functioned well and provided satisfactory QoS.

v) Addressed the relative QoS requirement of multiparty multimedia communications and proposed a set of parameters to present it, which has been submitted as an Internet Draft to IETF and picked up by IPPM working group as a study item. The research shows that all end users involved in a multiparty multimedia communication service require a certain level of similar QoS. The level depends on the applications. The heavier interactive, the higher the requirement is. The proposed parameters are to measure this requirement and present it in an easy and accurate means.

vi) Proposed a relative QoS optimization algorithm and simulation results proved that it can improve the relative QoS of multiparty multimedia communications to satisfy the requirements of various applications. This algorithm is based on dynamic measurement of the proposed multiparty multimedia communications parameters. It has the ability to tune the traffic class for each end user to reduce the difference of QoS among users.

1.4 Thesis Outline

This thesis consists of 8 chapters that are organized as the following.

Chapter 2 studies some important VoIP technologies, including signalling protocols, i.e. H.323, SIP, media transport protocols, i.e. Real-time Transport Protocol (RTP) and Real-Time Control Protocol (RTCP), Media gateway control protocol named Megaco, and some media coding technologies.

Chapter 3 presents multicast routing protocols including both intra-domain and inter-domain multicast technologies that is the foundation of the proposed hybrid routing architecture for the IP conferencing over satellite system.

My work on IP conferencing over a GEO satellite in the context of the IST ICEBERGS project is presented in Chapter 4, Chapter 5 and Chapter 6.

In Chapter 4, conference models will be studied based on SIP. A multiple Multipoint Control Units (MCU) model will be proposed to avoid multiple satellite hops transmission of media data.

Chapter 5 will present a hybrid routing architecture designed with the consideration of satellite networks and the proposed conference model. It enables both unicast and multicast users to join in the conferencing service. The multicast design is presented in details including both inter-domain and intra-domain multicast. It also presents how a conference can be setup step-by-step using this
routing architecture in a general satellite network.

A measurement regime carried out on a demonstrator of the designed system and the result evaluation will be presented in Chapter 6. This measurement regime was carefully planned and executed with positive results.

Chapter 7 will identify the relative QoS requirement of the multiparty multimedia communications and propose a set of network performance parameters and their metrics to measure this requirement. An algorithm will be proposed based on the measurement of these parameters to optimize the relative QoS for multiparty multimedia communications.

Conclusions are drawn in Chapter 8 and possible future work is identified.
Chapter 2. VoIP Technologies

2 Review of VoIP Technologies

VoIP should be viewed as a technology, whereas real-time multimedia applications are the applications that can be enabled by that technology [3]. It allows voice and other multimedia data to be transported across any IP-based network such as campus Local Area Networks (LAN), private or managed intranets, the public Internet, or wireless and satellite communication networks.

VoIP is the real-time delivery of voice and other multimedia data types between two or more parties across networks using the Internet protocols. It also includes the exchange of information required to control this delivery. VoIP offers the opportunity to design a global multimedia communications system that may eventually replace the existing telephony infrastructure.

The Internet has provided a unique infrastructure that is almost everywhere in Europe. In the old circuit-switched network telephony system, every call is established on a circuit that means a fixed physical connection is established and some system resources are allocated to this connection exclusively. No more than two people, a caller and a callee, can share the same resource in terms of bandwidth at the same time. In the IP packet-switched networks, no circuit is established during a call. All of the system resources can be shared by all of the users. All of the information needed to transport is formed into packets that are transferred in the networks. So no resource is fixed to any user. The network may use fewer resources to handle the same traffic as it does in a circuit-switched system. There is another kind of networks named ATM (Asynchronous Transfer Mode) networks standing between the circuit-switched networks and the packet-switched networks. ATM networks use fixed length cells, 53 bytes, and asynchronous time division multiplexing technique to transfer information over virtual circuits and enable. It is out of the scope of this thesis.

Given the history of VoIP, the most prominent challenge for it is to deliver multimedia quality comparable to circuit-switched alternatives, despite the non-deterministic character of the underlying IP network.

But because the current Internet is best effort network other than connection oriented, one like PSTN, the QoS could not be guaranteed while a call is carried over it. That means the voice packets may be lost in the networks, received not in right order by the destination and have variable transmission time. An IP network with QoS support, without significant bandwidth or performance bottlenecks, can delivery excellent quality speech using VoIP.
2.1 Teleconferencing Protocol Suite - H.323

H.323 is the first standard for VoIP recommended by the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T). The first version was published in June 1996 and the improved version in July 2003 [5]. H.323 sets multimedia standards for the existing infrastructure (i.e. IP-based networks). H.323 allows customers to use multimedia applications without changing their network infrastructure.

2.1.1 H.323 Entities

H.323 mainly consists of four network entities. They are Terminal, Gateway, Gatekeeper and MCU.

- **Terminals.** Real-time bi-directional multimedia communication endpoints. It can either be a Personal Computer (PC) or a stand-alone device, running H.323 supported multimedia applications.

- **Gateway.** An H.323 gateway providing connectivity between an H.323 network and a non-H.323 network. It provides the functions of translating protocols for call setup and release, converting media formats between different networks, and transferring information between the networks connected by the gateway.

- **Gatekeeper.** The brain of the VoIP systems. It administrates call control services for H.323 endpoints, such as address translation and bandwidth management.

- **Multipoint Control Unit (MCU).** It provides support for conferences of three or more H.323 terminals. All terminals in the conference establish connections with the MCU. The MCU manages conference resources, negotiates with terminals for the purpose of mining the audio or video coder/decoder (codec) to use, and may handle the media stream.

2.1.2 H.323 Protocol Set

H.323 regulates its communication protocol set to solve the compatibility problems and to allow users to communicate with each other by using different equipments and applications. E.g. it defines how conferencing systems communicate over packet-switched networks that do not guarantee Quality of Service (QoS).

2.1.3 H.323 Benefits

H.323 provides many benefits on control of VoIP networks. Some of the main advantages are listed below:
Chapter 2. VoIP Technologies

- **Codec Standards.** H.323 regulates the standards for compression and decompression of audio and video streams. Equipment from the different vendors will have common supports and can interoperate with each other.

- **Interoperability.** H.323 establishes methods to satisfy the users to eliminate the worries about compatibility by making sure that the receivers can communicate with the sender. It also regulates common call setup and control protocols.

- **Network Independence.** H.323 can run on top of many kinds of network architectures. It has the ability to provide solutions to allow systems to take advantages of the new enhanced networks and their bandwidth management technologies.

- **Platform and Application Independence.** H.323 is not tied to any hardware or operation system. H.323 platforms can be available in many sizes and shapes, including personal computers, dedicated platforms, IP-enabled telephone handsets, and cable TV set-top boxes.

- **Multipoint Support.** By employing multipoint control units (MCUs), H.323-based system can support three or more endpoints conference in a powerful and flexible manner. Multipoint capabilities can be included in other components of an H.323 system.

- **Bandwidth Management.** To solve the big bandwidth requirement of the multimedia traffic, H.323 establishes its own bandwidth management mechanism. Network managers can limit the number of simultaneous H.323 connections within their networks or limit the amount of bandwidth available to H.323 applications. These limits ensure that critical traffic will not be disrupted.

- **Multicast Support.** H.323 supports multicast transport in multiparty conferences. Generally, unicast transport technologies replicate the interested packet and send them to all of the destinations from transmitter in a one-to-one manner, while broadcast sends the packets to all of the terminals in the network without care of the destinations. In contrast, Multicast sends a single packet to a subset of destinations on the network without replication at the source.

- **Scalability.** The members of a conference can have different capabilities. For example, a terminal with audio-only capabilities can participate in a conference with terminals that have video and/or data capabilities. Furthermore, an H.323 multimedia terminal can share the data portion of a multimedia conference with a data-only terminal, while sharing voice, video, and data with other H.323 terminals.

- **Inter-Network Conferencing.** H.323 support common codec technologies from different multimedia conference standards to minimize transcoding delays and to provide optimum performance.
H.323 is the first signalling standard for VoIP applications and was widely used by the industrials before the Session Initiation Protocol (SIP) was developed. H.323 makes it feasible that people can talk on different terminals through IP networks or between IP networks and other non IP-based networks. But on the other hand, the whole standard is very big and complex. It is developed on the logic of the old telephony systems that make it a bit inflexible in dealing with the IP calls.

### 2.2 Session Initiation Protocol (SIP)

The Session Initiation Protocol (SIP) is designed to be a part of the overall Internet Engineering Task Force (IETF) multimedia data and control architecture and was originally developed in the Multiparty Multimedia Session Control (MMUSIC) working group of the IETF [10]. Then it was separated from MMUSIC working group and the SIP working group was organized to continue its development.

SIP is a signalling protocol used to establish sessions over IP networks. It emerges as the protocol of choice for setting up conferencing, telephony, multimedia and other new types of communication sessions such as instant messaging.

Many people consider SIP a powerful alternative to H.323 and say SIP is a more flexible solution, simpler, easier to implement and more suitable to the support of intelligent user devices. It has been developed as a mechanism to establish sessions; it just initiates, terminates and modifies sessions. This means that SIP is extensible, scales, and fits well in different architectures and deployment scenarios.

It is a request-response protocol that closely resembles two other Internet protocols, Hyper Text Transport Protocol (HTTP) [70] and Simple Mail Transfer Protocol (SMTP) [69]. Consequently, SIP integrates well with Internet applications. Using it, telephony becomes another Internet application and can be integrated easily into other Internet services. With other protocols together, it describes the session characteristics to potential session participants. SIP signalling should be considered separately from the media itself because the signalling can pass via one or more proxies or redirect servers while the media stream takes a more direct path.

#### 2.2.1 SIP Entities

Users exchange messages to establish a session. To achieve this, it is necessary to know the users' identifications, which are called SIP addresses identified by a SIP URL that takes a form similar to a mailto or telnet URL (Uniform Resource Locator), i.e., sip:user@host. That URL address can designate an individual, the first available person from a group of individuals or a whole group.
When making a SIP call, a caller first locates the appropriate server and then sends a SIP request. The most common SIP operation is the invitation. Instead of directly reaching the intended callee, a SIP request may be redirected or may trigger a chain of new SIP requests by proxies. Users can register their location/s with SIP servers [10]. SIP defines two basic entities: clients and servers.

- **A client** (also known as User Agent, UA) is an application program that contains both a user agent client and user agent server. The user agent client is an application that initiates the SIP request and the user agent server is an application that contacts the user when a SIP request is received and that returns a response on behalf of the user. The response accepts, rejects or redirects the request.
- **A proxy server** is an intermediary program that acts as both a server and a client for the purpose of making requests on behalf of other clients. A proxy interprets, and, if necessary, rewrites a request message before forwarding it.
- **A redirect server** is a server that accepts SIP requests, maps the address into zero or more new addresses and returns these addresses to the client.
- **A registrar server** is a server that accepts register requests and is typically co-located with a proxy or redirect server and may offer location services.
- **A location server** is a server used to obtain information about a callee’s possible locations.

### 2.2.2 SIP Messaging Syntax

SIP is text based. This allows easy implementation in languages such as Java, TCL and Perl, allows easy debugging, and most importantly, makes SIP flexible and extensible. Except a few differences in character sets, much of the message syntax is identical to HTTP/1.1. However, SIP is not an extension of HTTP.

A SIP message is either a request from a client to a server, or a response from a server to a client: Both types of messages consist of a start-line followed by zero or more headers and is optionally followed by a message body:

```plaintext
message = start-line
        message-header
        CRLF
        [message-body]
```

where

- **start-line**= request-line | status-line

The request line specifies the type of request being issued, while the response line indicates the success or failure of a given request. Message headers provide additional information regarding the request or response that is required for information exchange between users. The message body normally describes the type of session to be established.
2.2.3 SIP Message Headers

Message Headers are included in a request or response in order to provide further information about the message or to enable the appropriate handling of it. There are four main categories of headers that are general, request, response and entity headers.

General headers can be used within both requests and response and contain basic information that is needed for the handling of requests and responses. Some of them are:

- **To**: which indicates the recipient of the request.
- **From**: which is the originator of the request.
- **Call-ID**: which uniquely identifies a specific invitation to a session.
- **Contact**: it provides a URL for use in future communication regarding a particular session.

Request headers apply only to SIP requests and are used to provide additional information to the server regarding the request itself or regarding the client. Some examples are listed below:

- **Subject**: can be used to provide a textual description of the topic of the session.
- **Priority**: to indicate urgency.
- **Authorization**: to enable authentication.

Response headers apply only to response messages and are used to provide further information about what cannot be included in status line. Here are some examples:

- **Unsupported**: which identify those features not supported by the server.
- **Retry-After**: which indicates when a called user will be available.

The purpose of the entity headers is to show the type and format of information included in the message body. Example:

- **Content-Length**: which specifies the length of the message in octets.
- **Content-Encoding**: to indicate additional codings those have been applied to the message body.
- **Allow**: to specify what a called server supports.

2.2.4 Redirect and Proxy Servers

A redirect server does not issue any SIP requests of it own. After receiving a request other than CANCEL, it sends the list of alternative locations and returns a final response with redirection response code of 3XX, which could be any number between 300 and 399, or refuses the request.
A proxy server sits between a user agent client and the far-end user agent server. It accepts requests from a client and forwards them. One of the most important message headers for a proxy server is the Via header field that it is used to indicate the path taken by a request so far. When a request is generated, the originating client inserts its own address into a new Via header and each proxy along the way will do the same, therefore the collection of Via headers provides a map of the path taken through the network by a given request. It is very useful because if a proxy detects its own address on a Via header it indicates that a loop has been created.

A proxy may be stateless or stateful. It is stateless when takes an incoming request, forwards the outgoing request and forgets that anything happened, and it is stateful if remembers incoming and outgoing requests and can act intelligently on subsequent requests and responses related to the same session.

2.2.5 SIP Procedure

SIP is a relatively new standard. Its textual nature makes it simple and flexible to establish a call between conference members. By sending suitable header and parameter value, SIP protocol can exchange terminal’s capability and make further negotiation to decide what media type and codec should be used in a call.

Since VoIP is still immature, it is likely that additional signalling capabilities will be needed in the future. In addition, individual implementations and vendors may want to add additional features. By using protocol extension, SIP is designed so that the client can either inquire about server abilities first or proceed under the assumption that the server supports the extension and then “back off” if the assumption was wrong. Figure 2-1 shows the initiation of a simple IP telephony with a proxy bridge.

![Figure 2-1: SIP signalling Initiation procedure](image)
In Figure 2-1, the steps are shown as followings:

i) Caller sends an INVITE message to Callee that has been forward to the local SIP proxy first.

ii) The proxy enquires the Location Server for the position of Callee and

iii) The proxy gets response.

iv) The proxy forward the INVITE message to Callee

v) Callee returns an OK message.

vi) The proxy forwards this OK message to Caller

vii) Caller issued an ACK message to response.

viii) The ACK message is forwarded to Callee via the proxy.

The signalling procedure finishes at step viii) and two users will talk to each other using a separate media channel.

2.2.6 Integration with Existing Protocols

SIP integrates well with the two main Internet applications: Web and e-mail. SIP integrates with the Web on a number of levels:

First, SIP carries around MIME content, as does HTTP. This characteristic enables SIP to return Web content as a result of a call invitation. As a result, SIP would integrate extremely well with Web browsers.

SIP identifies a user by means of a Uniform Resource Locator (URL), which can be embedded in Web pages or e-mail and by clicking on that URL, it could be possible to initiate calls like a link can access a new Web page.

One of the main features of the Web is its programmability and because SIP looks like HTTP there is the possibility that CGI could be applied to Internet telephony as well.

2.2.7 SIP Benefits

SIP offers a number of key benefits like simplicity, extensibility, modularity, scalability and integration.

- *Simplicity:* It is a simple protocol because SIP encodes its messages as text simply. Moreover it is similar to HTTP so the existing HTTP parsers can be quickly modified for SIP usage.
Chapter 2. VoIP Technologies

- **Extensibility:** It is a key metric for measuring an IP telephony signalling protocol. By default, unknown headers and values in SIP are ignored. If any unknown value arrives to a server it returns an error code and lists the set of features it understands. To further enhance extensibility, numerical error codes are hierarchically organised as in HTTP. Using textual encoding to describe the header fields keeps their meaning self-evident.

- **Modularity:** Internet multimedia communications require many different functions that should have separate and general protocols to be duplicated in other applications. SIP is very modular and encompasses call signalling, user location and basic registration. A key feature of SIP is its ability to separate the notion of a session from the protocol used to invite a user to a session.

- **Scalability:** There are a number of different levels where we can observe scalability: domains (end systems could be located anywhere on the Internet), server processing (transaction through server could be stateful or stateless) and conference sizes (SIP scales to all different conference sizes. Conference coordination can be fully distributed or centralised).

- **Integration:** SIP has the ability to integrate with the Web, e-mail, streaming media applications and protocols.

### 2.2.8 SIP vs. H.323

SIP attempts to keep the signalling as simple as possible. The fact that SIP involves less signalling than H.323 means that calls can be established faster and a rapid call setup is a key requirement of a high quality service.

Furthermore, not only do the SIP messages themselves make SIP powerful, but also the various pieces of information that can be included within messages and responses make SIP a useful protocol. Not only does SIP enable a range of standard information to be included in requests and responses, but it also enables lots of non-standard information to be included. By enabling useful information to be included, it permits the user devices and the users themselves to make various intelligent decisions about call handling.

Some of the differences between SIP and H.323 are shown in the following table:

<table>
<thead>
<tr>
<th>Encoding</th>
<th>SIP</th>
<th>H.323</th>
</tr>
</thead>
<tbody>
<tr>
<td>Textual</td>
<td>Textual encoding typically results in higher bandwidth overhead; as dramatically more bandwidth is consumed by multimedia, this really does not matter; textual encoding is easy to extend, debug and process by text-processing tools</td>
<td>Binary. Debugging binary is more difficult since they are unknown to protocol analysers and unreadable to humans</td>
</tr>
</tbody>
</table>

15
Chapter 2. VoIP Technologies

<table>
<thead>
<tr>
<th>Used in 3GPP</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call set-up delay</td>
<td>1.5 RTT</td>
<td>1.5 RTT</td>
</tr>
<tr>
<td>Complexity</td>
<td>Adequate: HTTP-like protocol</td>
<td>High: use of several different protocols</td>
</tr>
<tr>
<td>Extensibility</td>
<td>The protocol is open to new protocol features</td>
<td>Vendor specific non-standard parameter at predefined positions only</td>
</tr>
<tr>
<td>Architecture</td>
<td>Modular: SIP encompasses basic call signalling, user location and registration; other functions (QoS, directory accesses, service discovery, and session content description) reside in separate orthogonal protocols</td>
<td>Monolithic: The mix of services provided by the H.323 components encompasses capability exchange, conference control, maintenance operations, basic signalling, QoS, registration, and service discovery. Monolithic design makes component updates difficult and expensive</td>
</tr>
<tr>
<td>Instant Messaging Support</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Addressing</td>
<td>Any URL including E-mail address, H.323, http etc.</td>
<td>Host (without username!), gatekeeper-resolved alias (arbitrary case-sensitive string, e.g. E-mail address), telephone numbers</td>
</tr>
<tr>
<td>Transport protocol</td>
<td>UDP and TCP, most implementations use UDP</td>
<td>UDP and TCP, most implementations use TCP. Usage of TCP results in higher call set-up time</td>
</tr>
<tr>
<td>Web-integration</td>
<td>Integration with other Internet services (e.g. a caller may send an E-mail to an unreachable callee)</td>
<td>No</td>
</tr>
<tr>
<td>Service standardization</td>
<td>&quot;Standardize protocols, not services&quot;: Only general interfaces are standardized.</td>
<td>Standardize everything: Well-known services standardized in detail</td>
</tr>
<tr>
<td>Codecs supported</td>
<td>SIP provides the means to advertise the codecs among users rather than specifies any codecs it supports</td>
<td>For audio, G.711 is compulsory. G.722, G.723.1, G.728, and G.729 may also be supported. For video, H.261, H.263 and H.264 are supported.</td>
</tr>
<tr>
<td>Interoperation</td>
<td>Only supported in IP networks. Can interoperate with other networks if appropriate gateway is used</td>
<td>Gateway for interoperation has been standardized</td>
</tr>
</tbody>
</table>

Table 2-1: SIP vs. H.323

The summary is that the primary reason of existence of two non-interoperable signalling protocols is that both ITU and IETF wanted to have protocols meeting their traditions. However, based on the technical aspects discussed in this section, SIP has a higher chance to be widely accepted as the standardized signalling protocol for VoIP.

2.3 Real-time Transport Protocol (RTP)

IP-based networks are best effort. They do not have fixed connection when real-time data such as voice and video are transmitted over it and, thus, the delay, desequencing and jitter will be more serious than for a circuit-switched network that is connection-oriented. These effects will decrease the Quality of Service (QoS) and make VoIP infeasible. To overcome these disadvantages, Real-Time Transport Protocol (RTP) [19] is developed that allows receivers to compensate for the jitter and desequencing introduced by IP networks. RTP can be used for any real-time stream of data,
including VoIP. RTP defines a way to format IP packets carrying isochronous data and including:

- Information on the type of data transported
- Timestamps
- Sequence numbers

RTP does not do anything to affect the behaviour of the IP network. The network can drop, delay or desquence an RTP packet like any other IP packet. It simply allows the receiver to recover from network jitter by appropriate buffering and sequencing, and to have more information on the network so that appropriate corrective measures can be adopted (redundancy, lower rate codecs, etc.)

RTP is typically used on top of User Datagram Protocol (UDP), or similar transport protocols. Both protocols contribute parts of the transport protocol functionality. RTP may not run on top of TCP because it is not adapted for data that is sensitive to delay. [19] recommended an even UDP port to assign to RTP.

When a host wishes to send a media packet, it takes the media, packetizes it, adds the RTP header, and places it in a lower-layer payload. It is then sent into the network, either to a multicast group or unicast to another participant. A RTP header has variable length, including 12 bytes long fixed part followed by variable length data Figure 2-2 shows RTP header format [6].

![RTP header format](image)

- 2 bits are used to indicate the version of the RTP. Now it's version 2.
- A padding bit P indicates whether the payload has been padded for alignment. If it was assigned a value of "1", the last byte of the payload field shows how many padding octets have been appended exactly to the original payload.
- An extension bit X indicates the presence of head extensions after the Contributing Source (CSRC) of the fixed header. Extension use the following format:
Chapter 2. VoIP Technologies

Defined by Profile Length
Header extension

Table 2-2: RTP header extension format

- Four bits are used to indicate Contributing Source (CSRC) Count Code (CC) that tells how many CSRC identifiers follow the fixed header. Usually there is none.

- Marker (M) occupies 1 bit and defined by the RTP profile. H.225.0 shows that for any audio coding that support silence suppression, it must be set to "1" in the first packet of each talk spurt after a silence period.

- Payload type (PT): 7 bits. The payload of each RTP packet is the real-time information contained in the packet. PT in the RTP header is just a number to indicate a corresponding payload type. The format is completely free and must be defined by the application or the profile of RTP in use. Some static payload types are defined in 1889 and in the 'assigned numbers' RFC. For H.323 the reference is H225.

- Sequence number: 16 bits. Start on a random value and is incremented at each RTP packet.

- Timestamp: 32 bits. The clock frequency is defined for each payload, e.g. H.216 payload uses a 90 kHz clock for the RTP timestamp. For most audio codecs the RTP clock frequency is set to 8000Hz. For video, the RTP timestamp is the tick count of the display time of the first frame encoded in the packet payload and for audio, the RTP timestamp is the tick count when the first audio sample contained in the payload was sampled. The clock begins on a random value.

- Four bytes are used for Synchronization Source (SSRC), which identifies the source of a stream of RTP packets that does not depend on network address.

- Another four bytes are used for Contributing Source (CSRC), which identifies sources that have contributed to the combined stream produced by an RTP mixer.

The sequence number and timestamp are used together to help the receiver to retrieve the right order of the received voice packets by manage a reception buffer. A VoIP application puts coming packets in a buffer of a fixed length (decided by analysis sequence number and timestamp) before their playback. In the buffer, RTP packets are sorted in their original order by checking their sequence number. If a packet doesn't arrive on time and still missing at playback time, the application may repeat the last frame of the current playing packet long enough to catch up with the timestamp of the next received packet, or use some interpolation scheme as defined by the
particular audio codec in use.

2.4 Real Time Control Protocol (RTCP)

The data transport protocol RTP is augmented by a control protocol, Real-time Transport Control Protocol (RTCP), to allow monitoring of the data delivery in a manner scalable to large multicast networks, and to provide minimal control and identification functionality. RTP and RTCP are designed to be independent of the underlying transport and network layers. RTCP provides statistics information for monitoring the quality of service of the voice call, and is based on the periodic transmission of control packets to all participants in the session. The underlying protocol must provide multiplexing of the data and control packets. RTCP provides additional information to session participants. In particular, it performs four functions [20]:

- **QoS feedback.** This feedback function is performed by the RTCP sender and receiver reports, and is related to the flow and congestion control functions of other transport protocols. Receivers in a session use RTCP to report back the quality of their reception from each sender. This information includes the number of lost packets, jitter, and round-trip delays. This information can be used by senders for adaptive applications which adjust encoding rates and other parameters based on feedback.

- **Intermedia synchronization.** For flexibility, audio and video are often carried in separate packet streams, but they need to be synchronized at the receiver to provide "lip sync". The necessary information for the synchronization of sources, even if originating from different servers, is provided by RTCP.

- **Session Control.** RTCP allows participants to indicate that they are leaving a session (through a BYE RTCP packet). Participants can also send small notes to each other, such as "out of the office".

- **Identification.** RTCP packets contain information such as the e-mail address, phone number, and full name of the participant to be displayed in the user interface. This allows session participants to learn the identities of the other participants in the session.

RTCP requires that all session participants (including those who send media and those who just listen) send a packet periodically which contains the information described above. These packets are sent to the same address (multicast or unicast) as the RTP media, but on a different port. The information is sent periodically since it contains time-sensitive information, such as reception quality, which becomes stale after some amount of time. However, a participant cannot just send an RTCP packet with a fixed period. Since RTP is used in multicast groups, there could be sessions with hundreds or thousands of participants. If each one were to send a packet with a fixed
period, the network would become swamped with RTCP packets. To fix this, RTCP specifies an algorithm that allows the period to increase in larger groups.

The specification defines several RTCP packet types to carry a variety of control information:

- **SR**: Sender report, for transmission and reception statistics from participants that are active senders. These SRs are sent periodically in RTCP packets to the same control port.

- **RR**: Receiver report, for reception statistics from participants that are not active senders. These RRs contain the user's name and information on the number of packets lost and the interarrival jitter for each source in the conference.

- **SDES**: Source description items.

- **BYE**: Indicates end of participation.

- **APP**: Application specific function.

The only difference between SR and RR forms, besides the packet type code, is that the sender report includes a 20-byte sender information section that contains timestamps, bytes sent, and packets sent on the data port for use by active senders, where packet and octet count are reported. Both the SR and RR forms include zero or more reception report blocks, one for each of the synchronization sources from which this receiver has received RTP data packets. Each reception report block provides statistics about the data received from the particular source indicated in that block that provide information about the QoS behaviour: ratio of packets lost, cumulative number of packets lost, highest sequence number received and interarrival jitter.

It is important to note that, even if the end-station actually is not sending any audio or video data, it is still multicasting periodic RTCP packets and, therefore, is a multicast source and receiver.

### 2.5 Relevance to Satellite Networks

To design a signalling system for multiparty multimedia communication services over satellite, one has to consider the impact of the satellite propagation delay and star network topology. Signalling between users should avoid multiple satellite hop transmission if possible to minimize the signalling delay. A centralized signalling relationship between all users and a signalling management server in a multiparty multimedia communication service is clearly more efficient than a fully meshed signalling system where each user will have to establish signalling relationship with the rest users. Satellite channels should be established permanently or temporarily to carry the signalling data for the services. If temporary channels are used, satellite earth stations should understand signalling protocols, such as SIP, in order to initialize and release satellite channels for them. If permanent channels are used, satellite earth stations should
Chapter 2. VoIP Technologies

recognized signalling packets and transmit them using the right channels.

2.6 Summary

This chapter presented several main technologies developed for VoIP. Two signalling protocols, H.323 and SIP, are introduced and compared with each other. H.323 is the first standard for VoIP recommended by ITU-T. It consists of a set of protocols to solve the compatibility problem and allow users to communicate with each other by using different equipments and applications. It is independent of any specific of network architectures and enables users to communicate in different networks. It also made the multiparty communication practical with support of multicast. However, it is developed on the logic of the old telephony systems that makes it a bit inflexible in dealing with the IP calls while the SIP protocol is playing better. SIP is designed to be a part of the overall IETF multimedia data and control architecture. It is used to establish sessions over an IP network for many types of communications. It is a more flexible solution, simpler, easier to implement and better suited to the support of intelligent user devices. It is extensible, and fits well in different architectures and deployment scenarios. It is a request-response protocol, like HTTP, and integrates well with Internet applications. SIP involves less signalling than H.323 that means that calls can be established faster and a rapid call setup is a key requirement of a high quality service. By enabling useful information to be included, it also permits the user devices and the users themselves to make various intelligent decisions about call handling. Table 2-1 shows the details of comparing H.323 and SIP.

Real-Time Transport Protocol (RTP) is developed that deliver media from user to user and allows receivers to compensate for the jitter and desequencing introduced by IP networks. RTP can be used for any real-time stream of data, for instance VoIP. RTP defines a way to format IP packets carrying isochronous data for VoIP applications. RTP does not do anything to affect the behaviour of the IP network. It simply allows receivers to recover from the network jitter by appropriate buffering and sequencing. The data transport protocol RTP is augmented by a control protocol, Real-time Transport Control Protocol (RTCP), to allow monitoring of the data delivery in a manner scalable to large multicast networks, and to provide minimal control and identification functionality.
3 Multicast Routing Protocols

To enable multiparty multimedia communications services, routing protocols have to be introduced to assign appropriate paths in the network on which information generated by those applications will be transported. In addition, the multiparty communication implies that these paths need to be built up efficiently among a group of users that is different with the traditional one-to-one path that connects only two users at its two ends. The technology developed for this requirement is Multicast and the paths established are multicast tree.

3.1 Introduction to Multicast

IP Multicast is an Internet protocol that enables transmission of data packets to a group of receivers. IP multicast makes efficient use of bandwidth by setting up a middle point between unicast traffic (one-to-one) and broadcast IP traffic (one-to-all in a network). This is well suited for one-to-many or many-to-many bulk data transfer or multimedia (audio/video) streaming transmission to a large number of heterogeneous receivers. IP Multicast efficiently supports this type of transmission by enabling sources to transmit a single copy of a message to a group of interested receivers.

This mode of transmission scales well with increasing number of receivers, unlike in the unicast case (one-to-one), where the source has to send an individual copy of a message to each interested receiver (limited by bandwidth from sender). IP multicast is also more efficient than IP broadcasting (one-to-many), since in broadcasting a copy of a message is sent to all receivers, including receivers who may not want to receive the message. Moreover, in the broadcasting case messages are limited to a single subnet (to avoid flooding the entire Internet) compared to the multicast case (where receivers choose to join/leave different groups as they wish).

A multicast datagram is delivered to all members of its destination host group with the same best-effort reliability as regular unicast IP datagrams. The membership of a host group is dynamic; that is, hosts may join and leave groups at any time. There is no restriction on the location or number of members in a host group. A host may be a member of more than one group at a time. In addition, a single group address may have more than one data stream on different port numbers (or different sockets in more than one application at the application layer).
For each host group, a multicast address is allocated. Users can have group memberships by joining particular multicast groups. The membership and other information of each group is processed and maintained across the entire WAN or Internet. A multicast tree is introduced to establish and maintain the fabric of the multicast network.

In order to support native IP multicast, both the sending and receiving nodes and the network infrastructure between them must be multicast enabled, including the intermediate routers. Native IP multicast at an end host requires support for IP multicast and delivery of data packets using the TCP/IP protocol stack.

3.1.1 Multicast Addressing

Multicast-Internet addresses have been introduced for IP multicast to define multicast host groups. These are class D addresses and their High order bits of the 1st Octet are “1110”. This means that all IP Multicast-group addresses will fall in this range: 224.0.0.0 - 239.255.255.255. There are reserved link-local addresses from 224.0.0.0 to 224.0.0.255, which are used by network protocols on a local network segment and packets with them will never be forwarded by router. These reserved addresses are always transmitted with a Time-To-Live (TTL) of 1.

This address range is only for the group address or destination address of IP Multicast traffic. The source address for multicast data is always the unicast source address. Sources send out their data to the multicast host group address that they have joined in and receivers listen on the group address for incoming packets.

3.1.2 Multicast Scope

The term scope refers to the region in which the data unit is forwarded. The scope of IP multicast can be unlimited. However, some algorithms have been employed to limit multicast scope for:

- Limitation of flooded network regions
• Multicast address reuse
• Privacy

Some multicast routing protocols use broadcast to initiate multicast tree, such as Distance-Vector Multicast Routing Protocol (DVMRP). Limitation scope can prevent the temporary flooding over the whole network. Due to the limitation of the multicast address resources, multicast address reuse is in demand. Because of scope limitation, multicast can be used multiple times as long as the domain of the groups does not overlap. Finally, by limiting the scope of the multicast groups, it can be helpful to guarantee a certain degree of privacy, e.g. users out of the scope cannot join the multicast group. There are two main mechanisms used for scoping:

• Scoping based on the TTL value
• Administrative scoping

The TTL parameter is used to specify how many routers the packet can pass before being dropped. Therefore, the maximum lifetime of an IP packet can be defined when it comes in the network.

Administrative scoping mechanism has not been used widely. This mechanism defines the scope of the multicast group by specifying groups of multicast address for different administrative regions. Only members of an administration region can join the corresponding group.

3.1.3 Internet Group Management Protocol (IGMP)

IGMP [24] is a protocol that gives a host the ability to support multicasting. It works between a host and the immediately neighbouring multicast router and the router will use multicast routing protocol to establish or join a multicast tree connecting to the source. The current IETF IGMP is version 3.

IGMP is used to manage the multicast groups. It enables the multicast router to track the membership information by using two types of IGMP massages: host membership query and host membership report. Host membership query is sent out periodically by multicast router to determine which multicast group has members on the local network. The query is sent to 224.0.0.1 (all multicast group members in local network) and hosts will generate a corresponding host membership reports to indicate the router to which multicast group they belong. Hence, a multicast router can establish a table to record the relationships of all hosts and groups. When a host wants to join a multicast group, it immediately transmits a join-group report for that group rather than waiting for a query. When a host wants to leave a group, it sends a leave-group report to the multicast router. IGMP can support more management services, e.g. source filtering and so on.
3.2 Multicast Routing Protocols

IGMP can only provide management services between hosts and the nearest router. The multicast router has to employ some routing protocols to establish and maintain the connection between senders and receivers.

Unlike IP routing (where routing table information, stored in routers, is used to determine optimal transmission paths for forwarding messages), IP Multicast routing is much more complex. In IP multicast, the sender is not concerned with the number or location of clients. The network delivers data to all group members and minimise transmission to parts of the network where there is no receiver interest. To do this, the multicast capable designated routers construct a spanning-tree (delivery tree), replacing the simple path in unicast, which is routed at each sender to the group. The spanning-tree approach ensures that there is only one path between every pair of routers and it is free of loops. Routers located at the branches duplicate the incoming messages and send copies down the branch where there are group members.

There are numbers of solutions to do this. In the following sections we provide an overview of multicast routing protocols. Basically, all multicast routing protocols can be categorized into two big classes. One is intra-domain protocols and the other is inter-domain protocols.

3.2.1 Intra-domain Multicast Routing Protocols

This group of protocols is called Multicast Interior Gateway Protocols (MIGPs) that are used to enable multicast routers to implement multicast communication within an Autonomous System (AS). Examples of such protocols are Distance-Vector Multicast Routing Protocol (DVMRP), Multicast Extensions to Open Shortest Path First (MOSPF), Protocol-Independent Multicast Dense Mode (PIM-DM) and Sparse Mode (PIM-SM), and Core Based Trees (CBT).

These protocols can generally be subdivided into three categories, sparse mode, dense mode and link-state protocols.

Dense mode assumes that most subnets in the network will be interested in multicast traffic. To inform other routers of multicast sources, it floods the multicast traffic to all routers in the network. A router with no receivers interested in this traffic will then tell its upstream router to stop forwarding this traffic or to prune this branch from the tree. This flood-and-prune mechanism allows these protocols to easily build a multicast distribution tree rooted at the source. A source-based tree guarantees the shortest and most efficient path from source to receiver. While this may be an ideal enterprise solution in many circumstances, the reliance on broadcast and flooding across the Internet simply will not scale. Examples of dense mode protocols are Distance Vector Multicast Routing Protocol (DVMRP) and Protocol Independent Multicast Dense Mode (PIM-
Sparse mode protocols implement a shared distribution tree. Here, the multicast distribution tree is rooted at a core router in the network called a Rendezvous Point (RP). When a source begins actively sends multicast traffic, its directly connected router, or designated router, registers with the RP. The RP will keep track of all active sources in a domain. When a router is connected to a host that wants to receive a multicast group, it will use RPF (Reverse Path Forwarding) to determine the shortest path to the RP. While the RP builds a tree to the source, all receivers join the tree at the RP. As long as all routers know which router is the RP, broadcast is not needed to distribute multicast routing information. Additionally, this limits the amount of routing state that all non-RP routers need to know. Protocol Independent Multicast Sparse Mode (PIM-SM) and Core-Based Trees (CBT) are examples of a sparse mode routing protocols.

Link-state protocols such as Multicast Open Shortest Path First (MOSPF) function much like dense mode protocols in that they both use shortest path trees to distribute multicast traffic to the receivers in the network. Link-state protocols, however, do not use the flood and prune mechanism that is used in DVMRP or PIM-DM. Instead, they flood special multicast, link-state information that identifies the whereabouts of group members (that is, receivers) in the network. All routers in the network use this group membership information to build shortest path trees from each source to all receivers in the group.

Below a short description of the mentioned protocols is provided.

### 3.2.2 Distance Vector Multicast Routing Protocol (DVMRP)

DVMRP is a source-based protocol using Reverse-Path Multicasting algorithm. To establish connections between multicast routers, the sender broadcast the first datagram that is forwarded to the entire network as long as the TTL of the packet and router interface thresholds allow. If the router has attached hosts belonging to the same multicast group, it duplicates the packet and sends to all of these hosts. If there is no this kind of hosts, the router sends back a prune message to its parent node. Hence, its parents will not send the same multicast pair (source, group) to it. When a member of a new group on a particular link appears, a cancellation message to undo the prune message is sent upstream by the router. All of the multicast routers exchange their routing table update message periodically with their neighbours. The exchanged information is used to establish their multicast routing tables. This technique to create multicast trees is known as broadcast-and-prune. Broadcast-and-prune protocols are also known as dense mode protocols, because they are designed to perform better when the topology is densely populated with group members. Routers assume there are group members downstream, and so forward packets. Only when explicit prune
messages are received does a router not forward multicast traffic. If a group is densely populated, routers are unlikely to ever need to prune.

DVMRP is relatively easy to employ because it is based on simple flooding algorithms. Another advantage is the low processing demands it places on the routers. However, it may cause bandwidth limitation for it's periodically broadcasting flood multicast traffic through the entire network to rebuild multicast trees. Additionally, DVMRP is constrained by the unicast routing protocols underlying.

3.2.3 Multicast Extensions to OSPF (MOSPF)

Multicast Extensions to OSPF (MOSPF) is also based on unicast routing protocol OSPF to provide multicast. OSPF can construct a database for each router called link state database, which describe the network topology of the AS. Routers use this database to calculate the shortest path from the source and forward those packets arrival from the shortest path only. MOSPF add a new group membership Link-State Advertisements (LSA) to OSPF to allow routers to flood the membership information collected using IGMP throughout the AS. This information is used to construct the link state database and to compute a shortest path tree for multicast: basically, MOSPF routers flood an OSPF area with information about group receivers. This allows all MOSPF routers in an area to have the same view of group membership. In the same way that each OSPF router independently constructs the unicast routing topology, each MOSPF router can construct the shortest path tree for each source and group. While group membership reports are flooded throughout the OSPF area, data is not. MOSPF is something of an oddity in terms of classification. It can be considered a dense mode protocol because membership information is broadcast to each MOSPF router, but it can be also considered an explicit join protocol because data is only sent to those receivers that specifically request it. The key to understanding MOSPF is to realize that it is heavily dependent on OSPF and its link state routing paradigm.

MOSPF routes multicast traffic inside a single OSPF area. However, MOSPF also provides mechanism to forward multicast packets between OSPF areas. The way that MOSPF handles inter-area multicast routing is in many ways similar to OSPF's handling of unicast. In OSPF, routers that connect a second-tier area to the backbone area are called Area Border Routers (ABR) and are responsible for forwarding routing information (primarily in the form of OSPF summary LSAs - which summarize the networks inside an area) and unicast traffic between the two areas. Area Border Routers do not pass router or network Link-State Advertisements between areas; therefore, the topology of one area is not seen by a bordering area. To support inter-area multicast, [25] defines inter-area multicast forwarders, which are a subset of the OSPF ABRs in the network and are configured to perform multicast-related tasks, such as summarizing group membership.
into the backbone area and forwarding multicast packets between areas. This subset is called Multicast Area Border Routers (MABRs).

MOSPF can forward normal unicast IP traffic at the same time it handles multicast traffic. It more rapidly adapts to changes of group membership and availability of network resources. Unlike DVMRP, MOSPF does not need flood packets to determine group membership or network topology. By using the same flexible-path metrics as OSPF instead of just a simple hop-count to create source-based tree, MOSPF is more convenient to support cost-based or QoS-based routing, for example. However, MOSPF does not support tunnels and the router needs a shortest path computation for all multicast source. Also, it is not well suited for handling sparse topologies due to the group information flooding and the difficulty to interwork with other domains that do not run OSPF for unicast.

3.2.4 Core Based Trees Protocol (CBT)

The Core Based Trees (CBT) [26][27] protocol is now being standardized by the IETF. CBT uses the basic sparse mode paradigm to create a single shared tree used by all sources. The tree is rooted at a core. All sources send their data to the core and all receivers send explicit join messages to the core. There are two differences between CBT and PIM-SM. First, CBT uses only a shared tree, and is not designed to use shortest path trees. Second, CBT uses bi-directional shared trees, but PIM-SM uses unidirectional shared trees. Bi-directional shared trees involve slightly more complexity, but are more efficient when packets travelling from a source to the core cross branches of the multicast tree. In this case, instead of only sending traffic “up” to the core, packets can also be sent “down” the tree. While CBT has significant technical merits and is on par technically with PIM-SM, few routing vendors provide support for CBT. The reason seems to be that the vendor community was only going to support one sparse mode protocol and the implicit selection was PIM-SM.

3.2.5 Protocol Independent Multicast (PIM)

PIM has two modes of operation: PIM-DM, which employs a reverse shortest path algorithm similar with DVMRP, and PIM-SM, which uses unidirectional-shared trees. The main difference between PIM-DM and DVMRP is that DVMRP is based on the particular unicast routing mechanism, e.g. RIP, while PIM-DM does not need any specification of underlying unicast protocol. DVMRP routers may selectively forward packet to its downstream interfaces so that the next node will recognize that the local node is on the shortest path between it and the source. While PIM-DM router broadcasts the received packet to all of its downstream interfaces.

PIM-SM is similar to PIM-DM in that routing decisions are based on whatever underlying unicast
Chapter 3. Multicast Routing Protocols

routing table exists, but the tree construction mechanism is quite different. PIM-SM's tree construction algorithm is actually more similar to that used by CBT than to that used by PIM-DM. PIM-SM provides rendezvous points (RPs) for receivers to meet new sources. RPs are used by sender to announce their existence. Routers explicitly send a join messages to RP when its local or downstream host want to join a multicast group.

Processing of this message by intermediate routers set up the multicast tree branch from RP to the host. If a source wants to send data to a multicast group, it just sends register message, piggybacked on the data packet, to RPs for that group. Then the RP responds by sending join message to the source. Processing of these massages by intermediate routers establishes a packet delivery path from the source to the RP. PIM-SM avoids flooding packets over ASes for it’s based on receiver initiation tree and centralized RPs. However, those RPs may be the network traffic bottles and the potential single point failures.

3.2.6 Inter-domain Multicast Routing Protocols

This kind of protocols is employed by boarder routers, which interconnect two ASes, to enable multicast communication between each AS. Border Gateway Multicast Protocol (BGMP) is an example. BGMP provides a method for providers to distinguish which route prefixes they will use for performing multicast RPF checks. It contains two components: MIGP component and BGMP component. MIGP component is used by border router to communicate with the others intra-routers in the same AS while BGMP for construction of a bi-directional centre-tree. The root of this tree is an AS that claims a multicast address by employing a global multicast address allocation protocol such as the Multicast Address-Set Claim (MASC) protocol. BGMP use TCP as its transport protocol. Border routers exchange BGMP messages over TCP connection across them to update routing tables by sending join/prune messages. The shortest path between an AS and the source may be different with the one crossing the root of shared tree. Hence, BGMP allows router attach a source-specific branch to the shared tree, i.e. choose the shortest path instead of using tree root. The main advantage of BGMP is that an Internet can support non-congruent unicast and multicast topologies. When the unicast and multicast topologies are congruent BGMP can support different policies for each. BGMP provides a scalable policy based inter-domain routing protocol.

Inter-domain multicast has evolved out of the need to provide scalable, hierarchical, Internet-wide multicast. Protocols that provide the necessary functionality have been developed, but the technology is relatively immature. The particular inter-domain solution in use is considered near-term, and is possibly only an interim solution: while the solution is functional, it lacks elegance and long-term scalability. As a result, additional work is underway to find long-term solutions.
Some of these proposals are based on the standard IP multicast model. Others attempt to refine the service model in hopes of making the problem easier. These new protocols, which are being considered by the IETF, promise to provide solutions to the complex issues of inter-domain multicast routing.

When the topic of extending inter-domain multicast beyond the outdated DVMRP-based Mbone was first discussed by several ISPs, it was agreed that some sort of sparse mode protocol was necessary. It seems that it would be preferred PIM-SM. However, this choice introduced some problems because it is very problematical to interconnect multiple sparse mode domains. The requirement of PIM-SM that there can only be one active RP for a given group presents a significant challenge in the inter-domain world of the Internet. For practical reasons, ISPs do not want to rely on shared or third parties RPs. ISPs require administrative control of their own RPs. A mechanism was needed for these different RPs to be able to communicate with one another to exchange information about the active sources in their respective domains. An early method of doing this was to have multicast peering exchanges on multi-access interfaces to which each ISP would connect its RP. By running PIM-DM on this multi-access interface, each RP would flood its source information to each other only on this interface. Among other problems with this hybrid PIM-SM/PIM-DM architecture was that each ISP would have to place its RP at the edge of the network to achieve this. As we will later examine, a well-designed PIM-SM implementation requires that the RP be well connected in the core of the network.

3.2.6.1 Near-term Solutions

Information given by industrial partners in the ICEBERGS project shows the following major requirements that must be met before they could consider deployment of native multicast in the Internet to be feasible:

1. An Explicit Join protocol inside the domain for efficiency
2. Use of an existing (unicast) operations model for multicast peering
3. Not dependent on competitor’s RPs
4. Flexibility regarding RP placement

The first item was, in general, already met by normal PIM-SM operation. However, at that time, the last three items required the development of two solutions: Multiprotocol BGP (MBGP) [29] and Multicast Source Discovery Protocol (MSDP) [30].

Multiprotocol Border Gateway Protocol (MBGP)
It creates extensions to the widely used Border Gateway Protocol (BGP). Route aggregation and abstraction, as well as hop-by-hop policy routing, are provided in unicast using the BGP. BGP offers substantial abstraction and control among domains. Within a domain, a network
Chapter 3. Multicast Routing Protocols

administrator can run any routing protocol desired. Routing to hosts in an external domain is simply a matter of choosing the best external link. BGP supports inter-domain routing by reliably exchanging network reachability information. This information is used to compute an end-to-end distance-vector-style path of AS numbers. Each AS advertises the set of routes it can reach and an associated cost. Each border router can then compute the set of ASes that should be traversed to reach any network. The use of a distance vector algorithm together with full path information allows BGP to overcome many of the limitations of traditional distance vector algorithms. Packets are still routed on a hop-by-hop basis, but less information is needed and better routing decisions can be made. The functionality provided by BGP, and its well-understood paradigm for connecting ASs, are important catalysts for supporting inter-domain multicast. A version of BGP capable of carrying multicast routes would not only provide hierarchical routing and policy decisions, but would also allow a service provider to use different topologies for unicast and multicast traffic. The mechanism by which BGP has been extended to carry multicast routes is called Multiprotocol Extensions to BGP4 (MBGP). MBGP is able to carry multiprotocol routes by adding the Subsequent Address Family Identifier (SAFI) to two BGP4 messages: MP_REACH_NLRI and MP_UNREACH_NLRI. Specifically for multicast, the SAFI field can specify unicast, multicast or unicast/multicast. With MBGP, instead of every router needing to know the entire flat multicast topology, each router only needs to know the topology of its own domain and the paths to reach each of the other domains. While MBGP is the first step toward providing inter-domain multicast, it alone is not a complete solution. MBGP is capable of determining the next hop to a host, but it is not capable of providing multicast tree construction functions. More specifically, what is the format of the join message? When should join messages be sent, and how often? Support for this functionality is not provided by MBGP; a true inter-domain multicast routing protocol is needed. Furthermore, conventional wisdom suggests that this protocol should not use the broadcast-and-prune method of tree construction. The near-term solution being advocated is to use PIM-SM, to establish a multicast tree between domains containing group members.

Multicast Source Discovery Protocol (MSDP)
PIM-SM is to be used to connect receivers and sources across domain boundaries. But, there is still one function missing from the near-term solution. This function is needed when trying to connect sparse mode domains together. Given that PIM-SM is the only sparse mode protocol that has seen significant deployment, this function tends to be heavily influenced by PIM-SM. The problem is basically how to inform an RP in one domain that there are sources in other domains. The underlying assumption here is that a group can now have multiple RPs. However, the reality is that there is still only one RP per domain, but now multiple domains may be involved. The approach adopted is largely motivated by the perceived needs of the ISP community. In fact, the
decision to have multiple RPs rather than a single root is what differentiates the near-term solution from other proposed solutions. A problem arises when group members are spread over multiple domains. There is no mechanism to connect the various intra-domain multicast trees together. While traffic from all the sources for a particular group within a particular domain will reach the group's receivers, any sources outside the domain will remain disjoint. Why is this the case? Within a domain, receivers send join messages toward one RP and sources send register messages to the same RP. However, there is no way for an RP in one domain to find out about sources in other domains using different RPs. There is no mechanism for RPs to communicate with each other when one receives a source register message.

The decision to maintain a separate multicast tree and RP for each domain is driven by the need to reduce administrative dependencies between domains. Two potential problems are avoided in this way.

1. It is not necessary for two domains to co-administer a single sparse mode cloud. Relevant administrative functions include identifying candidate RPs and establishing the group-RP mapping.

2. It becomes possible to avoid multi-party dependencies, in which multicast delivery for sources and groups in one or more domains is dependent on another domain whose only function is to provide the RP. Dependencies can occur when all sources and receivers in the RP's domain leave or become inactive. The domain with the RP has no group members and yet is still providing the RP service. Depending on how multicast and inter-domain traffic billing is handled, this could be particularly undesirable.

The near-term solution adopted for this problem is a new protocol, appropriately named the Multicast Source Discovery Protocol (MSDP). This protocol works by having representatives in each domain announce to other domains the existence of active sources. MSDP is run in the same router as a domain's RP (or one of the RPs). MSDP's operation is similar to that of MBGP, in that MSDP sessions are configured between domains and TCP is used for reliable session message exchange. MSDP operation is described below.

1. When a new source for a group becomes active it will register with the domain's RP.

2. The MSDP peer in the domain will detect the existence of the new source and send a Source Active (SA) message to all directly-connected MSDP peers.

MSDP message flooding:

- MSDP peers that receive an SA message will perform a peer-RPF check. The MSDP peer that received the SA message will check to see if the MSDP peer that sent the message is along the "correct" MSDP-peer path. These peer-RPF checks are
necessary to prevent SA message looping.

- If an MSDP peer receives an SA message on the correct interface, the message is forwarded to all MSDP peers except the one from which the message was received. This is called peer-RPF flooding.

3. Within a domain, an MSDP peer (also the RP) will check to see if it has state for any group members in the domain. If state does exist, the RP will send a PIM join message to the source address advertised in the SA message.

4. If data is contained in the message, the RP then forwards it on the multicast tree. Once group members receive data, they may choose to switch to a shortest path tree using PIM-SM conventions.

5. Steps 3-5 are repeated until all MSDP peers have received the SA message and all group members are receiving data from the source.

The issue of scalability is an important one to consider for MSDP. Because of the way MSDP operates, if multicast becomes tremendously successful, the overhead of MSDP may become too large. The limitation occurs if multicast use grows to the point where there are thousands of multicast sources. The number of SA messages (plus data) being flooded around the network could become very large. The generally-agreed-upon conclusion is that MSDP is not a particularly scalable solution, and will likely be insufficient for the long term. But, given that long-term solutions are not ready to be deployed, MSDP is seen as an immediate solution to an immediate need.

The short-term inter-domain solution just described is referred to with the abbreviations for the three relevant protocols: MBGP/PIM-SM/MSDP. This solution is relatively straightforward once a person understands all the abbreviations and understands the motivating factors that drove the design of the protocols. While some argue that the current set of protocols is not simple, it really is no more complex than many other Internet services, such as unicast routing. The key advantage of MBGP/PIM-SM/MSDP: it is a functional solution largely built on existing protocols. Furthermore, it is already being deployed with a fair amount of success. The key disadvantage is that, as a long-term solution, the MBGP/PIM-SM/MSDP protocol suite may be susceptible to scalability problems.

3.2.6.2 Long-term solutions

Although the combination of PIM-SM, MSDP, and MBGP is allowing many ISPs to deploy native inter-domain multicast service in their networks, there is still a need to continue researching and developing better solutions. Numerous such efforts are underway. These efforts can be broken down into two groups: efforts based on the standard IP multicast philosophy, and
Chapter 3. Multicast Routing Protocols

efforts which look to change this model in hopes of simplifying the problem.

Border Gateway Multicast Protocol (BGMP) / Multicast Address Set-Claim (MASC)
The BGMP [31] and its associated protocol, MASC [32], can form architecture for inter-domain multicast routing.

MASC forms the basis for hierarchical address allocation architecture. MASC temporarily and dynamically allocates multicast address ranges to domains using a "listen and claim" approach with collision detection. In this approach, child domains listen to multicast address ranges selected by their parent, select sub-ranges from their parents' range and propagate the claims to their siblings. The "claimers" wait for a suitably long period to detect any collision before communicating the acquire range to (1) the domain's multicast address allocation server and to (2) other domains, through BGP, as multicast-specific routes. MASC can then allocate, from its multicast address range, individual multicast addresses to groups initiated in their domain.

BGMP requires that each multicast group be associated with a single root or core and constructs a shared tree of domains, similarly to other shared tree protocols (e.g., PIM-SM and CBT). However, in BGMP, the root is an entire domain rather than a single router. BGMP is based on two main assumptions:

- That a rendezvous mechanism, whereby members get to know the identity of the sources without the need for global broadcast, is the most convenient for inter-domain multicast routing.
- The specific ranges of the class D space are associated (e.g., via MASC) with various domains. Each of these domains is chosen to be the root of the shared tree of domains for all groups whose address is in its range. This is because the root domain is very likely to be the domain initiator of the multicast group.

The actual number of multicast addresses, claimed by a domain using MASC, is a trade-off between two competing factors:

- If the number of multicast addresses available is high, the domain will become the root domain for a large number of groups.
- If the claimed address range is sufficiently large, groups initiated locally can get multicast addresses from the domain range, thereby becoming locally rooted.

The choice of the root of a shared tree in inter-domain routing has implications both in terms of policy and performance as it relates to end-to-end delay. In the intra-domain case, any router can be entitled to become core for the group. This is because the emphasis in the intra-domain case is on load sharing and the penalty on non-optimally located cores is not significant. The same cannot be said in the inter-domain case, that is, all possible root domains cannot be treated as eligible
candidates. In inter-domain routing, there are administrative issues concerning the ownership of the root domain and a greater risk from poor delay performance due to the location of the root. This is the reason why in BGMP, the root domain has been chosen to be the domain of the group initiator in the hope that this domain will source a significant portion of the multicast data.

BGMP uses the routes advertised by BGP to construct the multicast trees for active multicast groups. Since inter-domain routing involves the use of resources in autonomously administered domains, the routing policy constraints of such domains need to be accommodated. BGMP follows policy for multicast traffic using the selective propagation of group routes in [29].

BGMP runs on domain border routers and constructs a bi-directional shared tree that connects individual multicast trees built in a domain. Hence, the border routers also run protocols for multicast routing intra-domain (e.g., PIM-DM, PIM-SM). Such intra-domain routing protocols are also referred to as Multicast Interior Gateway Protocols (MIGP). The module of the border router that runs an MIGP is referred to as the MIGP component; the module running BGMP is referred to as the BGMP component. It is up to the MIGP component to inform the BGMP component about group membership in the domain. This triggers BGMP to send "Joins" and "Prunes" border router to border router until the root domain or a border router that is already on the tree.

In BGMP, the receiver domain is allowed to build source-specific uni-directional inter-domain distribution branches. However, such branches are not allowed to collide with the shared tree, for the sake of loop avoidance and possible introduction of duplicate packets. The need to construct such branches arises when the shortest path from the current domain to a source domain does not coincide with the bi-directional shared tree from the domain. This feature is very useful for domains running MIGPs, such as DVMRP and PIM-DM which support only source-based trees within the domain and only accept source traffic if it arrives from the shortest path back to the source (RPF check). The trick used by the ingress border router is to encapsulate the packets to the appropriate RPF-compliant border router, from where the packets can be injected into the domains' MIGP. If a source-specific branch is constructed, data is sourced into the domain via the appropriate border router avoiding the data encapsulation overhead. Source-specific branches differ from source-specific shortest path trees built by some MIGPs in that the source-specific branch stops where it reaches either a BGMP router on the bi-directional tree or the source domain. In shortest path trees, the source-specific state is set up all the way back to the source. It is also assumed that, since the inter-domain topology is sparser than the intra-domain topologies, the traffic concentration aspects related to the shared trees are not too much of a penalty for the protocol.

In order to ensure reliable control message transfer, BGMP runs over TCP but uses a different TCP port than BGP's. BGMP routers have TCP peering sessions with each other for the exchange
Chapter 3. Multicast Routing Protocols

of BGMP control messages (e.g., "Join" and "Prunes"). The BGMP peers for a certain group are determined via BGP. It is assumed also that BGP's route selection algorithm ensures that one border router, among the border routers of the domain, is chosen as the best exit router for each group route. This router has an external peer as its next hop toward the root domain of the group and the other border routers have the best exit router for each group route.

Data packets are forwarded in BGMP on a combination of BGMP and MIGP rules. Routers forward data packets to a set of targets according to a matching source-specific entry \((S, G)\) if it exists. If not, a matching shared tree state entry for the group is checked. If neither is found, then the packet is sent natively to the next hop peer for \(G\) that is the best exit border router for the root domain according to BGP rules (this is the case for a non-member sender). If a matching entry was found the packet is forwarded to all other targets in the target list. If a target is a MIGP component, then forwarding is subject to the rules of the MIGP protocol.

Satisfying policy constraints for an autonomous system's multicast traffic and considering heterogeneous routing domains can often translate into an increase of group state maintenance and delivery quality. BGMP goes around this problem by aligning multicast domains with autonomous systems and thus obtaining efficient policy support following the routes defined by BGP. Still, policy control is restricted to the policy constraint support of BGP's underlying hop-by-hop routing paradigm and path vector concept. This implies, for example, that network-specific policies cannot be supported. Furthermore, accepting traffic from "come-from" interfaces might not be discriminatory enough as a policy mechanism. This is because traffic barriers imposed by autonomous systems may be bypassed if a source is covered by a prefix that is homed to more than one domain.

Due to asymmetrical routing environments, such as some satellite networks, don't support bi-directional forwarding, BGMP can not be deployed in these environments. Moreover, BGMP can only support source-specific delivery criteria in limited cases, for the sake of reducing the complexity of the protocol. BGMP has been designed with the aim of being able to be used in heterogeneous multicast routing domains and to be independent of the MIGP deployed intra-domain. Thus, for a globally available multicast routing solution, the use of BGMP implies solving interoperability problems specific to whichever MIGP is in use. This has not proved to be an easy task and encapsulations cannot be avoided. This is the case when the MIGP protocol is suitable for regional deployment but not for supporting multicast transit traffic.

Considering the above, it can be argued whether inter-domain multicast routing would not be better served with a unique routing protocol used intra-domain and inter-domain or an adaptation of an existing protocol that could then be applied to both intra-domain and inter-domain.

**Root Addressed Multicast Architecture**
Chapter 3. Multicast Routing Protocols

In response to the perceived complexity of MBGP/PIM-SM/MSDP and BGMP, and to the need to address additional multicast-related issues like security, billing, and management, some members of the multicast community are looking to make fundamental changes to the multicast model. One class of proposals that has received much attention recently is called the Root Addressed Multicast Architecture (RAMA) [33]. There used to be an Internet draft in IETF discussing this proposal. The premise for RAMA-style protocols is that most multicast applications are single-source or have an easily identifiable primary source. By making this source the root of the tree, the complexity of core placement in other multicast routing protocols can be eliminated. This trade-off raises a number of important issues that are described at the end of this section. There are two primary RAMA-style protocols being discussed: Express Multicast and Simple Multicast. The key aspects of these two protocols are:

**Express Multicast:** Express multicast [35] is designed specifically as a single-source protocol. The root of the tree is placed at the source, and group members send join messages along the reverse path to the source. Express also provides mechanisms to efficiently collect information about subscribers. The protocol is specifically designed for subscriber-based systems that use logical channels. Representative applications include TV broadcasts, file distribution, and any single-source multimedia application. The key advantages of Express are that routing complexity can be reduced and that closed groups can be offered.

**Simple Multicast:** Simple multicast [34] and Express multicast are similar, but Simple Multicast has the added flexibility of allowing multiple sources per group. A particular source must be chosen as the primary, and the tree is rooted at this node's first-hop router. Receivers send join messages to the source, and a bi-directional tree is constructed. Additional sources send packets to the primary source. Because the tree is bi-directional, as soon as packets reach a router in the tree they are forwarded both downstream to receivers and upstream to the core. The advantages and disadvantages of this proposal are being heavily debated, but the proposal's authors believe that it eliminates the address allocation problem and the need to place and locate RPs. Address allocation is done by using the core address and the multicast group address together to uniquely identify a group. By routing on this pair of addresses, each root/core/source can allocate, without collision, up to $2^{32}$ addresses.

The Express and Simple multicast proposals have received significant attention in both the research community and the IETF. There is another question in addition to that of the merits of these new protocols. If these protocols are standardized, will they be expected to replace all existing protocols, or will they work in parallel with the existing multicast infrastructure? If the RAMA-style protocols are expected to work in cooperation with existing protocols, there will be
yet another set of protocols to deploy, evaluate, and interoperate with. This does not make the provision of Internet-wide multicast easier. If RAMA-style protocols are expected to replace the current set of protocols, the question becomes whether they have enough flexibility to support all types of multicast applications. The bottom line is that these new protocols are still proposals, and it is uncertain what their future will be.

3.2.6.3 Source Specific Multicast (SSM)

Till above section, all of the protocols discussed are for the IP multicast service model named Any Source Multicast (ASM) except the idea of Root Addressed Multicast Architecture. ASM was defined in [28]. In this service model, an IP datagram is transmitted to a "host group", a set of zero or more end-hosts (or routers) identified by a single IP destination address (224.0.0.0 through 239.255.255.255 for IPv4). End-hosts may join and leave the group any time, and there is no restriction on their location or number. Moreover, this model supports multicast groups with arbitrarily many senders - any end-host (or router) may transmit to a host group, even if it is not a member of that group. The RAMA proposed the idea of using only one source for each multicast group to solve the inter-domain multicast issues. Although two algorithms proposed for RAMA only stood as Internet drafts in IETF, there is a very similar multicast service model named Source Specific Multicast (SSM) that has been standardized as an informational RFC [36]. [36] and [38] discussed how the SSM handles the inter-domain multicast issues met in ASM and provided information regarding changes needed to routing protocols and applications to deploy SSM. The use of SSM in IPv6 has been introduced in [39] in terms of API implementation in Linux.

Both IPv4 and IPv6 have allocated address ranges for SSM. IPv4 designated 232/8 (232.0.0.0 to 232.255.255.255) as source-specific multicast (SSM) destination addresses. [40] gives operational recommendations to ensure source-specific behaviour within the 232/8 range by preventing local sources from sending to shared tree, preventing receivers from joining the shared tree, preventing RP's as candidates for 232/8, and preventing remote sources from being learned/joined via MSDP. The addresses with prefix FF3x::/32 in IPv6 networks are reserved for SSM [37].

With few modifications, IGMPv3 and MLDv2 can also used for SSM to allow a host to inform its neighbouring routers of its desire to receive IPv4 and IPv6 multicast transmissions, respectively [41]. The modifications have to be made due to two reasons. One is to avoid the confusion of the SSM-enabled hosts caused by non-SSM-enabled hosts and another one is to avoid using source filter mode in SSM.

In a SSM multicast group, there will be only one source that sends data to all receivers in that group. All packets carry address information \((S, G)\), where \(S\) is the source address and \(G\) is the group address. The network service identified by \((S, G)\) is referred to as a "channel." In contrast to the ASM model, SSM provides network layer support for one-to-many delivery service only.
SSM provides many benefits and solves issues that ASM cannot easily overcome.

**Address Allocation:** SSM identifies each multicast group by using channel \((S, G)\), while ASM uses only the destination address \(G\) as group identifier. The latter one will raise address collisions when allocating group address globally due to the possibility that more than one group might have the same identifier \(G\). SSM can avoid it by using the channel identifier \((S, G)\). For instance, two SSM multicast groups, \((S_1, G)\) and \((S_2, G)\), can use the same group address \(G\) but still can be distinct by the source \(S_2\) and \(S_2\).

**Access Control:** In ASM, any host can send data to the group. After hosts join a SSM group associated with a source \(S\), it will only receive data from the channel \((S, G)\). No other sources will have the power to send data to the hosts in this group. Therefore, it's much harder to "spam" an SSM channel than an ASM multicast group.

**Simplicity Routing:** The "shared tree" will be abandoned by SSM and, hence, routing protocols and algorithms can be much simplified from ASM. For instance, the Rendezvous Point (RP) of the PIM-Sparse Mode (PIM-SM) protocol does not need to be supported by SSM. The MSDP is no longer necessary because SSM uses higher layer functions to announce multicast source rather than using any network functions to discover sources located in other domains. There are many means to do so that includes via web pages, session announcement applications, etc. Thus the complexity of the multicast routing infrastructure for SSM is low, making it viable for immediate deployment.

To enable hosts to subscribe and unsubscribe to a SSM channel, Internet Group Management Protocol version 3 (IGMPv3) and Multicast Listener Discovery Version 2 (MLDv2) are used for IPv4 and IPv6 respectively [24][42].

### 3.3 Relevance to Satellite Networks

If a satellite network is involved in multicast services, it should have the ability to support multicast onboard. That means the satellite space segment should have the ability to duplicate packets and, with the management functionality of its ground segment, it should forward these packets to different spot beams where the multicast group receivers locate. Therefore, the satellite should have onboard processing functions, such as switching, and have multiple spot beams. The ground segment, i.e. the satellite Network Control Centre, should have the ability to configure the onboard processing functions according to the dynamic of the multicast group members.

If the multicast service is provided over an integrated network consisting of both satellite networks and terrestrial networks, the inter-domain multicast will be needed. With the
consideration that RAMA architecture is far from standardized, the MBGP/MSDP/PIM-SM solution will be the best choice for a near term inter-domain multicast solution to design a multiparty multimedia communication system over satellite. The SSM architecture could be the long term solution with the development of IPv6 protocol. The multicast routing architecture design for such an integration network with both satellite and terrestrial networks will also be affected by its star topology. The MBGP/MSDP should be adapted in this topology with the satellite Network Control Centre (NCC) working as the heart of the architecture.

3.4 Summary

The concept of multicast technology was introduced in the beginning of this chapter. It enables the many-to-many communications to be more efficient than broadcast. All Users together with the same interest using multicast to communicate with each other are called a multicast group and each of the users is called a multicast group member. Each multicast group is allocated a class D address as group address. IGMP was developed to manage the membership for each member.

Multicast routers have to employ some routing protocols to establish and maintain the connections between multicast group members. These connections construct a delivery tree named multicast routing tree. Those routing protocols can be categorized into two classes. One is intra-domain protocol and the other is inter-domain protocol.

Intra-domain multicast routing protocols is also called Interior Gateway Protocol (MIGP) that is used to enable multicast routers to implement multicast communication within an autonomous system (AS). Some routing protocols in this class were presented, including Distance-Vector Multicast Routing Protocol (DVMRP), Multicast Extensions to Open Shortest Path First (MOSPF), Protocol-Independent Multicast Dense Mode (PIM-DM) and Sparse Mode (PIM-SM), and Core Based Trees (CBT). All of these protocols were divided into three subgroups. DVMRP and PIM-DM are the representatives for the dense mode multicast routing protocol which assumes most users in the network are interested in multicast traffic and floods the traffic to all routers. In another hand, the sparse mode multicast routing protocols assume group members locate in different networks and packets flooding will cause too much resource wasted. It roots the multicast routing tree at a rendezvous point (RP) and all members will join the group explicitly by communicate to the RP. The representatives are PIM-SM and CBT. The third subgroup multicast routing protocol is link-state protocols such as Multicast Open Shortest Path First (MOSPF), which tries to find out all group members by flooding link-state information to the network.

Inter-domain multicast routing protocols are employed by boarder routers to enable multicast communication between different AS. Two solutions were presented in this chapter. One is so called near-term solution that uses PIM-SM inside each domain to build up intra-domain multicast
tree, Multicast Source Discovery Protocol (MSDP) to announce the active source to different domains and Multiprotocol BGP (MBGP) to establish the routes for multicast data between each domain. This solution is relatively straightforward once a person understands all the abbreviations and understands the motivating factors that drove the design of the protocols. It is a functional solution largely built on existing protocols and already being deployed with a fair amount of success. The drawback is the scalability problem caused by the MSDP overhead flooding when the multicast group size goes too big. The long-term solutions can be broken down into two groups. The Border Gateway Multicast Protocol (BGMP) / Multicast Address Set-Claim (MASC) solution was proposed based on the standard IP multicast philosophy, namely the Any Source Multicast (ASM). This solution is trying to establish bi-direction shared tree of domains associated with a single root or core domain. It will need the cooperation with policies regarding administration of each domain. How to choose a root domain is the key issue in this solution. In response to the perceived complexity of MBGP/PIM-SM/MSDP and BGMP, and the need to address additional multicast-related issues like security, billing, and management, some members of the multicast community are looking to make fundamental changes to the multicast model. The Root Addressed Multicast Architecture (RAMA) is one proposal that received much attention.

The premise for RAMA-style protocols is that most multicast applications are single-source or have an easily identifiable primary source. By making this source the root of the tree, the complexity of core placement in other multicast routing protocols can be eliminated. Express Multicast and Simple Multicast used to be two primary RAMA-style protocols. A very similar multicast service model namely Source Specific Multicast (SSM) has been standardized as an informational RFC [36] that discussed how the SSM handles the inter-domain multicast issues met in ASM and provided information regarding changes needed to routing protocols and applications to deploy SSM.

The relevance of multicast routing to satellite network was also discussed. It points out the requirements for both satellite space segment and ground segment to support multicast. The MBGP/MSDP/PIM-SM solution is considered the best choice for the near term with necessary modifications to be adapted to the star topology of an integration network including both satellite and terrestrial networks.
4 Multimedia Conference Model over Satellite

With the knowledge of the VoIP protocols and multicast routing protocols, one can try to build up a rough idea of a conference system. The physical network support is another key factor with the fast-developing real-time multiparty multimedia communications services. Due to their ability to deliver high data rates and their global coverage, GEO satellites become a good choice to provide large-scale IP conference services. With the consideration of satellite propagation delay, the On Board Processing (OBP) GEO satellite is a suitable choice because of their wide coverage with one satellite hop transmission. LEO and MEO satellite can also be used to provide wide coverage with less delay. However, the complexity of LEO and MEO satellite networks is much higher than GEO satellite in terms of routing between satellites and the mobility of satellites. In this chapter, some conference models will be studied in terms of the suitability to satellite environment and one will be proposed for the conference system over a GEO satellite.

The main differences between satellite and terrestrial networks are the satellite propagation delay and expensive bandwidth of the satellite network. The long single hop delay in the GEO satellite network could be up to 280ms and that makes it unacceptable to have media transmitted twice over satellite link. For instance, some conference models require User A sending data to User B over one satellite hop and User B has to relay the data to User C over another satellite hop. Therefore, the same data will have to be transmitted over the satellite hop twice and two satellite hop delay will be added to the communication between User A and User C. This should be avoided in a good conference model designed for a satellite conferencing service system. However, the signalling can be carried over satellite several times in order to establish the satellite connections because it is not as delay sensitive as media. So in this chapter, several existing conference models will be studied and a new one will be proposed to avoid the multi-hop satellite link for media transmission and save satellite bandwidth using MCUs. There will be subsections to describe the availability of all studied conference models based on the avoidance of multi-satellite-hop media transmission. Models with the need of multi-satellite-hop media transmission will only be studied very briefly since they are not suitable for a conference service over a GEO satellite.

Conference models are established heavily based on the used signalling protocol. A review of the existing conference models will be carried out first. These models are based on SIP considering
4 Multimedia Conference Model over Satellite

With the knowledge of the VoIP protocols and multicast routing protocols, one can try to build up a rough idea of a conference system. The physical network support is another key factor with the fast-developing real-time multiparty multimedia communications services. Due to their ability to deliver high data rates and their global coverage, GEO satellites become a good choice to provide large-scale IP conference services. With the consideration of satellite propagation delay, the On Board Processing (OBP) GEO satellite is a suitable choice because of their wide coverage with one satellite hop transmission. LEO and MEO satellite can also be used to provide wide coverage with less delay. However, the complexity of LEO and MEO satellite networks is much higher than GEO satellite in terms of routing between satellites and the mobility of satellites. In this chapter, some conference models will be studied in terms of the suitability to satellite environment and one will be proposed for the conference system over a GEO satellite.

The main differences between satellite and terrestrial networks are the satellite propagation delay and expensive bandwidth of the satellite network. The long single hop delay in the GEO satellite network could be up to 280ms and that makes it unacceptable to have media transmitted twice over satellite link. For instance, some conference models require User A sending data to User B over one satellite hop and User B has to relay the data to User C over another satellite hop. Therefore, the same data will have to be transmitted over the satellite hop twice and two satellite hop delay will be added to the communication between User A and User C. This should be avoided in a good conference model designed for a satellite conferencing service system. However, the signalling can be carried over satellite several times in order to establish the satellite connections because it is not as delay sensitive as media. So in this chapter, several existing conference models will be studied and a new one will be proposed to avoid the multi-hop satellite link for media transmission and save satellite bandwidth using MCUs. There will be subsections to describe the availability of all studied conference models based on the avoidance of multi-satellite-hop media transmission. Models with the need of multi-satellite-hop media transmission will only be studied very briefly since they are not suitable for a conference service over a GEO satellite.

Conference models are established heavily based on the used signalling protocol. A review of the existing conference models will be carried out first. These models are based on SIP considering
its simple structure and other advantages over the H.323 architecture discussed in Chapter 2.

The Session Initiation Protocol (SIP) can support multi-party conferencing in many different ways. In this section, the various multi-party conferencing models defined in [21] are reviewed and one new model was proposed. For each model, it is discussed how they are used and then analysed their relative benefits and drawbacks for the satellite environment, in such a way that some of them will be discarded since the satellite network only supports the models that do not require two satellite hops for delivering the media data (audio, video, text, etc.).

Hence, at this point, it will be described, in a consistent and complete fashion, the various multi-party conferencing models that an IP conference system should support. For each model, some key issues will be discussed including how the model works and if it is suitable for the conference system over satellite.

4.1 End System Mixing Model

The first model described in [21] is called "end system mixing". In this model, user A calls user B, and they have a conversation. Later, user A decides to invite user C to join the conference. To do this, user A calls user C, using a completely separate SIP call. This call uses a different Call-ID, different tags, etc. There is no call set up directly between users B and C. User A receives media streams from both B and C, and mixes them. Then, user A sends a stream containing A's and C's streams to B as well as a stream containing A's and B's streams to C.

This model is not suitable for satellite networks since more than one satellite-hop is required, so it is not more studied here.

4.2 Large-Scale Multicast Conferences Model

In a large-scale multicast conference, one or more multicast addresses are allocated to the conference (more than one may be needed if layered encodings [72] are in use). Each participant joins the multicast group, and sends their media to the group. Signalling is not sent to the multicast group. The sole purpose of the signalling is to inform participants of which multicast group to join, so they can learn them from: another participant through SIP (establishing a point-to-point SIP relationship with him), or another way (no signalling will be required at all), such as a web page, mail, etc.

Hence, if there are N participants, each participant sends a single media stream to the group, and receives up to N-1 streams at any time. Note that the number of streams that a user will receive depends on who is actually sending at any given time. If the stream is audio, and silence suppression is utilized, the number of streams a user will receive at any given time is equal to the
number of users talking at any given time. Even for very large conferences, this is usually just a small number of users.

Large-scale multicast conferences are usually pre-arranged, with specific start and stop times. Protocols such as the Session Announcement Protocol (SAP) [71] are used to announce these conferences. However, multicast conferences do not need to be pre-arranged, so long as a mechanism exists to dynamically obtain a multicast address. SAP itself was originally used for this purpose.

**4.2.1 Suitability of Large-Scale Multicast Conferences Model for IP Conference over Satellite**

It is supposed in this model that all the terminals in the conference are fully multicast enabled and they perform audio/video mixing or switching as needed (it is also possible to include non-multicast terminals by means of conference bridges). This model is well suited for high capacity terminals and minimizes the total bandwidth in the network. Moreover, within the mixed terrestrial plus satellite network topology, this is the preferred model because it takes advantage of the intrinsic multicast nature of satellite transmission. Audio mixing, a high CPU consuming process, must be done in each terminal. For video, switching is desired (select a video source to present to the user from the video pool).

**4.2.2 Inviting Users to Join in Large-Scale Multicast Conferences Model**

Inviting users to join is simple. Any user may invite any other user to join. The SIP INVITE request indicates multicast addresses. From Section B.3 of [10]: "For multicast, receive and send multicast addresses are the same and all parties use the same port numbers to receive media data. If the session description provided by the caller is acceptable to the callee, the callee can choose not to include a session description or may echo the description in the response." The called party then joins the multicast groups indicated in the INVITE request, using multicast protocols such as IGMP. Note that it is not even necessary for users to send each other BYE messages when the conference is over, especially for large-scale, pre-arranged conferences that have explicit end times indicated.

A participant can simply leave the conference at any time by leaving the multicast groups. No SIP signalling is needed to accomplish this.

Users can join a conference of this type without being invited. All they need is the multicast addresses, ports, and codecs being used. Conference descriptions can even be obtained from web pages, for example. Once the addresses are obtained, the user simply joins the appropriate multicast groups. Note that absolutely no SIP signalling is required in this case.
4.2.3 Scalability of Large-Scale Multicast Conferences Model

The scalability of conferences of this type can be excellent, especially for audio conferences. However, it is scalable under the assumption that multicast itself can scale to very large groups. Indeed, in local networks, protocols like DVMRP and PIM-DM have tremendous scalability for conferences with very large numbers of members (the so-called dense modes that are precisely the protocols desirable to be used in a satellite environment). Given the existence of scalable multicast, the primary bottleneck to scalability of this conference type is the periodicity of RTCP reporting when many users join a multicast RTP session at nearly the same time. Work has been done on improving this problematic case so that conferences with many members are possible.

The main problem is congestion: the access bandwidth for users is small compared to network bandwidth. Many multicast RTP sessions will exhibit rapid increases in groups’ membership at certain period in time. The result of these step joins is inaccurate in the group size estimation obtained by listening to the group: each newly joined member believes that they are the only one, at least initially. They send RTCP packets at their fair share of the RTCP bandwidth, which each believes is all of it. Combined with slow access links, the result is a flood of RTCP reports, causing access link congestion and loss. To solve this problem, an adaptive timer algorithm called reconsideration has been developed, which reduces the congestion problem by several orders of magnitude. This algorithm implements a simple back-off mechanism which causes users to hold back RTCP packet transmission if the group sizes are increasing [20].

Scaling is a bit harder for videoconferences using this model. Unlike voice, where silence suppression allows for no data to be sent during periods of inactivity, the same is not the case for video although adaptive coding techniques can be used for data rate control [73]. This makes it hard to scale without flooding users with lots of video packets.

4.2.4 Location of Service Logic in Large-Scale Multicast Conferences Model

This conferencing model does not require any SIP extensions. It does require that SIP UAs are prepared to receive SIP invitations with multicast addresses. These UAs need to be prepared to mirror these addresses in the response. They should also be prepared to never receive a BYE for the conference.

4.3 Dial-In Conference Model

A dial-in conference server acts as a normal SIP UA. Users call it, and the server maintains point to point SIP relationships with each user that calls in. The server takes the media from the users who dial into the same conference, mixes them, and sends out the appropriate mixed stream to
each participant separately.

Each UA (for example, A, B and C) has a point to point SIP and RTP relationship with the conference server. Each call has a different Call-ID. Each user sends their own media to the server:

- The media delivered to user A by the server is the media mixed from users B and C.
- The media delivered to user B by the server is the media mixed from users A and C.
- The media delivered to user C by the server is the media mixed from users A and B.

The conference is identified by the request URI of the calls from each participant. This provides numerous advantages from a services and routing point of view. For example, one conference on the server might be known as sip:conference34@servers.com. All users who call sip:conference34@servers.com are mixed together.

Dial-In conference servers are usually associated with pre-arranged conferences. However, the same model applies to ad-hoc conferences. An ad-hoc conference server creates the conference state when the first user joins, and destroys it when the last one leaves. The SIP and RTP interfaces are identical to the pre-arranged case.

Since conferencing servers are nothing more than SIP UASs, they can use any of the procedures SIP allows a UAS to use. This includes authentication. So, for example, a specific conference may have a password associated with it. Users who join are challenged (with a 401 - Unauthorized) using digest authentication. The realm, in this case, would identify the conference. The INVITE that comes back would have an Authorization header that includes the response to the challenge - the name of the user trying to join the conference, and the conference password.

Conferences can also limit the number of participants. When a new user tries to join, but the conference is full, the conference server can just reject the request with a 500 - Conference Full response.

4.3.1 Suitability of Dial-In Conference Model for IP Conference over Satellite

This model implies two satellite-hops for the media data so, by its own, it is not usable in the satellite network. Nevertheless, it is included here since it will be found as a part of the model proposed named Multiple Media Servers model, which proposes to include a MCU (Multipoint Control Unit) in each corporate or ISP network. Terminals will send multimedia streams to its MCU, which will collect and manipulate them, generating multicast flows received by all terminals. Hence, this model will consist of the addition of two different scenarios:
There will be a scenario which models the relationships between local-users and their own MCU in each corporate or ISP network environment. This model will be the "Dial-In Conference Servers" here explained.

The other mentioned scenario will model the relationships established among MCUs. Here, several models can be implanted.

4.3.2 Inviting Users to Join in Dial-In Conference Model

Inviting users to join is done using the SIP REFER message. If user A wishes to ask user B to join:

A would send B a REFER that looks like:

REFER sip:B@example.com SIP/2.0
From: sip:A@example.com
To: sip:B@example.com
Refer-To: sip:conference34@servers.com

This would cause B to send an INVITE message to the conference server:

INVITE sip:conference34@servers.com
From: sip:B@example.com
To: sip:conference34@servers.com
Referred-By: sip:A@example.com

Since the request URI identifies the conference, this will cause B to be added to conference 34.

An additional mechanism for inviting a user to join is to send REFER from A to the conference server, with a Refer-To containing the address of B. This REFER would look like:

REFER sip:conference34@servers.com SIP/2.0
From: sip:A@example.com
To: sip:B@example.com
Refer-To: sip:B@example.com

This approach has the advantage that it doesn't require REFER support from B, only from the conference server.

A problem with the mechanisms for adding a user is that they assume that the UA for user A (the one who adds another user to the conference) knows that it is indeed talking to a conference server. If the mechanisms in this section were applied to a UA which was not a conference server,
the result would be the creation of additional call legs, but not a conference. This means that we require some mechanism for identifying that a URL is a conference URL.

#### 4.3.3 Users Joining in Dial-In Conference Model

It is easy for users to join the conference. The participant that wishes to join simply sends an INVITE to the conference server, with the conference ID in the request URL. The conference ID (which is a SIP URL), can be learned by any number of means, including having it on a web page, receiving it in an e-mail, etc.

For example, if B wishes to join sip:conference34@servers.com, B would send the following request:

```
INVITE sip:conference34@servers.com
From: sip:B@example.com
To: sip:conference34@servers.com
```

#### 4.3.4 Scalability of Dial-In Conference Model

The scalability of this model is limited by the bandwidth and processing power of the conference server. If there are N participants in a conference, M of which are sending media streams, the server will need to manage N signalling relationships, perform M RTP stream decodes, and N RTP stream encodes (assuming M > 0). The encoding is the primary processing bottleneck, and the sending of the N media streams is the primary bandwidth bottleneck. However, conference servers can be built using heavy-duty hardware, and have high bandwidth access.

Furthermore, since the request URI is being used to name the conferences, standard SIP techniques can be used for distributing conferences across servers.

#### 4.3.5 Location of Service Logic in Dial-In Conference Model

The SIP UA of the conference participants does not require any special processing. The RTP implementation in those clients, however, should support RTCP and be prepared to receive contributing sources.

All of the new logic for providing this service resides in the conferencing server. No SIP extensions are needed, simply logic that resides above the SIP stack to manage the conferencing service.
4.4 Ad-hoc Centralized Conferences

In an ad-hoc centralized conference, two users A and B start with a normal SIP call. At some point later, they decide to add a third party. Instead of using end system mixing, they would prefer to use a conference server.

This model is not suitable in the satellite environment like the dial-in model where two satellite hops will be needed for media data transmission. So it is not studied further.

4.5 Dial-Out Conferences

Instead of the users joining the conference by sending an INVITE to the server, the server chooses the users who are to be members of the conference, and then sends them the INVITE. Typically dial out conferences are pre-arranged, with specific start times and an initial group membership list. However, there are other means for the dial-out server to determine the list of participants, including user presence. The model in no way limits the means by which the server determines the set of users. Once the users accept or reject the call from the dial out server, the behaviour of this system is identical to the dial-in server case. Thus, a dial-out conference server will generally need to support dial-in access for the same conference, if it wishes to allow joining after the conference begins.

Dial-out conferences are a simple variation on dial-in conferences so, for the same reason, it is not valid for the satellite network.

4.6 Centralized Signalling with Distributed Media

In this conferencing model, there is a centralized controller, as in the dial-in and dial-out cases. However, the centralized server handles signalling only. The media is still sent directly between participants, using either multicast or multi-unicast. Multi-unicast is when a user sends multiple packets (one for each recipient, addressed to that recipient). The case of multicast will be studied better in the next model. Interestingly, this conference model is possible with SIP and it is referred to as "Decentralized Multipoint Conference" in H.323.

It works through third party call control. The conference server uses re-INVITEs to each participant when a new one joins. The re-INVITEs add a media stream that gets sent to the new participant (and similarly in the reverse direction).

One can assume for the moment that a conference already exists with three participants. In this state, each participant is sending media directly to each other. This is because the SDP [74] that the conference server has given to each participant contains three media lines, each of type audio,
with connection addresses and ports corresponding to each of the three users.

The call flow from here is shown in Figure 4-1. There, the word after the INV or SIP response code refers to the connection address in the SDP message. +X means the addition of a stream with X as the recipient address.

When a new participant joins the conference, it does so by sending an INVITE to the server, with the conference ID in the request URI. The SDP in the INVITE contains a single media stream, with an IP address and port where it would like to receive media. The 200 response from the conference server contains a single media line with an IP address of 0.0.0.0 and a port, indicating hold.

The next step is for the server to obtain two more addresses where the new participant will be receiving media (it already has one from the original INVITE). To do this, it sends a re-INVITE to the new participant. This re-INVITE contains two additional media streams (for three total), all three of which are on hold. The 200 response to the re-INVITE contains two additional IP addresses and ports where the user is willing to receive media. Now the server needs to inform the other parties that they should begin sending media to the new user. It first sends a re-INVITE to user C. This re-INVITE adds an additional media stream to the two already that C has been sending. This new media stream uses one of the three connection addresses and ports returned by D in message. Call this address/port D1. The other two are D2 and D3. The 200 OK response from user C contains the address and port where C is willing to receive a new, third media stream. Call this port C3. The server holds on to this port, as it will use it later on, sending it to D, so that D sends media there. At this point, however, C can begin sending media to D.
Figure 4-1: Centralized signalling with decentralized media

This re-INVITE process happens for B and for A as well. In the re-INVITE to B, the server adds an additional media line (above the two already in use by C) using address/port D2. The response contains a new address/port to send media to B. Call this port B3. In the re-INVITE to A, the server adds an additional media line using address/port D3. The response contains a new address/port to send media to A. Call this port A3.

Finally, the server sends a re-INVITE to the new party. This re-INVITE takes all three streams off hold, and updates their connection addresses and ports with C3, B3, and A3, respectively. The 200 OK response returns the same ports and addresses returned in message (these addresses/ports must not change). Now, D can send media to A, B and C.

The result of these manipulations is, in this case of using multi-unicast, a full mesh of unicast RTP streams between all participants. Unlike the model of end system mixing, the stream sent by any participant to all of the others is identical. Each participant needs to mix, but it mixes the media it receives, and plays that out the speakers. This is normal behaviour for multiple streams of the same type. Note that the SIP relationship is still point-to-point. There are four calls at the end of the process, one from each participant to the server, each with a different Call-ID. Note that hybrids are easily possible. Certain users can instead be mixed (sending audio to the conference server), while others are set to send audio to each other.

4.6.1 Suitability of Centralized Signalling with Distributed Media Conference Model for IP Conference over Satellite

Although the signalling process will require more than one satellite-hop, that is not a restricted issue in our environment since the “suitability parameter” is the number of hops for the delivering of the media data, and this is maintained to one. Nevertheless, media data are sent several times through the satellite network because this model does not use the multicast capability of the satellite. So, this model is not required for IP conferencing over satellite. The next one follows the same scheme, but it does use the multicast support.

4.6.2 Inviting Users to Join in Centralized Signalling with Distributed Media Conference Model

Inviting users to join works identically to the dial-in conference bridge scenario. A user joins in the same way described in Section “Dial-In Conferencing Servers”.

51
4.6.3 Scalability of Centralized Signalling with Distributed Media Conference Model

The scalability of this conferencing model depends on many factors. From a media perspective, the conference server never even touches a single media stream. However, for N participants, each participant needs to be able to receive, decode, and mix N-1 media streams. For users accessing the server through dial-in modems, this will severely limit the sizes of these conferences. However, the processing burden is much less than that of the end system mixing model. This is because each end user needs to decode N-1 streams, but only encode 1. Decoding is less complex than encoding, so supporting many decoders is not necessarily a problem. This is especially the case when silence suppression is in use. In that case, streams are only sent by talking users. This means any given user only needs to decode (and receive) as many streams at a time as there are users talking. This can vastly improve scalability of the conference.

There is a signalling burden on the server, however. If there are N users in the conference, addition of a new user (the N+1th) requires N+3 INVITE transactions, each of which has three messages. Similarly, departure of a user requires N BYE transactions, each of which has 2 messages. For large N, and highly dynamic conferences, this can represent a potential burden.

4.6.4 Location of Service Logic in Centralized Signalling with Distributed Media Conference Model

Nearly all of the logic for implementing this conferencing service lives in the server itself.

The only requirement from the end users is that they support multiple, parallel media streams of the same type, and that they be prepared to mix those streams together. They must also support the third party control primitives, which don't require anything beyond baseline SIP, but are not likely supported unless explicit actions are taken to do so.

It is this combination - no need for media processing in the server, combined with no need for specialized SIP processing in the end systems, which makes this model attractive.

4.7 Centralized Signalling with Distributed Media – Multicast

If multicast is wished to be used for sending media data in the model described in section 4.6, one or more multicast addresses are allocated to the conference (more than one may be needed if layered encodings are in use). Each participant joins those multicast groups, and sends their media to those groups. Note that users can join a conference of this type without being invited. All they need is the multicast addresses, ports, and codecs being used. These can be obtained through any
number of means, including SAP. SDP conference descriptions can even be obtained from web pages, for example. Once the addresses are obtained, the user simply joins the appropriate multicast groups. Note that absolutely no SIP signalling is required in this case. If the user has not learnt the multicast addresses by another way, SIP signalling will not be sent to the multicast groups but to the conference server, which will inform participants of which multicast groups to join.

Although the signalling process will require more than one satellite-hop, that is not a restricted issue in our environment since the "suitability parameter" is the number of hops for the delivery of the media data, and this is maintained to one. This model is suitable for IP conferencing over satellite because it does use the multicast network support.

4.8 Multiple Media Server (Multiple-MCU) Model

In the GEO satellite system that is considered, a new scenario is defined. In this model, one or more MCUs (Multipoint Control Units) exist in the network. Terminals send multimedia streams to the MCUs, which collects the streams, manipulates them and generates multicast flows received by all terminals. This model minimizes the bandwidth in comparison to the unicast conference and simplifies the terminal requirements. By taking use of the functions of powerful MCUs, many features can be easily added and managed in a conferencing service, such as support to various media codecs and video resolutions, dial in / dial out capabilities, automatic change of video image layout and resolution dependent on the number of participants, media stream manipulation, and password management.

The mixed satellite-terrestrial network and the relatively high satellite delay imply that it is not desirable to send unicast audio/video from one terminal to a remote MCU through satellite networks and then receive the composite signal again through the satellite link. For this reason, several MCUs are needed, at least one in each corporate/business or ISP network, so found in the "Dial-In Conference Servers" scenario in each local network, where the conference server is now called MCU. In this way, a MCU acts as a normal SIP UA: users call it, and it maintains point-to-point SIP relationships with each local-user that calls in. The MCU takes the media from the local-users who dial into the same conference, mixes them, and sends out the appropriate mixed stream to the other participant-MCUs. For example, consider the situation shown in Figure 4-2:
As it has been explained, each MCU-sub-network is modelled by the “Dial-In Conference Servers” scenario reviewed before: User Agent A and B have point to point SIP and RTP relationships with the MCU1. In the same way, the User Agents C and D have point to point SIP and RTP relationships with the MCU2. Nevertheless, the relationships established among MCUs can respond to two different models already described in “Large-Scale Multicast Conference” or “Centralized Signalling, Distributed Media”. These two models will determine the remaining characteristics of the “Multiple-MCU” scenario.

4.8.1 Suitability of Multiple-MCU Model for IP Conference over Satellite

This model is very suitable for an IP conference over satellite with only one satellite hop media data transmission. It also optimizes the satellite bandwidth utilization by aggregating traffic from several sources together in the terrestrial networks. It allows both unicast end users and multicast end users to communicate in the same conference service.

4.8.2 Inviting Users to Join in Multiple-MCU Model

Inviting users to join can be done using the SIP REFER message, in the same way described in section “Dial-In Conferencing Servers”. If user A wishes to ask user C to join:

A would send C a REFER that looks like:

REFER sip:C@example.com SIP/2.0
From: sip:A@example.com
To: sip:C@example.com
Refer-To: sip:conferenceXX@servers.com

This would cause C to send an INVITE message to the MCU2:

INVITE sip:conferenceXX@servers.com
From: sip:C@example.com
Chapter 4. Multimedia Conference Model

To: sip:conferenceXX@servers.com
Referred-By: sip:A@example.com

If the MCU2 has a user already in this conference, the signalling process ends here since the request URI identifies an existing conference, so user C is added to conference XX. Nevertheless, if user C is the first one joining this conference in MCU2’s local network, more signalling is needed. In case of using the “Large-Scale Multicast Conference” model among MCUs, MCU2 will establish a point-to-point SIP relationship with MCU1 in order to learn which multicast groups to join. However, if the “Centralized Signalling, Distributed Media” model is employed, the Signalling Server will be the agent to contact.

In this SIP message exchange, it has been supposed that user A knows the IP address of user C and was able to send the REFER message directly to that address. This will not be the case in general. Moreover, this approach requires REFER support, which it is not the general case either. Figure 4-3 shows an example of a more typical SIP call with a SIP proxy server using INVITE messages. The proxy server sits in the middle of a SIP message exchange, receiving messages and forwarding them. It has been considered for simplicity that user C is not the first user at its network joining this conference.

![Figure 4-3: SIP call example with proxy server using INVITE](image-url)
Because user A does not know exactly where user C is currently logged on, the SIP proxy server is used to route the INVITE. First, a DNS lookup of user C’s SIP URL domain name (example.com) is performed, which return the IP address of the proxy server proxy.example.com which handles that domain. The INVITE is then sent to that IP address. The proxy looks up the SIP URL in the Request-URI (sip:C@example.com) in its database and locates user C. This completes the two-step process:

- DNS lookup by user agent to locate the IP address of the proxy; database lookup is performed by the proxy to locate the IP address.
- The INVITE is then forwarded to user C’s IP address with the addition of a second Via header stamped with the address of the proxy.

From the presence of two Via headers, user C knows that the INVITE has been routed through a proxy server, so the 180 Ringing message will be sent by user C to the proxy. The proxy server receives the response, checks that the first Via header has its own address (proxy.example.com), removes that Via header and then forward the response to the address in the next Via header: the user A’ IP address. If user C wants to accept the call, he/she continues like in the example before (without the proxy), sending an INVITE to the MCU2. Once user C receives the 200 OK confirmation from MCU2, he/she can send the corresponding 200 OK response to the INVITE from user A. Again, this response is sent to the proxy server, which forwards it to user A after removing the first Via header. The presence of a Contact header with the SIP URL address of user C in this 200 OK allows user A to send the ACK message directly to user C bypassing the proxy.

Once user C has been added to the conference, terminals send multimedia streams to its MCU, which collects the streams, manipulates them and generates multicast flows received by all terminals in the conference.

The media session is ended when user C sends a BYE message to both MCU2 and user A.

As it has been said, this mechanism has the advantage that it doesn’t require REFER support at all. Nevertheless, it will suppose a more messages exchanges, and consequently, delays increase. Figure 4-4 shows a simpler message exchange, using REFER and proxy server:
Figure 4-4: SIP call example with proxy server using REFER

The use of the Via and Contact headers is exactly the same to the one explained before.

### 4.8.3 Users Joining in Multiple-MCU Model

It is easy for users to join the conference. The participant wishing to join simply sends an INVITE message to its own MCU, with the conference ID in the request URI. The conference ID (which a SIP URL) can be learned by any number of means, including having it on a web page, receiving it in an e-mail, etc.

### 4.8.4 Scalability of Multiple-MCU Model

On the one hand, the scalability of this model is limited by the bandwidth and processing power of the MCU. If there are $N$ local-participants in a conference, $M$ of which are sending media streams, the MCU will need to manage $N$ signalling relationships, perform $M$ RTP stream decodes, and $N$ RTP stream encodes (assuming $M > 0$). The encoding is the primary processing bottleneck, and the sending of the $N$ media streams is the primary bandwidth bottleneck. However, MCUs can be built using heavy-duty hardware, and have high bandwidth access.

On the other hand, if the “Centralized Signaling, Distributed Media” model is employed, the conference server – or Signalling Server - never even touches a single media stream. However, for $N$ participant-MCU’s, each participant needs to be able to receive, decode, and mix $N$ media streams. For MCU accessing the server through dial-in modems, this will severely limit the sizes of these conferences. Moreover, there is a signaling burden on the server. If there are $N$ user-
MCUs in the conference, the addition of a new user (the \((N+1)^{th}\)) requires \(N+3\) INVITE transactions, each of which has three messages. Similarly, the departure of a user-MCU requires \(N\) BYE transactions, each of which has 2 messages. For large \(N\), and highly dynamic conferences, this can represent a potential burden.

4.8.5 Location of Service Logic in Multiple-MCU Model

The SIP UA of the conference participants does not require any special processing. The RTP implementation in those clients, however, should support RTCP and be prepared to receive contributing sources.

All of the new logic for providing this service resides in the MCU. No SIP extensions are needed.

4.9 Summary

Conference models are established largely based on the signalling protocol used. Firstly, a review of the existing SIP based conferencing models had been presented in this chapter. For each of them, the suitability to satellite networks was discussed. The main constraint is that they should not require more than one satellite hop for delivering the media data. After studying existing conference models, a new model was proposed to support the conferencing over satellite more efficiently and flexibly.

For those suitable and proposed models it is also discussed how users are invited to join the conference, how users can join an existing conference without being invited, how well the model scales, which entities need to be aware of the model and how participants learn about each other.

The study results showed that two existing conference models with multicast support can be used in a conference system over satellite. They are the large-scale multicast conferences model and the centralized signalling, distributed multicast media conference model. For large-scale multicast conferences model, all the terminals in the conference are fully multicast enabled and they perform audio/video mixing or switching as needed. It is well suited for high capacity terminals and minimizes the total bandwidth in the network. It takes advantage of the intrinsic multicast nature of satellite transmission. For the centralized signalling, distributed multicast media conference model, although the signalling process will require more than one satellite-hop, it is not an issue here since the "suitability factor" is the number of satellite hops for the delivery of the media data, and this is maintained to one. Therefore, it is also suitable for satellite networks.

Based on the studied conference models, a new model named multiple media servers or the multiple MCU model was proposed. In this model, one or more MCUs (Multipoint Control Units) exist in the network. Terminals send multimedia streams to the MCU, which collects the streams,
Chapter 4. Multimedia Conference Model

manipulates them and generates multicast flows received by all terminals. This model reduces the bandwidth in comparison to the unicast conference and simplifies the terminal requirements. At least one MCU should be located in each corporate/business or ISP network to avoid sending unicast audio/video from one terminal to a remote MCU through satellite networks. The relationships established among MCUs can operate with the two different models already described, the large-scale multicast conference model or the centralized signalling, distributed multicast media model.
Chapter 5. Routing Architecture for Conference over Satellite

5 Hybrid Routing Architecture for Multiple MCU Conference over Satellite

When the conference model is established, one has to address a routing architecture to fit with that model and carry traffic to enable the multimedia multiparty conference services. As discussed in Chapter 3, multicast routing technologies make the multiparty communication feasible and the PIM-SM/MSDP/MBGP solution is a suitable choice for the large-scale satellite conference system. However, one has also to consider the suitability to the proposed multiple MCU model and the compatibility with the existing networks. In this section, a proposed hybrid unicast/multicast routing architecture for the multiple MCU model will be discussed and the detail design of the satellite multicast algorithm will be presented as well. Finally, a multicast supported conference will be setup step by step over this architecture.

After the users join a conference, they will send and receive media data to and from the network. To seamlessly cooperate with the Multiple MCUs conference model, a multicast and unicast hybrid routing architecture was proposed for IP conference over satellite.

The first consideration to use this architecture is to minimize the requirement on the end user equipment, which will not need the high power to mix media stream from all active sources in the conference but receive mixed stream from their local MCUs via a unicast connection. All of the MCUs connect with each other using multicast, which can enable one-to-many communication and save the uplink bandwidth if via satellite. It has an additional effect to solve the problem that the end user may not want to receive what he sent out to his local MCU or different users want to receive from different sources, which means all end users cannot belong to the same multicast group. This proposed solution enables the end user to receive data streams from sources he is interested in using unicast from the MCU with which he is associated.

Another reason for this choice is that multicast is far from mature in the Internet. Multicast infrastructure cannot be easily established due to the huge demands on equipment replacements and reconfiguration. The hybrid routing solution can minimize this kind of work because the multicast domain is only needed among MCUs. Its multicast user capacity allows it to easily integrate with and/or migrate to a full-multicast enabled network.

The unicast routing part is no different with its traditional implementations in the Internet and therefore, there are no technical challenges for it. However, to deploy multicast over the GEO satellite to seamlessly cooperate with the terrestrial multicast network is not as straight forward.
5.1 Multicast over Satellite for IP conferencing

MCUs and multicast end users communicate with each other using multicast in the IP conference over satellite. Multiparty IP conferencing over satellite implies the possibility of widely geographically distributed participants. Therefore, both intra-domain and inter-domain multicast scenarios were studied.

Before going any further on the multicast over satellite design, one has to understand the characteristics of the satellite network that imposes requirements on the design.

5.1.1 General Satellite Architecture for Conference Service

This section will try to give a description about what the difference of the satellite links can bring to the terrestrial network, and what properties this integrated network has. It is trying to produce a plot of the general satellite and terrestrial integrated network from the system point of view, which will unveil the characteristics of this network and give a good understanding of the reasons that stand behind the choice of the multicast routing design.

This section provides a general view of the network without discussion on any particular protocols or particular services that can run over the network. However, as a supplement to IP conferencing over satellite, it places some requirements on the conference model and the multicast deployment due to the effects of the characteristics of the general satellite network.

Considering the integration of the GEO satellite with terrestrial networks, a network consists of two individual parts: the satellite networks and the terrestrial networks. They comprise independent administrative domains called Autonomous Systems (AS), which may belong to different Internet Service Providers (ISP). Figure 5-1 shows an integration topology for instance. In this figure, a satellite network comprising satellite AS linked with three terrestrial ASs via BGs that are Border Gateway routers. The terrestrial networks have connectivity through the Internet.
5.1.1.1 Terrestrial Networks

The terrestrial ISPs who are involved in a satellite conference system need to establish inter-domain connection with the satellite domain as their first hop in the network, such as the ISP\textsubscript{x}, ISP\textsubscript{y}, and ISP\textsubscript{z} shown in Figure 5-1. They are called Federated Domains, which are all connected to the Internet as well. Thus, any other non-federated ISP ASs can join the conference system by passing through any of these federated domains by means of inter-domain routing and agreements between them.

As conferencing services and multimedia streaming are bandwidth consuming in each federated ISP domain, therefore intra-domain multicast should be deployed to save bandwidth as well as provide multicast functionality. For the same reason, multicast should be deployed over the satellite to save bandwidth. To utilize the multiple MCUs model, a hybrid routing architecture including both unicast and multicast algorithms is needed.

Therefore, end users communicate with their local MCUs by unicast while all MCUs are connected by multicast. The intra-domain multicast are limited only between a few MCUs. This architecture requires many end users connecting to one single node (the MCU) by unicast. This suffers from single point failure and processing bottleneck, where backup MCUs might be required. However, this architecture enhances the scalability of the conference multicasting system, where many users are not part of the satellite multicast network.

Also, the low number of MCUs implies the possibility of using sparse mode routing since they can be sparsely distributed to cover each AS. Therefore, a sparse multicast routing protocol is needed in the scope of intra-domain. PIM-SM (see chapter 3) is recommended in this study.

The federated ISPs can access the satellite domain in two scenarios. In the first scenario, all of the
satellite terminals are owned by the satellite domain instead of the federated domains. To seamlessly interwork with the satellite domain to enable multicast in the whole system, the PIM-SM RP of each federated domain has to be located in satellite domain, i.e. collocated with the satellite terminal. All of the end users in federated domain have to connect to the RP in the satellite domain to join the multicast group. Thus, the inter-domain multicast can be ignored since all terminals are treated as a source from the view of the satellite domain and the multicast carried in the satellite domain becomes intra-domain multicast. However, using another domain’s RP is not acceptable for ISPs in the current market. Every ISP wants the complete administration over all intra-domain users. This is the reason that inter-domain routing protocols are useful. Therefore, the case that the satellite domain owns all terminals is not feasible.

The second scenario is the federated ISPs should have satellite terminals/gateways to access the satellite domain and reserve the satellite resources. From the routing point of view, these satellite terminals/gateways could belong to the federated ISPs or the satellite domain. If the federated ISPs own them, which implies that inter-domain routing has to be considered. Therefore, to employ the multicast techniques, which are strongly recommended for a satellite conference system, an inter-domain multicast solution is needed. Because of the star topology, the inter-domain multicast can be deployed more easily than in the terrestrial networks. The multicast routing tree in this environment can be as simply as a star free with a root in the satellite domain and its first level branches, which are terminated at the border gateway of each federated ISP domain.

5.1.1.2 Satellite Network

The satellite network works as the heart of the conference system and handled all inter-domain traffic. Basically its function is to forward incoming data to correct output interfaces to the routers connecting to each domain in the terrestrial networks. Therefore, the network structure is simplified. The two segments of the satellite, space segment and ground segment, play different roles to realise this forwarding function.

**Space segment**

GEO satellites are ideally for a conference in terms of their wide and stable beam coverage. However different orbit satellites have different coverage and characteristics such as LEO (Low Earth Orbit) and MEO (Medium Earth Orbit) satellites, which can be used to provide conferencing services as well with lower delays, but with more network complexity. This thesis focuses only on GEO satellites.

The space segment provides the physical part of the forwarding function. OBP (On-Board Processing) satellites maintain an entry table for the forwarding, which could be like a routing table in IP networks. Data transmitted to the satellite comes in from one input interface and are
forwarded to the destination ground satellite terminal via the correct output interface. Maintaining an entry table and forwarding multicast traffic are necessary for a satellite to support the multicast function. The multicast group address has to be associated with satellite output interface in the entry table. Another key function on board to support multicast is the capability to duplicate packets for more than one output interfaces or switching packets to all interface associated with a multicast group.

Compared to a real router, the satellite lacks the ability to intercept various routing information and cannot establish and update its entry table. The current best choice is to have the routing protocol support on the ground segment that can help the space segment update the entry table and switching for multicast.

**Ground segment**
The ground segment can play a powerful role in multicast routing where it can intercept routing information and provide control information to the satellite. It mainly consists of two types: the satellite terminals and the satellite Network Operation Centre (NOC). These two terminal types work together to enable the conference between different satellite spot beams.

As described all federated ISPs own their satellite terminals that closely locate to their PIM-SM RPs and border gateways. Each terminal opens a satellite channel connecting to the NOC to enable the ISP border gateway to communicate with the one in the satellite domain using an inter-domain routing protocol, such as Border Gateway Protocol (BGP). The routing information exchanged between these border gateway peers are used to establish the routing table in the NOC, which can be used to update the entry table on the satellite.

Therefore the NOC is the heart of the conference system in terms of routing decisions and group information if multicast used. For example, the NOC will provide all routing and multicast group information to all federated ISPs. This centralized star topology makes the network architecture much simpler and easy to understand while risks the single point failure and potential bottleneck. This risk cannot be avoided because of the nature of the satellite networks. However, the risk can be reduced by providing NOC redundancy and employing a hierarchical network architecture. To provide robustness over space segment failure, a conferencing system might have pure terrestrial network connectivity as a back up.

The hierarchical network architecture is aimed at distributing as many functions of the centre as possible to the branches. Therefore the whole network can be divided into several sub-networks. Each sub-network has its own sub-centre node while all of these sub-centre nodes are connected to the main centre of the whole system. Thus, when the main centre suffers from the single point failure, all the sub-networks can continue providing services and provide local information to the centre to help it recover. This concept can be fitted into the satellite conference system very well.
Each federated ISP can be treated as a sub-system who has full control to all its local network information while the satellite NOC plays both the satellite domain sub-centre and the main centre of the whole conference system. For the routing architecture, the Figure 5-1 shows clearly that each ISP AS sends its routing information to the satellite NOC and receives the information of other ASs from the NOC. Thus when the NOC fails, the conference service may continue. When the failure is fixed, the NOC can collect routing information that was stored in each ISP domain before the failure to recover soon.

To support multicast, this hierarchical architecture becomes more efficient considering group maintenance aspect. The group maintenance means realizing all group functions such as users join/leave a group and receive/send data from/to the group. Figure 5-2 shows an example of the hierarchical multicast architecture. In each AS, PIM-SM are deployed to establish the intra-domain multicast tree. The inter-domain multicast protocol MSDP are deployed to enable each federated ISP exchange the source information through the satellite NOC.

![Figure 5-2: Multicast hierarchical example](image)

Thus, each ISP AS has its own RP, named federated RP, which can store the intra-domain multicast information using PIM-SM and the other ISPs multicast source information informed by the satellite NOC. The NOC holds a Satellite RP that handles all source information of the whole system, which includes federated ISPs and the satellite network itself.

Improvement can be achieved based on this hierarchical architecture if we limit the routing information exchanged between each ISP and the satellite domain rather than between ISPs. This implies that the NOC become both source and receiver for each federated ISPs. Actually, the NOC collects network information, including routing information and control information, from all federated ISPs, and distributes conference information to each ISP. The media traffic is distributed directly by the satellite space segment to all ISPs.
Each ISP does not need to know what happened in the others and it only communicates with the NOC. The key difference between this improved hierarchical architecture and the original one is that communication terminates on the NOC in the new architecture rather than on each ISP's border gateway in the old one. For example, in the improved system, if the F-RP in ISP2 wants to join a multicast tree for a group named group 1, it sends request to the S-RP by assuming the source is in the satellite domain, where all multicast group information was collected. Then the S-RP reconfigures the satellite, i.e. update onboard entry table, to forward the media traffic of group 1 to ISP2 F-RP. In the old system, the request from ISP2 has to be transmitted to other ISP, say ISPy, to join the group 1. Thus, the MSDP protocol can be skipped and all F-RPs work as multicast users from the view of the NOC. Therefore, the protocol stack can be simpler as well.

With a good understanding of the satellite network, one can now put it together with the multicast technologies. As described in the hybrid routing architecture and the multiple MCU conference model that MCUs and multicast end users communicate with each other using multicast. Multiparty IP conference via satellite implies the possibility of widely geographically distributed participants. Therefore, both intra-domain and inter-domain multicast scenarios were studied. The design separate into two areas due to the different requirements. One is intra-domain multicast design and another one is inter-domain multicast.

5.1.2 Intra-domain Multicast for IP Conferencing over Satellite

As far as intra-domain multicast forwarding is concerned, according to what stated in Chapter 3, the adoption of the PIM-SM has been proposed. Two deployment scenarios have to be analysed in order to properly frame this protocol:

- PIM-SM deployment only over terrestrial networks of federated ISPs;
- PIM-SM deployment over both terrestrial networks of federated ISPs, and the satellite links (if any) between these network domains.

In the former case the following scenario arises:

- A far as terrestrial end users (both unicast and multicast) are concerned, PIM-SM procedures execution would completely framed in the multicast network domain of each federated ISP, such procedures would be carried out by the Designated Router (DR) interfacing terrestrial multicast sources/receivers of a given group G towards the RPs associated to this group located in the ISP domain. This entails that each federated ISP will autonomously manage its own RPs (no dependency on third-party RP), where group receivers will join the Rendezvous Point Tree (RPT), and group sources will join their own Shortest Path Tree (SPT). For terrestrial unicast end users the associated DR may be
located in the unicast-multicast domain boundary.

- As far as satellite end users are concerned, a membership protocol (namely IGMP, see Chapter 3) would be executed either locally, if a multicast router is co-located with each satellite terminal (Satellite enabled Multicast Router-SMR), or would be proxied at NOC if satellite terminals are not provided with an SMR. Concerning with PIM-SM, when an SMR is co-located with a satellite terminal, the associated sessions should be accomplished between the SMR and a so called Satellite Rendezvous Point (SRP) located in the NOC. Otherwise, PIM-SM signalling should be proxied at NOC as well.

Figure 5-3 tries to summarise what stated above. In particular, in the context of the ISPk network domain both IGMP and PIM-SM have been pointed out (in red path and brown path respectively).

![Figure 5-3: PIM-SM deployment of the IP conference over satellite network with satellite terminals provided with multicast router](image)

For terrestrial multicast end users belonging to the ISPk multicast domain, IGMP takes place between them and their respective Multicast Router (MR), namely their Designated Router (DR), while PIM-SM is carried out between each DR and the RP of the multicast group, which for the sake of clarity has been considered co-located with an EMR.

As far as ISPk terrestrial unicast end users are concerned, their membership protocol will be carried out in different ways depending on the Multiple MCUs model for multiparty
videoconferencing. IGMP shall take place between the MCU and the associated DR that is
supposed to be located in the unicast-multicast domain boundary as shown in Figure 5-3.

When dealing with multicast routing, PIM-SM sessions associated to terrestrial unicast/multicast
end users in ISP_k will take place between the MR located at ISP_k Unicast-Multicast domain
boundary and the RP of the multicast group as shown by brown-dashed lines in Figure 5-3.

Finally, as far as satellite end users are concerned, Figure 5-3 shows satellite end users provided
with an SMR, therefore IGMP sessions may be locally carried out as detailed in Figure 5-4, while
PIM-SM sessions are performed between each co-located SMR and the SRP of the NOC.

![Figure 5-4: Satellite enabled Multicast Router co-located with satellite terminal](image)

Concerning satellite communication resources, it is to be stressed that satellite multicast end users
require a point-to-point bi-directional satellite traffic channel to the NOC in order to carry out
either only PIM-SM sessions or IGMP sessions. This satellite traffic channel should be “logically
available”, which should be automatically set-up at terminal registration phase and left “silent” as
long as the satellite terminal is switched on. Whenever multicast “signalling” is going to be sent to
the NOC, physical resources should be required to the Traffic Resource Manager (TRM) on­
board, and released as soon as the signalling flow ends.

It can be extended to the case of satellite end users without a local SMR. In this case the IGMP
sessions will take place between satellite end users and a SMR located in the NOC. The
associated PIM-SM sessions will be proxied at NOC.

Each federated ISP domain provided with satellite access may have its own RP active for a given
group, without forcing sources/receivers of a group to join a remote RP. This means that in the PIM-SM will be deployed only over terrestrial network domains, while inter-domain mechanisms will be used over the satellite links.

### 5.1.3 Inter-domain Multicast for IP Conferencing over Satellite

In IP conference over satellite system, PIM-SM/Multicast Source Discovery Protocol (MSDP)/Multiprotocol Border Gateway Protocol (MBGP) is a general multicast routing framework from practice perspective because it is a functional solution largely built on existing protocols. Furthermore, it is already being deployed with a fair amount of successes. In particular, MBGP should play the role of multicast routing protocol while MSDP is responsible of the forwarding process between multiple PIM-SM domains.

To tackle the MSDP deployment shown in Figure 5-5, a single hop peering seems to fit well to the IP conferencing over satellite context. In particular, each MR (multicast router), which plays the role of both RP and MSDP/MBGP peer, should peer with the SRP (Satellite Rendezvous Point) residing in the NOC. In turn, the SRP should peer with all MR belonging to the confederation. This way, Source Active (SA) messages are firstly sent to the SRP of the NOC by the RP of the ISP domain where the multicast source is, and then forwarded to all the RPs of the confederation that are directly reachable through satellite links. This means that two satellite hops are necessary in order to advertise each SA message to all the confederation RPs. Such traffic may be carried out over the same point-to-point bi-directional satellite traffic channel used to carry out MBGP sessions. An alternative scheme could consist in deploying MSDP mesh groups, but in this case a greater number of “latent” satellite traffic channels would be required. In fact, if \( N \) is the number of RPs to be peered, then the former scheme requires \( N \) point-to-point bi-directional satellite traffic channels for both MBGP and MSDP sessions execution, while the latter requires

\[
\binom{N}{2} + N
\]

point-to-point bi-directional satellite traffic channels: the MSDP mesh group has

\[
\binom{N}{2}
\]

the combination of any 2 elements in total of \( N \), satellite channels, and \( N \) channels for MBGP peering with NOC BGP peer. Obviously the former scheme has the drawback of requiring two satellite hops, but since this MSDP does not carry multicast traffic, then this is a minor drawback compared to the waste of satellite logical resources required by the second scheme.
Chapter 5. Routing Architecture for Conference over Satellite

The following paragraphs will describe the deployment of both MBGP and MSDP in the IP conferencing over satellite system.

5.1.3.1 MBGP Deployment for IP Conferencing over Satellite

MBGP deployment towards terrestrial end users belonging to the confederation ISPs consists in a straight upgrading of the BGP protocol (namely BGPv4). It is deployed in the frame of the confederation. This means that MBGP sessions should take place between each federated ISP domain and the Satellite domain, and therefore BGP peers should be located both in the federated ISPs domains and in the NOC where the Routing Arbiter (RA) of the confederation runs. In order to accomplish MBGP sessions over the satellite network, point-to-point bi-directional satellite traffic channel shall be available for BGP sessions in the confederation.

Concerning satellite end users, since, from the network domain standpoint, they belong to the Satellite AS domain, then the BGP speaker of the NOC will be in charge of advertising to the federated ASs reachability information to the satellite end users. Their BGP peer is coincident with the BGP speaker of the NOC, namely the RA.

Thanks to MBGP sessions occurring as described above, two Routing Information Bases (RIBs) will be populated into each BGP peer:

- The Unicast Routing Information Bases (URIB), which contains unicast prefixes for unicast forwarding, and which is populated with BGP unicast NLRI (Network Layer Reachability Information) (AFI (Address Family Identifier) = 1, Sub-AFI (Subsequent Address Family Identifier) = 1 or 3).
• The Multicast Routing Information Bases (MRIB), which contains unicast prefixes for RPF checking, and which are populated with BGP multicast NLRI (AFI = 1, Sub-AFI = 2 or 3)

A point-to-point bi-directional satellite traffic channel should be made available between each satellite enabled BGP peer and the NOC BGP peer in order to properly carry out the MBGP sessions inside the ISPs confederation.

The dashed-double dotted purple path in Figure 5-6 summarises what stated above about MBGP protocol. MBGP protocol sessions associated to ISP_k of AS_n are accomplished between the SMR of ISP_k terrestrial network domain and the RA of the NOC. Note that without lack of generality, the satellite enabled BGP peer of the federated ISP domain has been supposed coincident with the SMR of the federated ISP_k multicast network domain. In Figure 5-6 satellite end users provided with multicast router capabilities have been considered that their BGP peer resides into the RA of the NOC. Obviously, even if satellite end users do not have multicast routing capabilities, their BGP peer still resides in the RA of the NOC since it plays the role of BGP peer on behalf of every satellite end user (pure satellite network) belonging to the confederation.

Figure 5-6: MBGP/MSDP deployment over the IP conferencing over satellite system with satellite terminals provided with Multicast Router
5.1.3.2 MSDP Deployment for IP Conferencing over Satellite

Concerning with MSDP deployment over the IP conferencing over satellite network, one has to consider what has been said above about intra-domain multicast routing deployment. Since the MSDP protocol should take place by means of TCP connections between MSDP peers, which are likely to be congruent to the connections in the BGP routing system, for each federated ISP, both MSDP peer and BGP speaker can be co-located in the same network node. Moreover, each federated ISP multicast domain is provided with (at least) an Satellite enabled Multicast Router (SMR) which can also play the role of RP in the context of the terrestrial multicast ISP domain. This RP may be either unique or can be one of the possible RPs available for a given group in the PIM-SM multicast domain of the federated ISP. Without lack of generality one can consider both the functionality of RP co-located with the SMR of each federated ISP domain, associated to operation of PIM-SM, and the functionalities of BGP/MSDP peers. This concept is shown in Figure 5-6 by means of the dashed-dotted blue path.

For satellite end users, both with and without multicast routing capabilities, the conclusions about MBGP still apply in the MSDP case. Their MSDP sessions take place between the SRP of the NOC and each SMR belonging to federated ISP terrestrial multicast domain. It is to be noticed that, for satellite end users without SMR, MSDP sessions associated to an active satellite multicast source are triggered by the SRP of the NOC when it receives a PIM REGISTER message from the co-located PIM proxy. For satellite end users with SMR, such message is sent directly by the SMR co-located with the satellite terminal to the SRP of the NOC. The same considerations can be made when satellite multicast sources leave the multicast group. Figure 5-6 sketched this notion by taking into account also the PIM-SM protocol deployment in the IP conferencing over satellite network for both types of satellite terminal (with and without SMR).

It can be concluded by stating that for whatever satellite end user, both MBGP sessions and MSDP sessions towards terrestrial federated ISPs domains take place between the SRP of the NOC and each SMR, supposed to be both BGP and MSDP peer, located into the federated ISP terrestrial multicast domain.

5.2 Step-by-Step Conference Setup Procedures over Satellite

Using the routing designed for the IP conference system, a conference can be established. This section presents the setup procedure step by step. All the procedures described here is for general satellite. This section will not describe the detail satellite channel setup procedure considering its generality. However, it will mention very briefly the corresponding steps if necessary to keep the connection with the routing architecture.
The process of the multiparty multicast conference service scenarios for general satellite network is shown below:

**Preconditions:**

All ASs including federated ISPs and the satellite domain implement the PIM-SM/MSDP/MBGP multicast solution.

### 5.2.1 New conference establishment

Considering the two different end users types, a *source* is defined in this subsection as either a multicast end user or an integration of a unicast end user and its local MCU.

There are two ways to establish a conference:

- **Pre-arranged conference**, in which the conference organizer pre-configured a conference with a fixed multicast group address and distributed it on the Internet by variant means. All conference users join the conference by obtaining this multicast address. The RPs of this multicast group in the terrestrial networks are located using any exiting location mechanisms.

- **Source initiated conference**, in which a source obtains a multicast address for a conference and registers to the associated RP. For the case of unicast end users and its MCU, the user has to dial another user with its MCU involved. Thus, the MCU can obtain a multicast address for the conversation triggered by the signalling message.

### 5.2.2 New sources join a conference

Sources in the satellite domain and terrestrial network joining a conference are not exactly the same.

**Terrestrial source join a conference:**

1. The unicast end users dial their local MCUs to inform which multicast group they want to join.
2. If this source is not the first one in this multicast group connecting to the MCU, the MCU will begin to mix the multimedia streams from this new source with the other sources and send the mixed stream out. The join procedure finishes. If this source is the first one in the multicast group connecting to this MCU, the MCU will send the media data to the DR, which is the nearest multicast enabled router. The multicast enabled end users sends data directly to their designated router (DR) using IGMPv2.
3. Upon receipt of the multicast data, it register-encapsulates and unicasts the data packet to
the RP.

iv) Then the RP will send out SA messages to inform all its MSDP peering that a new source is active for the multicast group. For the ISPs who don't have satellite access, one of the SA message will be sent to another ISP who has satellite access, where it is forwarded to the satellite domain. The SA messages generated in the satellite-connected ISPs will be directly sent to the satellite domain through MSDP peering between the local RPs and the satellite RP. All of the inter-domain routings are based on the MBGP.

v) From the satellite RP, the SA messages are forwarded to all the other federated ISPs that have MSDP peering associated with this multicast group.

vi) Those federated RPs who are listening to this group will then send JOIN messages to the source RP through satellite channels. However, for the listeners in the satellite domain, the satellite RP should trigger the satellite channel establishment to add branches for each of the satellite terminals, which has a set of group listeners. Further discussion happens in step iii.

vii) These JOIN messages will trigger the p2mp channel establishment that can add branches to the group-shared tree via the satellite to those interested RPs.

viii) The receiver RP begins to receive the media data from the new source.

Figure 5-7 shows the procedure of a new terrestrial unicast sources joins a conference described above from step i to step viii.
Satellite sources join a conference:

i) If satellite sources have their local RPs, then the join procedure is the same as terrestrial sources.

ii) If satellite sources don't have local RPs, they have to register to the RP located in the NOC through satellite. It is recommended that this RP is the satellite RP. Thus, the MSDP peers might be only between federated RPs and the satellite RP.

iii) The procedure between end users and their local MCU and DR is the same with terrestrial source does.

iv) The DR register-encapsulates the data received from the MCU and unicast them to the satellite RP, which triggers an SA generation.

v) The SA message is forwarded to all interested MSDP peers including federated RPs and the satellite RP itself.

vi) The federated RPs send JOIN (S, G) message to the ‘S’ indicated in the SA message through a satellite channel.
vii) The satellite RP is responsible to add branches for all listeners in the satellite domain that is triggered by receiving the SA message. This can be done by directly informing the NOC to update the satellite entry table or by using MSDP proxy to inform all the satellite end user terminals to begin the multicast tree branch setup procedure.

Figure 5-8 shows the procedure that a new satellite unicast sources joins a conference as described above from step iii to step vii.

Figure 5-8 The procedure of a new satellite unicast sources joining a conference

5.2.3 New receivers join a conference

The procedures for terrestrial receivers joining a conference over one satellite:

i) For the unicast end users, they have to dial their local MCU to join a conference. For multicast enabled end users, they send IGMPv2 report to their DRs that will send JOIN message to local RP.

ii) If the MCU has already listened to this multicast group, it will simply send unicast media data that it mixed to the end users. Otherwise, it will join the multicast group by sending IGMPv2 report to their DRs who will send JOIN message to local RP.

iii) If the RP has already been in the multicast tree, it will simply forward the data of that multicast group to the DR. Thus a new branch was established between the DR and the RP. Otherwise, the RP will check the MSDP SA cache for the source information of the
Chapter 5. Routing Architecture for Conference over Satellite

requested multicast group, i.e. the (S, G) entry within the SA messages. Then the RP triggers a (S, G) JOIN event towards the source RP, which could be any federated RP or the satellite RP through a p2p satellite channel.

iv) The remaining steps are the same with the steps vii and viii for terrestrial sources and the step vii for satellite sources.

Figure 5-9 shows the procedure of a terrestrial unicast receiver joining a conference as described above from step i to step iv.

Satellite receivers join a conference:

i) If a satellite end user wants to join a conference, it will dial its local MCU, the same as the terrestrial end user does. If the MCU is already in the multicast tree, it will begin to send the data of this multicast group to the end user. Otherwise, MCU will join to the satellite RP.

ii) The satellite RP checks its MSDP SA cache for current active sources in this multicast group.

iii) Two ways to establish the p2mp branch for the end user:

    a. The local DR of the MCU has MSDP peer through a proxy, like the ESW does, with the satellite so that the satellite RP can sends the active sources' IP address to the DR. then the DR send the (S, G) JOIN messages to all of the sources to establish multicast tree branches as shown by the steps vii and viii for terrestrial
sources and step vii for satellite sources.

b. The satellite RP have interface with the NOC so that the RP can trigger the update of the satellite entry. This is recommended to avoid MSDP functionality at the DRs of the satellite users. And the add-branch procedure can be much simplified without two satellite terminals, the source one and the receiver one involved.

Figure 5-10 shows the procedure of a satellite unicast receiver joining a conference as described above from step i step iii.

![Figure 5-10 The procedure of a satellite unicast receiver joining a conference](image)

5.2.4 Sources leave a conference

Terrestrial sources leave a conference:

i) The MCU stops sending data to the RP after its last local unicast end source stop sending data to the multicast group.

ii) The periodic SA message will be stopped between the RP and the satellite RP MSDP peer.

iii) If the SA message stops for a period (timeout) time, it also can be after a period of time without any media data going out, the p2mp satellite channels will be tired down and all satellite resources will be released.

Figure 5-11 shows the procedure of a terrestrial unicast source leaving a conference as described above from step i to step iii.
Satellite sources leave a conference:

i) The MCU stops sending data to the DR after its last local unicast end source stop sending data to the multicast group.

ii) If there is MSDP peering between the DR and the satellite RP, the SA will be stopped, which after a timeout can cause the tear-down of the p2mp satellite channel.

iii) If no MSDP peering between the DR and the satellite RP, the absence of the outgoing media traffic will trigger the tear-down procedure.

The sequence chart of the procedure of a satellite unicast source leaving a conference is very similar to the procedure of a terrestrial unicast source leaving a conference and it is not plotted here.

5.2.5 Receivers leave a conference

Terrestrial receivers leave a conference:

i) The receiver has to tear down the signalling connection with the MCU. And the MCU will send Leave Group message to its DR if all of its local receivers in the multicast group leave.

ii) Then the DR sends PRUNE (*, G) message to its RP.

iii) The RP stops forwarding the data of that multicast group to the DR. If there are no more downstream branches for the multicast group, the RP will clear its (*, G) entry and the PRUNE (*, G) message will trigger the satellite channel tear-down procedure.

Figure 5-12 shows the procedure of a terrestrial unicast receiver leaving a conference as described above from step i to step iii.
Satellite receivers leave a conference:

i) The satellite end users have to tear down the signalling connection with their local MCUs. If there are no receivers for a multicast group, the MCU will send a Leave Group message to its local DR to delete the multicast tree branch from the DR to the MCU.

ii) If the DR is not a local multicast router but in the NOC, the Leave Group message will be sent to the NOC via a satellite channel, which will trigger the satellite channel tear-down procedure.

iii) The local DR will send the PRUNE (*, G) message to the satellite RP if it has no local receivers for the multicast group, which will trigger the satellite channel tear-down procedure.

Figure 5-13 shows the procedure of a satellite unicast receiver leaving a conference as described above from step i to step iii.
Chapter 5. Routing Architecture for Conference over Satellite

Figure 5-13 The procedure of a satellite unicast receiver leaving a conference

5.3 Summary

This chapter presented a proposed hybrid routing architecture for multiple MCU conference over satellite in the scope of ICEBERGS project (IST-2000-31110), which was an EU IST project under the Fifth Framework Programme. The architecture consisted of both unicast and multicast routing technologies. It was designed to accompany the multiple MCU conference model proposed in the previous chapter.

The hybrid routing architecture can minimize the requirement on the end user equipment, which does not need high power for mixing media streams from several sources active in the conference but receive the mixed stream from their local MCUs. All of the MCUs connect with each other using multicast, which can enable one-to-many communication and save the uplink bandwidth if via satellite. It also enables the end user to receiver the media data sent by specific source that he is interested in by unicast. This approach was also designed with the consideration that multicast infrastructure cannot be easily established due to the huge works on the equipment replacements and reconfigurations. It can minimize this kind of work because the multicast domain is only needed among MCUs. It is easy to integrate with and/or migrate to a full-multicast enabled network in the future.

Both intra-domain and inter-domain multicast scenarios were studied with the consideration of the satellite network. PIM-SM was deployed over terrestrial networks of federated ISPs as well as satellite domains to enable intro-domain multicast. MSDP and MBGP were deployed between terrestrial domains and the satellite control centre to announce active source and carry the multicast data to all receivers in the same group belong to different domains. The satellite network control centre acts as a central routing management entity that receives MSDP messages from all domains and configures the satellite to set up the one-to-many connections for multicast communication. It can also work as a RP proxy for satellite domain users who do not have local RPs.

At the end of this chapter, how to establish a conference step by step was presented in the designed system using the proposed conference model and routing architecture.
6 Measurement and Evaluation of IP Conferencing System over GEO Satellite

The hybrid routing architecture makes the conferencing service over GEO satellite feasible at the network level. And the multi-MCU conference model is designed to enable this service to take advantage of SIP for conference control functionalities. However, there is another key aspect that should be mentioned here, the performance. As mentioned in the first chapter, the VoIP application will not win its final success in the fighting with the traditional telephony services without an acceptable performance. In this chapter, some concepts of the IP network traffic measurement will be introduced that is the base to measure the performance for multiparty communications. Then some parameters will be discussed that are used to measure the end-to-end performance. Some major measurement methodologies will then be presented that can be used to measure these parameters. With the IP network traffic measurement background, a test plan was designed and carried out over a demonstrator built within the ICEBERGS project to completely validate the proposed technologies in the previous chapters. The demonstrator architecture and a detailed test plan will be presented. The objective measurement results will be analysed to show the success of the designs described in previous chapters.

6.1 IP Network Traffic Measurement

IP network traffic measurement is crucial to the traffic engineering function. It provides the means to have the insight of the network operation state and problem anticipation. It is useful for optimising the network because it can provide the feedback data for the engineer to adaptively optimise network performance in response to events and stimuli originating within and outside the network. It is essential to determine the quality of network services and to evaluate the effectiveness of traffic engineering policies. And experience indicates that measurement is most effective when acquired and applied systematically [43].

To deploy the measurement on a network, one has to address the following questions:

- Why is measurement needed in this particular context?
- What parameters are to be measured?
- How should the measurement be accomplished?
- Where the measurement should be performed? When should the measurement be
Chapter 6. Measurement and Evaluation of Conference over Satellite

performed?

- How frequently should the monitored variables be measured?
- What level of measurement accuracy and reliability is desirable?
- What level of measurement accuracy and reliability is realistically attainable?
- To what extent can the measurement system permissibly interfere with the monitored network components and variables?
- What is the acceptable cost of measurement?

The answers of these questions will determine what measurement tools and methodologies are suitable for the particular engineering context.

Measurement in support of the TE function can occur at different levels of abstraction. For example, measurement can be used to derive packet level characteristics, flow level characteristics, user or customer level characteristics, traffic aggregate characteristics, component level characteristics, and network wide characteristics [43].

6.1.1 Traffic Measurement Methodologies

Lots of research has been done to develop measurement methodologies, e.g. using LOG files and capturing packet from the Internet using some software and hardware. The measurement methodologies can be divided into two main groups: passive approach and active approach. Both have their values and should be regarded as complementary, in fact they can be used in conjunction with one another.

6.1.1.1 Passive Measurement

The passive measurement approach implies to use devices, to monitor the traffic when it passes by. These devices could be some specific tools such as sniffer hardware, or they can be pure software built into some network equipments such as routers, switches and end node hosts. Examples of such built in techniques include Remote Monitoring (RMON) [46], which enables various network monitors and consoles to exchange the monitored network data using a kind of database named Management Information Base (MIB), and Simple Network Management Protocol (SNMP) [48][47], which is one network management protocol by using specific messages and MIB, capable devices. The passive measurement will not create or modify traffic on the network. This is in contrast to active measurement, in which specific test packets are introduced into the network, and these packets are timed as they travel through the network being measured. The basic principle of the passive measurement can be shown in Figure 6-1.
Figure 6-1: Basic principle of passive measurement

Figure 6-1 shows that two Entities are connected via an IP network through network links. On the path of the connection, the traffic between these two Entities is measured at a measurement point. The “Entity” in Figure 6-1 can represent various network situations. For example, the two Entities could be two end users or one organization internal network and the external Internet or two routers and so on. They could also be two part of the Internet, e.g. the backbone and one edge network.

Passive measurement can provide a set of detailed information of the interested traffic at one point in the network that is being measured. Examples of the information passive measurements can provide are:

- Traffic / protocol mixes
- Accurate bit or packet rates
- Packet timing / inter-arrival timing

It can also be deployed as a network application debugging method by capturing the entire packet contents. All of these advantages make the passive measurement valuable in network troubleshooting, single node behaviour study, source modelling and capacity management. It requires collecting the data and the traps and alarms all generated network traffic, which can be substantial. Moreover, the gathered data can be substantial especially if one is trying to capture information on all packets flowed in a network.

There are two major categories that passive measurement systems can fall into. The first is on-line processing that deals data in the real time. For instance, observing packet number and type, throughput in a period of time and so on. It is useful to monitor the instantaneous network status and bandwidth utilization situation. The second category is off-line processing that enables the monitoring devices save the captured packet into trace files as well as additional information such as arrival time. These trace files are processed and analysed after the measurement.

The on-line processing requires very powerful devices to capture the packet while doing extra calculation when monitoring a high-speed network. For example, one wants to know the instantaneous throughput using a graph drawing based on the captured traffic flow. Normally, the monitor device should save captured packets in a buffer and update the throughput graph
periodically. When the link is heavily loaded, say up to hundreds megabytes per second or even more, it is highly possible that the monitoring device might drop packets passing by while it has to calculate the throughput and update the graphic user interface. It could be worse when users want to know many instantaneous parameters. Therefore, analyzing the traces off-line will be much suitable though it costs more storage space to save packets and relevant additional information. One can spend more time to derive more details, such as interarrival time, packet lost rate, flow distributions and so on, from these saved traces. People have tried to save partial packets instead of its entire contents in order to save both processing time and storage space. One very common subset of the data that is saved is the IP header and transport layer headers. One other common subsection of data captured is the data link layer headers. This is used primarily in ATM networks, but this type of capture has limited use for IP level analysis. The IP header provides information on the source of the datagram, the destination of the datagram, the length of the datagram and which transport protocol is carried in the payload. The transport layer can give an indication of what type of traffic was contained within the packet.

Header traces are commonly used for both of the on-line and off-line passive measurement configurations discussed, and wherever the network speed allows traces to be taken. Full capture of all packet data on a link is normally restricted to the on-line process situation. The data rates created by a single computer are low when compared to backbones and gateways. Full capture allows complete analysis of the actual data passing on the network, which could be used for debugging purposes and also allow later 'playback' of the entire data stream.

There are some projects that are passively measuring the Internet and saved the monitored traffic in trace files for research purposes. These projects include Passive Measurement and Analysis (PMA), a National Laboratory for Applied Network Research (NLANR) Measurement and Network Analysis Group research project located at San Diego Supercomputer Centre (http://pma.nlanr.net/PMA/) with around 22 measurement sites, and Widely Integrated Distributed Environment (WIDE), which forms a shared research platform connecting about 140 organizations and provides various raw data traces from 1999 till now (http://www.wide.ad.jp).

**6.1.1.2 Active Measurements**

The active approach relies on the capability to inject test packets into the network or send packets to servers and applications. It increases the network traffic. The volume and other parameters of the introduced traffic are fully adjustable and small traffic volumes are enough to obtain meaningful measurements.

Active measurement provides very little information about a single point of a network. It instead provides a representation of the characteristics of the entire network path between two hosts. Active systems can provide such indications of a networks performance as:
- Packet round trip time (RTT)
- Average packet loss
- Connection bandwidth

Some active systems can also give indications of the following:

- Asymmetric delay times
- Alterations in routing paths between hosts

The active approach provides explicit control on the generation of packets for measurement scenarios. This includes control on the nature of traffic generation, the sampling techniques, the timing, frequency, scheduling, packet sizes and types (to emulate various applications), statistical quality, the path and function chosen to be monitored. Being active implies testing what you want, when you need it. Emulation of scenarios is easy and checking if QoS or Service Level Agreements (SLA) are met is relatively straightforward.

There are several public projects deploying active measurement. They can be divided up by whether they make one-way measurements or round trip (two-way) measurements. Surveyor project (http://www.advanced.org/surveyor/) and Réseaux IP Européens (RIPE) (http://www.ripe.net/ripe/) make one-way delay measurements and require a Global Positioning System (GPS) to provide clock synchronization between sites. NLANR Active Measurement Program (AMP) (http://amp.nlanr.net/AMP/) & Ping End-to-end Reporting (PingER) (http://www-iepm.slac.stanford.edu/pinger/) make two-way measurements using the Internet Control Message Protocol (ICMP) ping facility today, and do not require a GPS. The Cooperative Association for Internet Data Analysis (CAIDA) using a measurement tool named skitter (http://www.caida.org/tools/measurement/skitter/) aims more at global Internet measurements and so tends to be more generic than the others and in some ways the most unique among the five.

Passive measurement and active measurement are very different in terms of increase of network traffic and measurement objective. However, they are complementary rather than against each other. For example, the active measurement probe can schedule passive measurements of appropriate metrics at appropriate points along the path, while the active measurements are being made. When the active measurement is completed then the appropriate passive measurements can be paused thus reducing the gathering of unnecessary data. By comparing and contrasting the active and passive measurements, the co-validity of the different measurements can be verified, and much more detailed information on carefully specified/scheduled phenomena is made available. It is very common that one may need both measurement results to gain the final conclusion.
These measurement methodologies show how to measure the IP network traffic. Performance parameters will tell what to measure.

6.1.2 One-to-one Performance Parameters

To determine what parameters are needed to measure is the most important factor before one launch his measurement. It is the key to decide measurement tools, methodologies and accuracy. The Internet Engineering Task Force (IETF) IP Performance Metrics (IPPM) working group developed a set of standard metrics that can be applied to the quality, performance, and reliability of Internet data delivery services. Another working group named Benchmarking Methodology (BMWG) made a series of recommendations concerning the measurement of the performance characteristics of various internetworking technologies, which includes terminology, identifying a set of metrics that aid in the description of traffic characteristics, and methodology, required to collect said metrics. Additionally, the ITU-T Working Group T1A1.3 made similar network performance parameter definition [44].

The IPPM developed a set of measurement parameters as well as the correspondent measurement methodologies with the cooperation with other relative working group such as BMWG, TEWG, ITU-T SG 12 and SG 13. Those parameters include:

- **Connectivity**: If a packet transmitted from source A to destination B at time T can arrive B, it is said that A has the connectivity to B at time T.

- **One-way delay**: The difference between the time when the source sends out the first bit of the packet and the time when the destination receives the last bit of the packet.

- **One-way loss**: If a packet transmitted from source A to destination B cannot arrive B in a certain time threshold, it is said that this packet is lost.

- **Round-trip delay**: The sum of the times needed for a test packet travel from source A to destination B and from B back to A.

- **One-way delay variation**: The difference of the one-way delays of a selected pair of packets in the stream going from source A to destination B.

- **Loss patterns**: The packet loss distribution.

- **Bulk transport capacity**: The expected long term average data rate (bits per second) of a single ideal TCP implementation over the path in question.

The IPPM defined a general framework [45] for particular parameter metrics that can be deployed to gain common understanding by Internet users and Internet providers of the performance and reliability both of end-to-end paths through the Internet and of specific 'IP clouds' that comprise
portions of those paths. The term “metric” is defined as a carefully specified quantity that is relative to the Internet performance and reliability people is interested in. It recommends defining particular metrics under some criteria and disciplines in order to allow people to speak clearly about Internet traffic performance. In several IETF meetings criteria for these metrics have been specified as follow [45]:

- These metrics must be concrete and well defined,
- A methodology for a metric should have the property that it is repeatable: if the methodology is used multiple times under identical conditions, it should result in consistent measurements,
- The metrics must exhibit no bias for IP clouds implemented with identical technology,
- The metrics must exhibit understood and fair bias for IP clouds implemented with non-identical technology,
- The metrics must be useful to users and providers in understanding the performance they experience or provide,
- The metrics must avoid inducing artificial performance goals.

Each parameter metric will be defined in terms of standard units of measurement. The international metric system will be used, with the following points specifically noted [45]:

- When a unit is expressed in simple meters (for distance/length) or seconds (for duration), appropriate related units based on thousands or thousandths of acceptable units are acceptable. Thus, distances expressed in kilometres (km), durations expressed in milliseconds (ms), or microseconds (us) are allowed, but not centimetres (because the prefix is not in terms of thousands or thousandths).
- When a unit is expressed in a combination of units, appropriate related units based on thousands/thousandths of acceptable units are acceptable, but all such thousands/thousandths must be grouped at the beginning. Thus, kilometres per second (km/s) is allowed, but meters per millisecond is not.
- The unit of information is the bit.
- When metric prefixes are used with bits or with combinations including bits, those prefixes will have their metric meaning (related to decimal 1000), and not the meaning conventional with computer storage (related to decimal 1024). In any RFC that defines a metric whose units include bits, this convention will be followed and will be repeated to ensure clarity for the reader.
Chapter 6. Measurement and Evaluation of Conference over Satellite

IPPM gives 6 sets of standardized metrics for the following parameters under the above criteria:

- Metrics for measuring connectivity [50]
- One-way delay metric [51]
- One-way packet loss metric [52]
- Round-trip delay [53]
- One-way loss pattern [54]
- Packet delay variation [55]

Each of these metrics is normally defined with three sections including metric name, metric parameters and metric units. The metric name contains basic information of the measurement such as packet type, unidirectional or bi-directional, and parameter name. The metric parameter section defines what traffic parameters should be recorded in the metric that can be used for further analysis. The metric unit part describes the unit type of the metric. For instance, the one-way delay metric is named “Type-P-One-way-Delay” that means packets measured in this metric are all type P packet where P could be protocols such as TCP, UDP and ICMP. Its metric parameters are $Src$, the IP address of the packet source, $Dst$, the IP address of the packet destination, and $T$, the time the source sent out the first bit of the type P packet.

Corresponding to each metric, at least one measurement methodology is defined to acquire data from the network. These methodologies should have the property that it is repeatable: if the methodology is used multiple times under identical conditions, it should result in consistent measurements or continuity results with small variations.

These traffic parameters and their measurement methodologies were defined by IETF for the purpose of network performance and reliability analysis. They are vital for the network evaluation, especially QoS evaluation. Some of these metrics will be introduced that are relevant to later work one by one in the following paragraphs.

**One-way delay**: The definition of one-way delay of a packet is the difference between the time when the source sends out the first bit of the packet and the time when the destination receives the last bit of the packet (whenever a time, i.e., a moment in history, is mentioned in this document, it is understood to be measured in seconds (and fractions)) [51].

**Packet delay variation (jitter)**: The one-way delay variation of a pair of packets within a stream of packets is defined as the difference of the one-way delays of a selected pair of packets in the stream going from measurement point MP1 to measurement point MP2 [55].

**Round-trip delay**: The round-trip delay is defined as the sum of the times needed for a test
packet travel from the source to the destination and from the destination back to the source [53].

**One-way packet loss:** If a test packet does not arrive at its destination in a time threshold, it is defined lost [52].

### 6.1.3 Relevance to Conferencing over Satellite

The active measurement methodology implies that the measurement has to be taken at end hosts. Therefore, it is reasonable that IPPM recommends using active measurement methodology to measure all of the end-to-end parameters they proposed. IPPM recommends to use test traffic with fixed packet size. It is understandable that the RFCs are for a very general scope of network measurement scenarios and the fixed packet size can eliminate the consideration of the difference packet size distribution of all kinds of applications. Therefore, evaluating network performance can be easier to carry out without differentiation of each application. However, when studying the network designed particularly for one or more services, e.g. multimedia conferencing system, we have to take its packet size distribution into our consideration when designing the measurement process. This is because of the routing delays will verify with different packet size. And therefore, it will affect the delay variance in the measurement. Moreover, the RFCs do not mention any concern on the service differentiation. That means the measurement result might depend on the test traffic priorities. Considering the fact that the service differentiation functions were mostly implemented in the network rather than end host, the test traffic might not trigger those functions at all in the measurement. Therefore, the best choice for evaluating networks designed for some particular services is to use real application traffic as test traffic in the active measurement. This can be treated as the mixture of both active and passive measurement methodologies. That's why the measurement carried out over the conferencing testbed described later in this chapter used the real application traffic during the whole test campaign.

### 6.2 Conferencing Demonstrator Architecture and Implementation

This section will introduce the Conferencing over satellite demonstrator's general architecture from a system view. The innovative aspects and the advancements of the designed system were demonstrated via a fully representative configuration of the hybrid satellite/terrestrial network architecture. In addition, a satellite emulator was used in the demonstrator to provide complementary onboard switching functionality. The overall demonstration of the functions, signalling service, multicast media transmission and performance, using satellite terminals and a representative Internet, including core and edge routers sub-networks, allows the full characterization of the system choices, with possible feedback at system and element design level. A comprehensive evaluation of the trials' results was carried out systematically based on the
collected measurement data.

The demonstrator prototype consists in an end-to-end network composed of remote nodes connected with a Federated Access network through satellite access.

The demonstrator prototype implementation is based on two scenarios:

- Demonstrator with the actual satellite access network.
- Demonstrator with the EuroSkyWay (ESW) emulator having the ESW onboard switching functionalities

Figure 6-2 depicts the demonstrator prototype general layout, showing the satellite end user side (on the left in the figure) and ICEBERGS Federated Access Network (I-FAN) section (on the right). The satellite environment is valid for the actual and emulated mode.

Three Federated Internet Service Providers (F-ISP) and a Core Network, emulating the connection backbone, compose of the I-FAN section. Each F-ISP can host a traffic generator to load the network during the test phase.

The scenario with the actual satellite access used SESAT 1 satellite (Ku band transparent satellite positioned at 36° East of longitude) launched by European Telecommunications Satellite Organization (Eutelsat). This scenario will make available the possibility to analyse the effect on the service of the actual satellite geostationary delay and the on the sky availability.

Figure 6-3 shows the two remote node configurations considered in the demonstrator prototype. On the left side, the unicast configuration is shown, characterised by the presence of the MCU system. The right side picture presents the multicast enabled remote node configuration.
Before continuing with the description of the main elements constituting the I-FAN and having the intention to facilitate the demonstrator system understanding, avoiding any confusion in the sub-systems description, a set of symbols have been defined and introduced; they are shown in Figure 6-4.

The final layouts of the I-FAN implemented in the laboratory are shown in Figure 6-5 and Figure 6-6. The I-FAN is composed by four main groups representative of the three Federated Internet Service Providers (F-ISP), each one hosting LAN of users for the multiparty videoconference service or shareable facilities, and a Network Connection Provider (NCP) emulating the Internet backbone, linking the F-ISP providing QoS management capabilities and part of the sub-systems allowing the multiparty videoconference service utilisation.
The F-ISP.1 has the role to provide to the I-FAN the satellite access hosting the satellite gateway. In the ESW network scenario, it hosts in particular all the proprietary equipment and sub-system to access the network, having the ESW Network Interface Unit (E-NIU) as bridge system interfacing, in a transparent mode for the user, the IP based network with ESW one.

The F-ISP.1 core is based on an EDge Router (ED-R) allowing the connection to the terrestrial network through the NCP and to the sub-systems to manage the SIP Proxy and Conference Server.

Two scenarios have been implemented in the demonstrator: the multicast user scenario shown in
Figure 6-5 and the unicast one shown in Figure 6-6. The difference in the two cases consists in the presence of the MCU in the unicast layout allowing the logical interfacing and management of unicast users in an IP Multicast based system. This difference is visible comparing the scenarios in Figure 6-5 where in the F-ISP.2 and 3 the MCU is presented, and Figure 6-6 where this entity is not presented. During the performance test and validation regime execution, also a combined scenario has been configured.

Concluding the description of the implemented I-FAN, the NCP represent the core system to manage the QoS capabilities and hosting the RP system to allow the conferencing service execution in an IP Multicast environment. The router typologies are presented including the CoRe Router (CR-R) representing the access point to the global network and the Border Routers (BD-R) representing the interface routers towards the sub-network (ISP or other).

6.3 Test Plan and Result Analysis

A detailed test plan has been developed to validate all technologies proposed and measure the performance of the designed conference system. The test mainly consists of three sub-tests:

- Functionality test
- Objective test

The function test was carried out within the test procedures of objective test. It included tests on signalling functionalities, MCU functionalities, Interworking functionalities, scalability, QoS support functionalities, media control functionalities and so on. The objective test was designed based on the functionality test and will be presented in the following sections.

6.3.1 Objective Test Parameters

The objective measurements of the demonstrator mainly focused on network level, where a set of parameters needs be measured and calculated collaborating with the work of the IP Performance Metrics (IPPM) Working Group in IETF. IPPM one-way parameter metrics have been introduced in previous sections. These parameters are:

- One-way delay: the time needed for packets travelling from a source host to a destination host.
- End-to-end delay: the mean of one-way delay.
- One-way delay variation (jitter): the difference in the One-way delay of selected packets.
- End-to-end delay variation: the mean of one-way delay variation.
Chapter 6. Measurement and Evaluation of Conference over Satellite

- Round-trip delay: the time needed for packets travelling from a source host to a destination host and then immediately being sent back to the source host.

- One-way packet loss: the number of packets lost from a source host to a destination host during a measurement interval time.

- End-to-end packet loss rate: the ratio of one-way lost packet to the totally transmitted packet.

The measurement is disaggregated into sub-measurement in terms of different network segments. These segments are:

- From unicast end user to local MCU,
- From MCU to MCU,
- From local MCU to end user,
- From end-user to end-user.

This disaggregating measurement is helpful to understand the performance of different networks involved in the whole IP conferencing service system over GEO satellite. It can provide sufficient information to locate potential network problems if any happened during the evaluation. For instance, one-way parameters measured from unicast end user to its local MCU can reflect the performance of the local ISP network and how it impacts on the end-user to end-user performance.

6.3.1.1 One-way Delay

The definition of one-way delay of a packet is the difference between the time when the source sends out the first bit of the packet and the time when the destination receives the last bit of the packet (whenever a time, i.e., a moment in history, is mentioned in this document, it is understood to be measured in seconds and fractions).

The format of the one-way delay metric of a sampled packet stream is:

- Metric name <ICEBERGS one-way delay metric – type – beginning time/date/duration>

Measurement parameters:

- The source IP address
- Destination IP address
- Delay time
- Packet length
Packet type (audio, video and data)

The motivations to measure one-way delay are:

i) It is needed to rate a VoIP system.

ii) It has to be used to calculate the one-way delay variation in real-time applications.

iii) It has to be used to calculate the end-to-end delay.

iv) The minimum value of this metric provides an indication of the delay due only to propagation and transmission delay [51].

v) Values of this metric above the minimum provide an indication of the congestion present in the path [51].

For ICEBERGS demonstrator, the one-way delay will be measured in segmented networks:

- Unicast end user to local MCU delay,
- MCU to MCU delay,
- Local MCU to end user delay,
- End-user-to-end-user delay.

Different one-way delays are needed to be studied to clarify the effect of each of the segments of the network on the performance of the conferencing over satellite systems. For instance, if all of the four delays are known, the delay effects caused by processing packets on the two MCUs in the link can be calculated.

Figure 6-7: One-way delay measurement scenarios for ICEBERGS
Here, the time required for a packet to travel through the network is measured by comparing the time reported by a clock at one end of the packet’s path, corresponding to when the packet first entered the network, with the time reported by a clock at the other end of the path, corresponding to when the packet finished traversing the network [45]. Synchronization of the source and the destination were guaranteed by using GPS.

Network measurement tools, such as Ethereal and Analyzer, are running on selected conference end users and conference MCUs to monitor measured packets. All of these nodes should be synchronized, which means the offset of the clocks of one measured host pair should be zero or very close to zero. The relative skews and drifts [45] of the two host clocks should be minimal. The output clocks should get their time notion from an external source, e.g. a GPS device.

An alternative can be achieved based on the round-trip delay measurements. The idea is to divide the round-trip delay by 2 to approximate the one-way delay. The advantage is that this method does not need two hosts have accurately the same time. However, effects of the skew and drift of the two clocks cannot be avoided. The disadvantage is that the links of the round trip may be asymmetrical, i.e. the link from host 1 to host 2 may not be the same as the link from host 2 back to host 1. Thus, simply using half of the round-trip delay as one-way delay will introduce inaccuracy.

In the measurement, a host should keep on sending fixed length video, audio and data packets to another host in a sampling period. To have an accurate metric of the one-way delay, some issues have to be noted:

i) To collect samples, periodic sampling [45] method can be implemented considering its simplification. However, to minimize the effects of the network periodic behaviour, the sampling should be made separately at different intervals of time.

ii) The packets transmitted during measurement should be the same type. This means all of the packets must have the same source and destination, the same UDP/TCP port number and the same length. To minimize the effect of packet fragmentation [45], the packet lengths in the measurement should all be the maximum

The measurement procedures are listed below:

i) Arrange that the source and the destination are synchronized using external GPS systems; that is, that they have clocks that are very closely synchronized with each other and each fairly close to the actual time.

ii) At the source host, form a 1518 bytes long video or audio or data test packet with the selected destination IP address and TCP/UDP port. Any ‘padding’ portion of the packet needed only to make the test packet a given size should be filled with randomised bits to
avoid a situation in which the measured delay is lower than it would otherwise be due to compression techniques along the path. Video, audio and data packets can be sent simultaneously with respect to the real conference applications.

iii) At the destination host, arrange to receive the packet.

iv) At the source host, place a timestamp in the prepared test packet, and send it towards destination.

v) Assign a timestamp as soon as possible upon the receipt of the packet at the destination. By subtracting the two timestamps, an estimate of one-way delay can be computed. Error analysis of a given implementation of the method must take into account the closeness of synchronization between source and destination. If the delay between source's timestamp and the actual sending of the packet is known, then the estimate could be adjusted by subtracting this amount; uncertainty in this value must be taken into account in error analysis. Similarly, if the delay between the actual reception of the packet and destination’s timestamp is known, then the estimate could be adjusted by subtracting this amount; uncertainty in this value must be taken into account in error analysis [45].

vi) If the packet does not arrive at the destination before the sampling period and a threshold time is finished, it will be treated as a lost packet. The destination needs to keep monitoring the network until the threshold time after the measurement to ensure the last packet can be measured if it arrives before the threshold time. In the threshold time after the measurement, only packets with source timestamps falling into the measurement period can be recorded and assigned destination timestamps.

6.3.1.2 End-to-end Delay

The mean one-way delay was chosen to be end-to-end delay. The types of end-to-end delay correspond to the types of the one-way delay.

The audio end-to-end delay QoS classes refer to [63]. The Telecommunications and Internet Protocol Harmonization over Networks (TIPHON) recommended end-to-end delays are shown in Table 6-1:

<table>
<thead>
<tr>
<th></th>
<th>3 (WIDEBAND)</th>
<th>2 (NARROWBAND)</th>
<th>1 (BEST EFFORT)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2H (HIGH)</td>
<td>2M (MEDIUM)</td>
</tr>
<tr>
<td>End-to-end delay</td>
<td>&lt;100 ms</td>
<td>&lt;100 ms</td>
<td>&lt;150 ms</td>
</tr>
</tbody>
</table>

NOTE: The delay for best effort class is a target value

Table 6-1: End-to-end delay for TIPHON Systems

However, there are no any mature references for the video end-to-end performance.
6.3.1.3 One-way Delay Variation (Jitter)

The one-way delay variation of a pair of packets within a stream of packets is defined as the difference of the one-way delays of a selected pair of packets in the stream going from measurement point MP1 to measurement point MP2.

The format of the one-way delay variation metric of a sampled packets stream is:

- Metric name <ICEBERGS one-way delay variation metric - type - beginning time/date/duration >

Measurement parameters:

- The source IP address
- Destination IP address
- Packet chosen function (defining unambiguously the two packets from the stream selected for the metric)
- Delay time 1
- Delay time 2
- Delay variation
- Packet length
- Packet type (audio, video and data)

The motivation to measure one-way delay variation is:

i) Very important factor for the audio and video streams and sizing of play-out buffers for applications requiring the regular delivery of packets [55].

ii) To determine the dynamics of queues within a network (or router) where the changes in delay variation can be linked to changes in the queue length process at a given link or a combination of links [55].

iii) Robust to synchronization of the clocks of the two hosts with respect to time difference and skew.

For ICEBERGS, the types of one-way delay variation are correspondent to those of the one-way delay, already mentioned:

- Unicast end user to local MCU delay variation,
- MCU to MCU delay variation,
- Local MCU to end user delay variation,
• End user to end user delay variation.

The one-way delay variation measurement can be done during the measurement of one-way delay. Upon each measured packet arrival in step v) in the one-way delay measurement, the destination records its arrival wire-time and calculates its one-way delay time. If the packet fits into the packet chosen function criterion, it was selected and the difference of one-way delays between it and the previous selected packet is the one-way delay variation of this pair of packets. Then a triple \(< T(i), T(j), D(j) - D(i) >\) are collected for this metric, where \(T(i)\) and \(T(j)\) are the \(i^{th}\) and \(j^{th}\) packets that match the packet chosen function criterion in that measurement and \(D(i)\) and \(D(j)\) denote the one-way delay of \(T(i)\) and \(T(j)\) respectively.

The function that is used to sample packets for this demonstration specifies packets whose sequence numbers are the integers bigger than the first received test packet and smaller but the last received test packet. For video and audio packets, the sequence number is the RTP packet sequence number. For data packet, the source host has to assign a sequence number for every packet it sends out.

**6.3.1.4 End-to-end Delay Variation**

The mean one-way delay variation was chosen to be the end-to-end delay variation. The types of end-to-end delay variation correspond to the types of the one-way delay defined previously.

**6.3.1.5 Round-trip Delay**

The round-trip delay is defined as the sum of the times needed for a test packet travel from the source to the destination and from the destination back to the source.

The format of the round-trip delay metric of a sampled packet stream is:

- Metric name \(<ICEBERGS\ \text{round-trip delay metric -- type -- beginning time/date/duration}>\)

Measurement parameters:

- The source IP address
- Destination IP address
- Delay time
- Packet length
- Packet type (audio, video and data)

Although the round-trip delay measurement introduces inaccuracy, there are motivations to measure round-trip delay in this demonstration:
i) Round-trip delay provides an alternative to measure one-way delay because it is easy to implement. Unlike the one-way delay measurement, it often does not need install any measurement software at the destination. Moreover, the high requirement of the synchronization of the two clocks at the source and destination are not necessary.

ii) In some stages (including signalling and establishing multicast trees via satellite), the key procedures combine the link from the source to the destination and the link from the destination to the source. Round-trip delays are needed to evaluate these performances in these scenarios.

The most important round-trip delay is the delay between two satellite enabled RPs in different satellite spot beams because the procedures mentioned in the second motivation mainly happen between them. The MCU to MCU round-trip delay is the one most close to the RP to RP round-trip delay. If necessary, it can be used to approximate the RP to RP round-trip delay.

The round-trip delay can be measured in a similar way as one-way delay. The first step in the one-way delay is skipped and the measurement starts as:

i) See 2nd and 3rd step in the methodology for one-way delay

ii) At the source host, send the prepared packet towards destination and record the packet leaving time and sequence number.

iii) Upon the receipt of the test packet at the destination, a same length acknowledgement is created with the sequence number and sent back to the source.

iv) Software installed on the source inspects incoming network traffic to determine if a successful reply is received before a threshold. The round-trip delay is calculated by subtracting test packet leaving time from the reply arrival time. The unit is second.

v) If no reply received before the threshold, the round-trip delay is ‘undefined’.

vi) The same procedure is repeated during a measurement interval time, which can be double of the interval time for one-way delay measurement.

The sampling method is the same with one-way delay measurement.

6.3.1.6 One-way packet loss

If a test packet does not arrive at its destination in a threshold, it is defined lost. The format of the one-way packet loss metric of a sampled packet stream is:

- Metric name <ICEBERGS one-way packet loss metric - type - beginning time/date/duration >

Measurement parameters:
• The source IP address
• Destination IP address
• Lost or not (Boolean). "0" means packet is transmitted successfully. "1" means packet lost.
• Packet length
• Packet type (audio, video and data)

The motivations to measure one-way loss are:

i) Excessive packet loss may make it difficult to support certain real-time applications

ii) The larger the value of packet loss, the more difficult it is for transport-layer protocols to sustain high bandwidths

iii) The sensitivity of real-time applications and of transport-layer protocols to loss become especially important when very large delay-bandwidth products must be supported.

For ICEBERGS, the one-way loss is measured on the same kind of link for the one-way delay measurement.

One-way packet loss conditions on different links are needed to study the effect of different parts in the conferencing over satellite system on the network performance. They provide more information to improve the system.

The procedures to measure the one-way packet loss are very similar with those for one-way delay. They are described as:

i) See the step i, ii, iii and iv in the methodology for one-way delay.

ii) If the packet arrives within a threshold time, the one-way packet-loss is taken to be zero. The threshold time is a parameter that depends on one-way delay. Considering the long delay caused by satellite and the requirement of the real-time application, the threshold cannot be bigger than 1.5 times of the one-satellite-hop travel time. For different measurement links, the threshold will be different.

iii) If the packet fails to arrive within the threshold time, the one-way packet-loss is taken to be one

6.3.1.7 End-to-end packet loss rate

The ratio of the lost one-way packet to the very transmitted packet is a very important statistical parameter to judge the performance of a conferencing over satellite system. Both the lost packet number and the very transmitted packet number can be found from the one-way packet loss
6.3.2 Signalling Test Output Analysis

In this section, the details of test cases will be presented for signalling function test as well as the results analysis.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Result</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Users Join a Conference: Unicast User</td>
<td>Mean call setup delay = 2002.699ms.</td>
<td>This result fulfils the requirements for national and international calls stated in ETSI TS 101 511 and ETSI TS 102 024-9.</td>
</tr>
<tr>
<td></td>
<td>Maximum call setup delay = 2144ms.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minimum call setup delay = 1974ms.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean call setup delay = 1996.561ms</td>
<td>This result fulfils the requirements for international and national long-distance calls stated in ETSI TS 101 511 and ETSI TS 102 024-9.</td>
</tr>
<tr>
<td></td>
<td>Maximum call setup delay = 2114ms</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minimum call setup delay = 1964ms</td>
<td></td>
</tr>
</tbody>
</table>

Table 6-2: Call setup delay for unicast and multicast users

Table 6-2 shows clearly that the call setup time for both the case of a unicast user joining a conference and a multicast one joining a conference are around 2 seconds, which is fulfils the requirement for the international calls stated in ETSI TS 101 511 and ETSI TS 102 024-9. This is the strong evidence that the designed signalling system for the conferencing over satellite system is very successful.

More information is provided by the test. Table 6-3 shows some statistic parameters for two basic SIP functions: registration and deregistration. Due to the simple processes of these two functions, actually no need to query the location server in the processes. Their mean delays are much less than the mean delay of the invitation function. It is important that the location server and the SIP proxy should be located together or very closely. Separating the SIP proxy and the location server by satellite links will dramatically increase the invitation delay because of the location queries between proxies and location servers. Also due to the similarity of the processes of registration and deregistration, their mean delay are very close to each other, i.e. for unicast users, the difference between the mean registration delay and the mean of deregistration delay is 1889.389 ms – 1889.307 ms = 0.082 ms, which could be ignored compare the mean delay value.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Registration delay</th>
<th>Deregistration delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Users Join a Conference: Unicast User</td>
<td>Mean: 1895.151 ms</td>
<td>Mean: 1928.576 ms</td>
</tr>
<tr>
<td>(with a MCU).</td>
<td>Maximum: 2025 ms</td>
<td>Maximum: 2144 ms</td>
</tr>
<tr>
<td></td>
<td>Minimum: 1884 ms</td>
<td>Minimum: 1894 ms</td>
</tr>
</tbody>
</table>
End Users Join a Conference: Multicast User (without MCU).

<table>
<thead>
<tr>
<th>Mean: 1889.389 ms</th>
<th>Mean: 1889.307 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum: 2024 ms</td>
<td>Maximum: 2015 ms</td>
</tr>
<tr>
<td>Minimum: 1854 ms</td>
<td>Minimum: 1854 ms</td>
</tr>
</tbody>
</table>

Table 6-3: SIP registration and deregistration delay for unicast and multicast users

Furthermore, these data can be used to calculate the MCU delays for SIP invitation, registration and deregistration. The formula is:

\[ \text{MCU delay} = \text{Mean delay with MCU} - \text{Mean delay without MCU} \]

Thus the MCU delay for the SIP invitation is:

\[ 2002.699\text{ms} - 1996.561\text{ms} = 6.138 \text{ms} \]

The MCU delay for the SIP registration is:

\[ 1895.151 \text{ms} - 1889.389 \text{ms} = 5.762\text{ms} \]

The MCU delay for the SIP deregistration is:

\[ 1928.576 \text{ms} - 1889.307 \text{ms} = 39.269\text{ms} \]

It can be seen that both invitation and registration process suffer only around 6ms MCU delays. This delay is due to that the MCU should transform the unicast requests from end users to multicast request to the SIP entities and vice versa. The delay of the deregistration is longer, around 39ms. This is because the MCU’s programme does not do any optimisation for it regarding that the delay of deregistration is not as important as the invitation and the registration.

Therefore, the measurement results show that the designed signalling system for the conferencing over satellite system can provide very good QoS in terms of the call setup delay, which is around 2 seconds. The delays of registration and deregistration is less than the invitation delay because no over satellite transmission. The MCU delay is only around 6ms for these three services, which can be neglected comparing to the 2 seconds invitation delay.

6.3.3 Performance Analysis: Network Level Parameters

This section will present the test results measured in several scenarios as well as the corresponding performance analysis based on the measured parameters defined in the Objective Test Parameters section.

6.3.3.1 Conference Test Scenario for Multicast

This test scenario was designed to verify the QoS improvement of QUASIMODO [16] designed for inter-domain multicast in the IP conferencing over satellite system. It will show how the performance of a conference system deploying QUASIMODO can be improved with typical
Internet background traffic.

The scenario is shown in Figure 6-8. In this scenario, all four domains are involved. Six end users will join one conference. Two edge routers (ED-R), three border routers (BD-R), which are also DEsignated Routers (DE-R), and one core router should be configured and deployed in this scenario. All of them should support QUASIMODO. One RP is needed to support PIM-SM that is plotted as RV-Point in Figure 6-8. Three traffic generators were deployed in the local networks with the end users to provide background traffic.

The background traffic was generated using developed tools, which can generate both multicast and unicast traffic. The volume ratio of unicast traffic to multicast traffic was around 1:7. The total background traffic occupied from 25% up to 75% resources of routers so that the effect of QUASIMODO can be clearly measured. The background traffic caused congestion on one edge router.

Test Procedures are described as below:

i) The measurement tools are installed in all the six end users.

ii) Generate the background traffic using the traffic generators.

iii) Launch the measurement tools and all six users join a conference.

iv) Monitor the PIM-SM messages when users join in the group.

v) Let one unicast user send fixed-length test cases to the group using customer software and monitor all of the receivers.
vi) Record the one-way delay, one-way jitter and number of packet lost for video and audio streams. These measurements should last 30 minutes. The measurement methods were described in section 6.3.1. Note that only end user to end user parameters are measured in this case without any satellite link. For each parameter, the factors involved in its metric have to be recorded as well.

vii) Repeat step v and step vi for 5 times and keep all of the measurement results.

viii) Apply step v to the multicast user and repeat step vi and vii experiments.

ix) Setup the network to support QUASIMODO.

x) Repeat step iii to viii and keep all of the results.

xi) Compare the two groups results (the results of these two group experiments), one for normal PIM-SM and the other for QUASIMODO, to see the advance of QUASIMODO.

Even though the end to end traffic traverse several network segments (users->MCU, MCU-MCU in multicast, MCU->users), QUASIMODO improvement is performed in the multicast traffic. So, only the network performance parameters of the multicast segment were considered.

When the traffic load in the demonstrator occupied 25% network resources, it can be seen in the following picture the impact of the QoS solution regarding the delay. In the first part of the graph, QUASIMODO is disabled. It is enabled when packet sequence is 17000. It can be seen clearly that QUASIMODO had significant improvement on end-to-end delay.

Figure 6-9: One-way delay for pure terrestrial test scenario

Figure 6-9 shows the packet delay result plotting for one measurement interval. The measurement results are very similar in other test intervals. In Figure 6-9, it is very clearly that the delay is
Chapter 6. Measurement and Evaluation of Conference over Satellite

Chapter 6. Measurement and Evaluation of Conference over Satellite

sharply reduced from around 430ms to 275ms in the half way of the measurement. It shows that when the QUASIMODO functions turned on, the delay performance was improved for around 150ms. It gives a solidly confirmation that the QUASIMODO does have the ability to statistically guarantee the QoS for the real-time multicast traffic over the DiffServ network. The delay improvement resulted by the QUASIMODO in the measurement is 255ms. The main reason of such a significant value is that in the demonstrator, the QUASIMODO used as many network resources as it needed while the background traffics were completely ignored. Therefore, no queuing delay was measured for the test traffic and the end-to-end delay became the one-satellite-hop delay.

Jitter, the variance of packet delay, is another more critical parameter for a video application than the delay because it directly affects the video quality. To calculate the jitter, of the \( i^{th} \) packet, the equation (9.1) was deployed.

\[
J(i) = J(i-1) + \frac{|D(i) - D(i-1)| - J(i-1)|/16}{16}
\]

(9.1)

Where \( J(i) \) is the jitter of the \( i^{th} \) packet, \( J(i-1) \) is the previous packet. \( D(i) \) is the delay of the \( i^{th} \) packet and \( D(i-1) \) is the delay of the previous one. Figure 6-10 shows the plotting of the calculated result.

![Jitter: BE VS. QUASIMODO](image)

Figure 6-10: One-way jitter for pure terrestrial test scenario

In Figure 6-10, the blue line shows the packet jitter measured for the best effort network while the red curve represent for the QUASIMODO supported network. Clearly, the QUASIMODO improves the QoS of the traffic by reduced the packet jitter. Moreover, Figure 6-10 shows that the jitter curve for best-effort has a wave contour. In another words, it has peaks and troughs.
following each other. This reflects that the tested video traffic was affected by the background traffic, which has the burstiness characteristic of the typical Internet traffic. In the view of a router in the core network, when a lot of packets flood into its queue in a traffic burst, it cannot give any special support to the video flow and that causes variance of delay between each of the video packet. When the QUASIMODO functions are switched on, the contour of the jitter curve (the red one) is more constant. It means that the routers in the path accept the video traffic only when they can maintain enough resource to allocate to it and fit the video flow’s bandwidth requirement. Therefore, the background traffic causes much less impact to the tested video flow. This is the expected result to prove that the QUASIMODO does work well to support the multicast QoS using admission control algorithm over DiffServ to maintain enough local resources for the QoS required real-time multimedia flows.

In another scenario where the traffic carried over the demonstrator occupied 75% of the network resources, the delay and jitter were also improved by QUASIMODO. However, the packet loss ratio benefited from the algorithm as well. This is mainly because the algorithm reserved network resources along the multicast tree branches to enforce the QoS delivered from sources to receivers [16].

The test execution has shown a significant improvement when QUASIMODO is applied. The packet loss ratio goes from 16% in the best effort case to 0.27% when QUASIMODO is applied.

6.3.3.2 Scenario for One Terrestrial Unicast User and One Satellite Unicast User

In this test scenario, the parameters defined in section 6.3.1 will be measured and recorded for later statistical analysis. It will verify and evaluate the simplest two-member-conference via satellite scenario. The main objective of this scenario is to examine the impact of the satellite link to the performance of the conference system.

Both the real satellite link and the ESW emulator are used in the scenario. The satellite provided really satellite link traffic pattern and the emulator emulated the onboard switching functions of ESW. Two PCs, one running the Windows Messenger and the other one running the Mbone tool, worked as the two end users. Two designated routers, two MCUs, two traffic generators, signalling entities, one PIM-SM RP, two hubs and two MCUs were deployed in the scenario as shown in Figure 6-11.
Test procedures are as described below:

i) The measurement tools should be installed on both end users and both MCUs.

ii) Generate the background traffic using the new traffic generator.

iii) Launch the measurement tool and end users join the conference.

iv) Monitor the PIM-SM messages when the MCUs join in the same group.

v) Monitor the both video and audio traffic while one unicast user talks and another user listen.

vi) Use measurement tools to record the one-way delay, one-way jitter and number of packet lost for video and audio streams. The measurement lasts 30 minutes. The measurement methods were described in section 6.3.1. For each parameter, the parameters involved in its metric have to be recorded as well. The arrival time of each incoming packet on both end should be also recorded, which will be used for traffic analysis later. The satellite bandwidth consumed should be also recorded.

vii) Repeat step vi for 5 times and keep all of the measurement results.

Table 6-4 shows measurement result in terms of one-way parameters. For the one-way delay and jitter and the number of packet lost, all parameters defined in the corresponding metrics in section 6.3.1 should be stored. Another important parameter is the MCU delay shown in Table 6-5, which can be calculated using the method described in section 6.3.2.

<table>
<thead>
<tr>
<th>Net. Segment</th>
<th>One-way delay</th>
<th>One-way jitter</th>
<th>Packet lost Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>UniTer 1 – MCU1</td>
<td>&lt;5ms</td>
<td>&lt;5ms</td>
<td>0</td>
</tr>
<tr>
<td>MCU1-MCU2</td>
<td>275.85 ms</td>
<td>4.15 ms</td>
<td>0.27%</td>
</tr>
<tr>
<td>MCU2 – UniTer 2</td>
<td>&lt;5ms</td>
<td>&lt;5ms</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 6-4: Measurement result for unicast end user to unicast end user

Note: MCU 1 is the MCU in the ISP-1 domain and MCU 2 is the one in satellite domain.

<table>
<thead>
<tr>
<th>MCUs performance</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU1 delay:</td>
<td>25ms</td>
</tr>
<tr>
<td>MCU2 delay:</td>
<td>25ms</td>
</tr>
</tbody>
</table>

Table 6-5: Delay of MCUs in unicast end user to unicast end user scenario

MCU delay for media data is not measurable in the demonstrator directly. However, with the method described in section 6.3.2, the value of MCU delay could be calculated that is around 25ms. This value is much smaller than the one-way delay and, hence, the MCU will not be the key issue to the performance of the conferencing over satellite system. Therefore, it can be found that the unicast end user to unicast end user one-way delay is 50ms (two MCU delays) in additional to the MCU to MCU one-way delay in the end-to-end link budget. Therefore, the measured results from different segment of the network can be used to calculate the end-to-end Budget.

- Delay:
  \[ D_{tot} \equiv 275.8 + 25 + 25 = 324.8 \text{ ms} \]
- Packet loss ratio:
  \[ P_{tot} \equiv 0.27\% \]
- Delay variation:
  \[ DV_{tot} < 9.1 \text{ ms} \]

6.3.3.3 A Terrestrial Multicast User Joins the Previous Scenario

The objective of this scenario is to examine performance of the procedures designed for a multicast terrestrial end user joining in a conference, i.e. the delay. It will also test if the multicast user can coexist with unicast users in a conference. This will reflect the scalability and compatibility of the conferencing over satellite system. The two main objectives are how long it will take for a new multicast user to join a conference, i.e. from its dialling in to receiving first data packet, and if it affects the existing users in terms of delay, jitter and packet loss. The reason of using two different terrestrial ISPs is to make it easy to enlarge to more complicated scenarios later.

A multicast enabled terrestrial end user should join the conference established in previous scenario. This new participant and the previous terrestrial user belong to different ISPs. All of the equipment used in the previous scenario will be used again in addition to one PC running MBone.
tools. Besides, one border router, one traffic generator and one designated router were additionally deployed. Figure 6-12 shows a connection example.

![Figure 6-12: Three-participant-conference scenario](image)

Test Procedures are described below:

i) Measurement tools were installed on all end users.

ii) Generate the background traffic using the traffic generators.

iii) Launch the measurement tools and the new terrestrial multicast user joins the conference established in previous scenario.

iv) Monitor the PIM-SM messages when the new terrestrial multicast user joins in the group.

v) Monitor the both video and audio traffic while one unicast user talks and the other users listen.

vi) Use measurement tools to record the one-way delay, one-way jitter and number of packet lost for video and audio streams. The measurement lasts 30 minutes. The measurement methods were described in section 6.3.1. For each parameter, the parameters involved in its metric have to be recorded as well. The arrival time of each incoming packet on both end should be also recorded, which will be used for traffic analysis later. The satellite bandwidth consumed should be also recorded.

vii) Repeat step vi for 5 times and keep all of the measurement results.

The outputs of one-way parameters in this scenario are almost the same with the previous one expect that a new measurement node, the multicast user, should be considered with the same parameters. In addition, the delay of a multicast terrestrial end user joining in a conference was
calculated based on the packet arrival time, i.e. the difference of the arrival time of the first JOIN message from the terrestrial multicast user to the multicast group and the arrival time of the first data packet received from the group.

Table 6-6 shows the measurement result of one-way parameters for this scenario and Table 6-7 shows the MCU performance.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>One-way delay</th>
<th>One-way jitter</th>
<th>Packet lost Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>UniTer 1</td>
<td>&lt;5 ms</td>
<td>&lt;5 ms</td>
<td>0</td>
</tr>
<tr>
<td>MCU1-MCU2</td>
<td>275.85 ms</td>
<td>4.15 ms</td>
<td>&lt;0.27%</td>
</tr>
<tr>
<td>MCU1-MulTer1</td>
<td>&lt;5 ms</td>
<td>&lt;5 ms</td>
<td>0</td>
</tr>
<tr>
<td>MCU2-MulTer1</td>
<td>275.85 ms</td>
<td>4.15 ms</td>
<td>&lt;0.27%</td>
</tr>
<tr>
<td>MCU2 - UniTer 2</td>
<td>&lt;5 ms</td>
<td>&lt;5 ms</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6-6: Measurement result for three-participant-conference scenario

Note: MCU1, UniTer1 are the equipment in the ISP-1 domain. MCU2, UniTer2 are equipment in satellite domain. MulTer1 is in ISP-2 domain. "MulTer" stands for multicast end user.

<table>
<thead>
<tr>
<th></th>
<th>MCU1 delay:</th>
<th>MCU2 delay:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>25 ms</td>
<td>25 ms</td>
</tr>
</tbody>
</table>

Table 6-7: MCU delay in three-participant-conference scenario

It was also found that the delay of a multicast terrestrial end user joining in a conference is 3480 ms.

When increasing the traffic load over the demonstrator, it was found that the traffic load does not have impact on the conference QoS. This is because the QUASIMODO protocol uses a Call Admission Control mechanism that limits the number of users simultaneously present in the QoS portion of the bandwidth.

The end-to-end budget for the communication between UniTer2 and MulTer1 is:

- End-to-end delay: \( D_{\text{tot}} < 275.8 + 25 = 300.8 \text{ ms} \)
- End-to-end packet loss ratio: \( P_{\text{tot}} \equiv 0.27\% \)
- End-to-end Delay variation: \( DV_{\text{tot}} < 9.1 \text{ ms} \)

The end-to-end budget for the communication between UniTer1 and UniTer2, which is the worst case in this scenario, is:

- \( D_{\text{tot}} < 275.6 + 25 + 25 = 325.6 \text{ ms} \)
- \( P_{\text{tot}} \equiv 0.27\% \)
- \( DV_{\text{tot}} < 9.1 \text{ ms} \)
6.3.3.4 One Satellite Multicast User Joins the Previous Scenario

This scenario presents a more complicated conference test with the forth participant joining. The main objectives are to examine the effects of a satellite multicast end user joining a conference and of a new source joining a multicast group. The interested point was what effects this new user can bring to the conference. It is considered the worst join condition in this system because the new source has to be a receiver first when he applies to join the group and then convert to a source by sending its first data packet. There will be two satellite link setting up procedures happening in this case that may cause significant delay.

A multicast enabled satellite end user would join the conference established in previous scenario. This new participant would be a source in the multicast group. All of the equipment used in the previous scenario will be the same in addition to one PC running Mbone tools as the new multicast end user. Figure 6-13 shows a connection example.

![Figure 6-13: Four-participant-conference test scenario](image)

Test procedures are shown as following:

i) Repeat the step i and ii in the measurement procedure in the previous scenario.

ii) Launch the measurement tools and the new multicast satellite user joins the conference established in previous scenario and all the other users stop talking. The new user begins to talk immediately after it joins the conference successfully and let he continue for 30 minutes.

iii) Monitor the PIM-SM messages, Video and Audio packets on all four users and all MCUs simultaneously. The time of the new user sending the first JOIN request and the time of receipt of the first date packet on any other user should be recorded to calculate the delay.
of a new source joining in the multicast group.

iv) Use the measurement tools to record the one-way delay, one-way jitter and number of packet lost for video and audio streams. The measurement lasts 30 minutes. The measurement methods were described in section 6.3.1. For each parameter, the parameters involved in its metric have to be recorded as well. The arrival time of each incoming packet on both end should be also recorded, which will be used for traffic analysis later. The satellite bandwidth consumed should be also recorded.

v) Repeat step iv for 5 times and keep all of the measurement results.

The output in this case should be similar with previous scenario except of a new measurement node, the satellite multicast user, and the delay of a new source joining the multicast group. To calculate this delay, the delays on each of the three users have to be found out, which is the difference of the time of the JOIN message from the satellite multicast user and the receipt time of the first data packet arrival at each receiver. The source join-delay is the mean of the delays measured on all three users.

Table 6-8 shows the one-way parameter measurement results for this scenario and Table 6-9 shows the MCU performance.

<table>
<thead>
<tr>
<th>Parameter:</th>
<th>Net. Segment</th>
<th>One-way delay</th>
<th>One-way jitter</th>
<th>Packet lost Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>UniTer 1 - MCU1</td>
<td>&lt;5ms</td>
<td>&lt;5ms</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>MCU1-MCU3</td>
<td>275.85ms</td>
<td>4.15ms</td>
<td>&lt;0.27%</td>
<td></td>
</tr>
<tr>
<td>MCU1-MulTer2</td>
<td>5ms</td>
<td>5ms</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>MCU3-MulTer3</td>
<td>5ms</td>
<td>5ms</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>MCU1-MulTer3</td>
<td>275.85ms</td>
<td>4.15ms</td>
<td>&lt;0.27%</td>
<td></td>
</tr>
<tr>
<td>MCU3-MulTer2</td>
<td>275.85ms</td>
<td>4.15ms</td>
<td>&lt;0.27%</td>
<td></td>
</tr>
<tr>
<td>MCU3-UniTer3</td>
<td>&lt;5ms</td>
<td>&lt;5ms</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table 6-8: Measurement result for four-participant-conference scenario

Note: MCU1 and UniTer1 are the equipment in the ISP-1 domain. MCU3, MulTer3 and UniTer3 are equipment in satellite domain. MulTer2 is in ISP-2 domain. "MulTer" stands for multicast end user.

<table>
<thead>
<tr>
<th>MCU1 delay:</th>
<th>25ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU3 delay:</td>
<td>25ms</td>
</tr>
</tbody>
</table>

Table 6-9: MCU delay in four-participant-conference scenario

It was found that the delay of a new source joining the conference is 8020ms.

The end-to-end budget for the communication between UniTer1 and UniTer3, which is the worst case in this scenario, is:

- End-to-end delay: \( D_{tot} < 275.8 + 25 + 25 = 325.8 \text{ ms} \)
- End-to-end packet loss ratio: \( P_{tot} \equiv 0.27\% \)
Chapter 6. Measurement and Evaluation of Conference over Satellite

- End-to-end Delay variation: $DV_{tot} < 9.1$ ms

6.4 Summary

This chapter introduced the IP traffic measurement technologies in the beginning and then, based on these technologies, it described a measurement regime designed and carried out over a conferencing over satellite system demonstrator. Measurement results show the success of the designed system in terms of both functionalities and network performance.

Traffic measurement is crucial to the traffic engineering function. It provides the means to have the insight of the network operation state and problem anticipation. It is useful for optimisation the network because it can provide the feedback data for the engineer to adaptively optimise network performance in response to events and stimuli originating within and outside the network. It is essential to determine the quality of network services and to evaluate the effectiveness of traffic engineering policies.

Researchers have developed a lot of measurement methodologies, e.g. using LOG files and capturing packet form the Internet using some software and hardware. They can be divided into two main groups: passive approach and active approach. Passive measurement approach implies to use devices, to monitor the traffic when it passes by. It will not generate any extra traffic in the network. This is in contrast to active measurement, in which specific packets are introduced into the network, and these packets are timed as they travel through the network being measured. Passive measurement and active measurement are complementary rather than against each other. By comparing and contrasting the active and passive measurements, the co-validity of the different measurements can be verified and much more detailed information on carefully specified/scheduled phenomena is made available.

After the study of the measurement methodology, this chapter presented a set of one-to-one performance parameters and their metrics recommended by IETF in IPPM working group. These parameters are widely accepted and used by many researchers. In total, 6 parameter metrics were standardized in IPPM and four of them that have close relations with the measurement regime for the designed conferencing over satellite demonstrator were briefly introduced. They are one-way delay, packet delay variation (jitter), round-trip delay, one-way packet loss. Also the relevance to the measurement for a satellite conferencing system were considered and discussed.

With the background study of IP traffic measurement technologies and the special consideration of the conferencing over satellite system, a measurement plan has been established and successfully carried out on a conferencing over satellite system demonstrator within the ICEBERGS project, including definition of a set of performance parameters and their metrics and...
different test cases.

The demonstrator prototype implementation is based on two scenarios. One was using the actual satellite access network based on SESAT satellite to investigate the feasibility of the design in real satellite environment and collecting data with the effect of the real satellite. However, considering the transparent characteristics of the SESAT satellite, an ESW emulator was used in another scenario to emulate the ESW onboard switching functionality.

The test regime consisted of functionality tests and objective tests. The functionality tests included signalling functionalities, MCU functionalities, Interworking functionalities, scalability, QoS support functionalities, and so on. It was carried out within the test procedures of the objective test. A set of parameters was defined in the test plan to describe the network performance provided by the demonstrator as well as the corresponding metrics and measurement methodologies. The measurement was carried out in several different test scenarios to testify all the functionalities and the delays they introduced. The results showed that the designed conferencing over satellite system can provide less than 300ms end-to-end delay including 25ms MCU delay as well as around 0.27% end-to-end packet lost rate and less than 9.1ms end-to-end jitter.
7 Performance Parameters and Relative QoS Optimization Algorithms for Multiparty Multimedia Communications

The term "QoS" means objective quality of service in this chapter. The concepts of "performance" and "QoS" need to be clarified. They are the same thing but viewed from different aspects. The "performance" is used to describe how good or bad the network is working while "QoS" is the term to present how good or bad users are serviced. Therefore, "performance" is the view of the "QoS" from the network point and "QoS" is the service point of view of "performance". If the network performance is better, then the QoS that users are served will be better. That is one of the reason that the performance parameters and relative QoS optimization algorithm will be presented in the same chapter. Another reason is that the relative QoS optimization algorithm is proposed based on the multiparty multimedia communications performance parameters.

The term "communication" in this thesis has a very general scope. It does not only mean the "talk" by languages of human being but also includes all kinds of information exchange between human being and between computers driven by human being. For instance, IP conferencing is a kind of communication because human beings are talking with each other in this service. Online gaming is also a kind of communication because human beings are driving the computers to communicate with each other.

The measurement regime that was carried over the conferencing over satellite demonstrator provided positive results to prove the success of the designed system. However, the measurement results also lead to a more general consideration about the performance of multiparty multimedia communications. The consideration was raised by the very different performance observed at different end users in the conferencing system where a user who is one satellite hop away from the source has much longer delays than those who located in the same Ethernet with the source. This performance difference results in the various QoS that end users are served. Research has been carried out based on this consideration and it was found that real-time multiparty multimedia services over IP networks introduced a new QoS requirement. This new QoS requirement will be identified in this chapter and a set of new QoS parameters will be proposed that can be derived from the end-to-end parameters to describe this new requirement. The potential use of these proposed parameters are also discussed. Then, an adaptive QoS optimisation algorithm for
Chapter 7. Performance Parameters and Relative QoS for MMC

Multiparty multimedia communications will be proposed based on the traffic measurements corresponding to the new QoS requirement and parameters. Finally, a simulation and its results will be presented and analyzed to evaluate the proposed algorithm.

7.1 Performance Parameters for Multiparty Multimedia Communications

All IPPM performance parameters are defined for one-to-one connections. Further attention should be put on the multiparty applications that use multicast routing protocols, e.g. the IP conferencing services and online gaming services. The basic consideration is that in the multiparty communication, a group of people are involved in the action rather than two. One may say that no matter how many people join the communication, the connections can still be treated as a set of one-to-one connections. However, the performance a multiparty communication service can not be described by a set of one-to-one measurement metrics both because of the difficulty for understanding and the lack of convenience. For instance, an engineer might not describe the connections of a multiparty online conference in terms of one-way delay for user A and B, B and C, and C and A because people might be confused. And if he uses the one-to-one parameters with the worst and the best value to give users an idea of the QoS range of the service they are providing, it is not clear enough and might not be accurate in a big multiparty communication service. The suggestion is to use a more sophisticated way after mathematic deriving, i.e. mean, variation and etc. the new parameter and the corresponding metrics will be more efficient and accurate to express the performance situation among a group of users.

From the QoS point of view, the multiparty communication services not only require the absolute QoS support but also the relative QoS. The relative QoS means the difference between absolute QoS of all users. Directly using the one-to-one parameters cannot present the relative QoS situation. If one uses the variations of all users’ one-to-one parameters, new metrics can be derived to measure the difference of the absolute QoS and hence provide the threshold value of relative QoS that a multiparty multimedia service will demand. A very good example of relative QoS requirements is with online gaming. A very light worse delay will result in failure in the game. New metrics have to be used to define exactly how small the relative delay the online gaming requires. There are many other services, e.g. online biding, online stock market, etc., need a rule to judge the relative QoS requirement. Therefore, it can be seen the importance of new metrics to feed this need. Two groups of parameter are proposed.

To conveniently define new metrics, all of the users in the same multiparty multimedia communication are called a user group. This user group should not be mixed with the multicast user group. Group members could use either pure unicast or multicast to communicate or even
mixed, i.e. some of the users in the group could use unicast while others use multicast.

When talking about a new metrics it always needs an observe point that is one of the users in the group. The new metrics can be classified into two groups based on the fact that one user could be either a source or a receiver. Therefore, one group of metrics will describe the performance of the traffic coming out from the group to one particular user and another group describe the performance of the traffic going into the group. They are one-to-group parameters and group-to-one parameters.

These new proposed parameters are established on the base of the one-way metrics defined in the corresponding RFCs in the IPPM working group. To be compatible, no modification should be added to those one-way metrics in any aspects.

### 7.1.1 One-to-group Parameters Metrics

One-to-group parameters are defined to measure the performance in the view of a group user. Two subset parameters are introduced:

1. One-to-group (algorithm) mean
   a) One-to-group mean delay
   b) One-to-group mean jitter
2. One-to-group variation
   a) One-to-group delay variation
   b) One-to-group jitter variation

The one-to-group parameters are measured based on only one source in a multiparty communication group. Whenever saying one-to-group parameter, it should be associated with a source. The Figure 7-1 shows this concept.
In Figure 7-1, user A, B, C, D and E belong to the same multicast group. User D is the only active source in the multicast group when measuring the one-to-group parameters. User B and C are connected with user D through terrestrial IP network, user E are in the same Ethernet with user D, and user A are connected with user D via a satellite network. The one-to-group parameters measured in this scenario should be associated with user D.

### 7.1.1.1 One-to-group (Arithmetic) Mean

One-to-group mean parameters are trying to measure the overall network performance for a multiparty communication group. The definition of the One-to-group mean is the mean of a one-way parameter, such as one-way delay, one-way jitter and packet loss rate, measured simultaneously on all of the group members except of the active source. The word "simultaneously" implies the one-way parameter should be measured based on the same sampling interval at each user.

The One-to-group mean parameter can be calculated as:

\[
P_{OGM\_para} = \frac{\sum_{i=1}^{N} P_i}{N}
\]

where \( P_{OGM\_para} \) is the One-to-group mean parameter, \( P_i \) is the corresponding one-way parameter. \( N \) is the number of the users except the active user in the group during the sampling interval. "para" means the one-way parameter's type such as delay, jitter and packet loss rate.

**Metric Name:**
Type-P-One-to-group-Mean-Parameter

The "Parameter" could be any one of the one-way parameter defined in IPPM including delay, and jitter.

**Metric Parameters:**
- Src, the IP address of a source
- Grp, the multicast group address is multicast or empty for non-multicast
- M, a derived value corresponding to one-way parameter

**Metric Units:**
The value of a Type-P-One-to-group-Mean-Parameter is depends on what one-way parameter is used. It should be the same corresponding to the one-way metrics defined in IPPM.

**Methodologies:**
As the metric is derived from the corresponding one-way metric, the methodology to obtain those one-way parameters can be referred to the corresponding RFCs. This thesis only discuss the
methodology to derive One-to-group mean metric from one-way parameter without consideration of details on synchronization, test packetizing, time and etc.

1. Simultaneously measure the interested one-way parameters, one-way delay, one-way jitter or packet loss, on all of the receivers in a multiparty communication group when there is only one source active.

2. Calculate the mean of one-way metric value using equation (6.1) to obtain the One-to-group mean metric for this source. The question of when to calculate the One-to-group mean metric will be discussed later.

3. Change the active source and repeat the step 1 and 2 until all of the group members have been active as sources.

In the second step of the methodology, it has to be decided when to do the One-to-group mean metric calculation. There are three ways to do so. The first way is to do the calculation based on each packet arrival. The active source sends packet one by one with sequence number in the packet headers so that all receivers could identify each packet. The One-to-group mean calculation is executed for each packet received by all receivers. The resulted metric is similar to the singleton metrics defined for one-way parameters corresponding to every packet received by all users. It will provide the most accurate record of the group mean during a sampling interval with the heaviest calculation overhead.

The second way to calculate the One-to-group mean is to use the mean of one-way parameter rather than the parameter itself. The calculation could be scheduled to be executed periodically. For instance, it can be triggered for every \( T \) seconds. During the \( T \) seconds, all one-way parameters measured have to be recorded at each receiver. At each \( T \) second, the mean of the recorded parameter will be calculated first at each receiver and used as \( P_t \) in equation (6.1) to calculate the One-to-group mean metric value. This way can reduce the heavy calculation overhead required by the first one. However, it would provide less detailed information and need more storage space to record one-way parameters for more than one packet.

The third way to calculate the One-to-group mean metric is to mix the previous two ways together. One can periodically calculate the One-to-group mean parameter using directly the corresponding one-way parameter metric value rather than using its mean. For instance, the calculation can be prearranged to be triggered for every \( T \) seconds. The receivers don't need to record the one-way metric value for all of the packets received during each \( T \) seconds. One would calculate the One-to-group mean metric value at each \( T \) second using the corresponding one-way parameter of the latest received packet. Therefore, the One-to-group mean metrics of all receivers calculated at the same time would not be for the same packet. However, that would not affect engineers to use these metrics because they can still present the network situation at each \( T \)
second. Hence, the sequence number seems not necessary for One-to-group mean delay and jitter metrics. However, it still has to been added to the test packets to provide the packet loss notification. By calculating the One-to-group mean metrics in this way, It can overcome the requirement of big storage space on each receiver and the calculation overhead. One point has to be mentioned here is the calculation of the One-to-group mean packet loss rate. Because the packet loss rate itself is a statistic parameter for a certain measurement interval, the second way has to use to calculate the One-to-group mean packet loss rate.

Clearly, the One-to-group mean calculation period $T$ is a very important factor in the implementation of the measurement. If it is too small, it will not save any calculation overhead. If it is too big, it might loss most of the network performance information. And it might be different for various applications as well. Therefore, how to find an appropriate $T$ will depend on different applications. The calculation period $T$ will be discussed later within the QoS optimisation algorithm proposed.

### 7.1.1.2 One-to-group Variation

One-to-group variation parameters are trying to measure how the QoS varies among all of the users in a multiparty communication group relative to one source. The word "variation" in this chapter is the population standard deviation. The definition of the One-to-group variation is the population standard deviation of a one-way parameter, such as one-way delay and one-way jitter, measured simultaneously at all of the group members except of the active source. Therefore, there are One-to-group delay variation and One-to-group jitter variation. The word "simultaneously" implies the one-way parameter should be measured based on the same sample interval at each user. Considering the case shown in Figure 7-1 as an example, when $D$ is active, a set of packets are simultaneously monitored from $P_i$ to $P_n$ on all of the rest 4 users respectively. Then, the interested one-way parameter of these packets is calculated for each of user. The corresponding One-to-group mean metric could be calculated based on the one-way parameter. Finally, the One-to-group variation parameter can be calculated as the variation of these 4 values of the one-way parameter measured on 4 receivers. The One-to-group variation parameter can be denoted by $P_{OGV\_para}$, where the symbol "para" means the one-way parameter's name such as delay, jitter and packet loss rate. And the calculation should be:

\[
P_{OGV\_para} = \sqrt{\frac{\sum_{i=1}^{N} (P_i - P_{OGM\_para})^2}{N}}
\]

where $P_i$ is the one-way parameter value (delay, jitter and packet loss rate) and $P_{OGM\_para}$ is the corresponding One-to-group mean parameter value. $N$ is the number of the receivers.
Chapter 7. Performance Parameters and Relative QoS for MMC

Metric Name:
Type-P-One-to-group-Variation-Parameter

The "Parameter" could be any one of the one-way parameter defined in IPPM including delay, and jitter.

Metric Parameters:
- Src, the IP address of a source
- Grp, the multicast group address is multicast or empty for non-multicast
- V, a derived value corresponding to one-way parameter

Metric Units:
The value of a Type-P-One-to-group-Variation-Parameter is depends on what one-way parameter is used. It should be the same corresponding to the one-way metrics defined in IPPM.

Methodologies:
As the One-to-group variation parameter metric has to be derived on the base of the group mean metric, the corresponding One-to-group mean parameters have to be calculated first. So the methodology become simple inheriting from the one defined for the One-to-group mean metric.

1. Find out the One-to-group mean parameters
2. Calculate the One-to-group variation parameters using equation (6.2).
3. Repeat the step 1 and 2 for all users in the same multiparty communication group.

As the One-to-group variation parameters must be derived based on the One-to-group mean parameter, its calculation must be corresponding to the one of the One-to-group mean parameter described in section 7.1.1.1.

7.1.2 Group-to-one Parameter Metrics

Group-to-one parameters are defined to measure the QoS in the view of one multiparty communication user with respect to the fact that this user is receiving from more than one source in the group. Similar to the one-to-group parameters, two subset parameters are proposed:

1. Group-to-one member (arithmetic) mean
   a) Group-to-one mean delay
   b) Group-to-one mean jitter
2. Group-to-one variation
a) Group-to-one delay variation

b) Group-to-one jitter variation

The group-to-one parameters are measured based on only one receiver in a multiparty communication group. Whenever one refers to group-to-one parameters, they should be associated with the receiver. The Figure 7-2 shows this concept.

![Figure 7-2: Group-to-one measurement scenario example](image)

Figure 7-2 shows almost the same information as Figure 7-1. The difference is in Figure 7-1, user D is the receiver who received data from all of the rest group members simultaneously or consequently. The group-to-one parameters measured in this scenario should be measured and associated with user D.

In the following sections, these parameters and their metrics will be defined. The definitions are very similar to one-to-group parameters. One might question the necessity of having separate definitions of one-to-group and group-to-one parameters. The answer is positive and will be discussed after the definition.

### 7.1.2.1 Group-to-one (arithmetic) Mean

Group-to-one mean parameters are trying to measure the QoS of a multiparty communication group received by one user. The definition of the Group-to-one mean parameter of a user is the mean of a one-way parameter, such as one-way delay and one-way jitter, measured on that user when it simultaneously receiving data from the rest of the users in the group. The word "simultaneously" implies the one-way parameter should be measured based on the same sample interval on the measured user. The Group-to-one mean parameter can be calculated as:

$$P_{GOM_{\text{mean}}}(\text{param}) = \frac{\sum_{i=1}^{N} P_i}{N}$$  \ (6.3)
where $P_{GOM_{\text{para}}}$ is the Group-to-one mean parameter, $P_i$ is the corresponding one-way parameter from each of the source to the measured user. $N$ is the number of the users except the measured user in the group during the sampling interval. "para" means the one-way parameter's name such as delay and jitter.

**Metric Name:**

Type-P-Group-to-one-Mean-Parameter

The "Parameter" could be any one of the one-way parameter defined in IPPM including delay and jitter.

**Metric Parameters:**

- Dst, the IP address of a receiver
- Grp, the multicast group address is multicast or empty for non-multicast
- M, a derived value corresponding to one-way parameter

**Metric Units:**

The value of a Type-P-Group-to-one-Variation-Parameter is depends on what one-way parameter is used. It should be the same corresponding to the one-way metrics defined in IPPM.

**Methodologies:**

As the group-to-one mean parameter metric is also derived based on the corresponding one-way parameter metric, this thesis only discusses the methodology to derive Group-to-one mean metric from the one-way metric without consideration of the synchronization details, test packetizing, time and etc..

1. Simultaneously measure the interested one-way parameters, one-way delay or one-way jitter, on the measured user while all of the rest of users in the multiparty communication group sending data to it. All the one-way parameters should be measured based on the source and destination pair.

2. Calculate the mean of one-way metric value using (6.3) to obtain the Group-to-one mean metric for the measured user. The question of when to calculate the group-to-one mean metric will be discussed later.

3. Change the active source and repeat the step 1 and 2 until all of the group members have been measured.

Clearly the three ways proposed for the One-to-group mean parameter can be used to calculate the Group-to-one mean parameter. The only difference is the former has many measurement points
and calculation has to be done with the information provided by all of these measurement points while for the later all information needed for the group-to-one parameter can be provided by a single measurement point.

### 7.1.2.2 Group-to-one Variation

Group-to-one variation metrics are trying to measure how the QoS varies at one user in the multiparty communication group when the rest of users sending data to it. The definition of the Group-to-one variation is the population standard deviation of a one-way parameter, such as one-way delay and one-way jitter, measured at one user in a multiparty communication group while all of the rest group members sending data simultaneously to it. Therefore, it can give Group-to-one delay variation and Group-to-one jitter variation. The word "simultaneously" implies the one-way parameter should be measured based on the same sample interval at the measured user.

Considering the case shown in Figure 7-2 as an example, when D is chose as the measured user, a set of packets from \( P_i \) to \( P_n \) sent by each of the rest 4 users respectively should be simultaneously monitored. Then, the interested one-way parameter of these packets is calculated for each pair of users, i.e., D and A, D and B, D and C and D and E. The corresponding Group-to-one mean metric could be calculated based on the one-way parameter. Finally, the Group-to-one variation parameters are calculated as the variation of these 4 values. The One-to-group variation parameter can be denoted by \( P_{GOV-para} \), where the symbol "\( para \)" means the one-way parameter's name such as delay and jitter packet, and the calculation should be:

\[
P_{GOV-para} = \sqrt{\frac{\sum_{i=1}^{N} (P_i - P_{GOM-para})^2}{N}}
\]

Where \( N \) is the total user number in the multiparty communication except the measured one and \( P_i \) is the one-way parameter for each of the \( N \) users.

**Metric Name:**

Type-P-Group-to-one-Variation-Parameter

The "Parameter" could be any one of the one-way parameter defined in IPPM including delay, and jitter.

**Metric Parameters:**

1. Dst, the IP address of a receiver
2. Grp, the multicast group address is multicast or empty for non-multicast
3. \( V \), a derived value corresponding to one-way parameter
Chapter 7. Performance Parameters and Relative QoS for MMC

Metric Units:

The value of a Type-P-Group-to-one-Variation-Parameter is depends on what one-way parameter is used. It should be the same corresponding to the one-way metrics defined in IPPM.

Methodologies:

The methodology can be simply inherited from the one defined for the Group-to-one mean metric as:

1) Find out the Group-to-one mean parameters
2) Calculate the Group-to-one variation parameters using the equation (6.4)
3) Repeat the step 1 and 2 for all users in the same multiparty communication group

As the Group-to-one variation parameters must be derived based on the Group-to-one mean parameter, its calculation must be corresponding to the one of the Group-to-one mean parameter described in section 7.1.2.1.

7.1.3 Reasons for Two Groups of Similar Parameters

As mentioned in the beginning of section 7.1.2, the definitions of One-to-group parameters and Group-to-one parameters are very similar. There are reasons they should be separately defined. Firstly, it is because of the metric parameter definition. The One-to-group metrics have a common parameter, \(Src\), the IP address of the active source during the measurement interval. It must be changed to \(Dst\) parameter for the Group-to-one metrics to present the measured user. It's not like the case for the one-way parameter measurement where the destination and the source are single host in the same level. They can be exchanged in the measurement without any difficulty. Therefore one metric is enough for measurement between one pair of hosts. In the multiparty communication, the source and the destination cannot be exchanged because one of them represents more than one user. Two metrics have to be defined for the measurement in the two directions. For instance, if user A and user B communicates with each other, the one-way delay metric can be used for both direction traffic by exchanging the \(Src\) and \(Dst\) parameter [51]. However, if user C joins their communication, the proposed new metrics have to be used to measure the QoS for the multiparty communication. The One-to-group mean delay metric and the One-to-group delay variation can show clearly the QoS received by user A and user B in the group relative to user C. One cannot use the same metrics to measure the QoS received by C relative to both user A and user B by simply exchanging the \(Src\) and \(Grp\) parameter in the metric because of the methodology described for One-to-group parameter.

Secondly, Group-to-one and One-to-group parameters and their metrics should be defined separately because of the transporting technologies used for multiparty communications. There
might be the coexistence of both unicast and multicast in either direction. One host in a multiparty communication group might use unicast to receive data from other hosts and use multicast to send data to the others. The delay of each direction would be different due to the difference of the transport technologies. If it can be said that for one-to-one communication, delays for both directions can be approximately the same, it might not have the same conclusion for the multiparty communications. Therefore, two groups of metrics are needed to describe the network situation regarding the traffic direction.

7.1.4 Relative QoS and the Proposed Multiparty Multimedia Communication Parameters

There is an interesting point in the methodologies for obtaining the Group-to-one parameters that all of the users send data simultaneously rather than work separately in order. The same question can also be asked for the methodology of the One-to-group parameter. As discussed in the motivation part that the second reason of proposing these new parameters and their metrics is that the multiparty communications have extra requirements on the relative QoS beside the absolute QoS. These proposed parameters could be used to describe and measure the relative QoS, which implies that all the one-way parameters needed to derive the Group-to-one and One-to-group parameters have to be measured in the same measurement interval rather than separately in an order. The relative QoS cannot be described by comparing the same parameter for different connections in different time slots.

Here are some examples to show how to use the proposed metrics to describe and measure the relative QoS. For instance, the relative delay can be measured by using the Group-to-one and One-to-group delay variation metric. Group-to-one delay variation measures the difference of delays received by one user in a multiparty communication group relative to the rest sources. A centralized multiparty communication where all clients have to communicate with the rest group members through a central server might require the transmission delays from all clients to the server satisfy a Group-to-one delay variation threshold to guarantee that no clients suffer much bigger delay than others or enjoy much smaller delay than others. Typical examples are the services that need their users to compete with each other. The One-to-group delay variation measures how different for each user to receive data from one source in a multiparty communication group, which is another relative QoS issue.

An example of the use of the proposed metrics might be the adaptable priority optimisation algorithm. The basic idea is to dynamically change the priority for each group member according to the network situation to guarantee that all members in the group are served with relatively similar QoS. In other words, the Group-to-one variation and One-to-group variation parameters
should be kept under a certain threshold to satisfy the relative QoS requirements for various applications. The detail of this adaptable priority optimisation algorithm will be described later.

### 7.1.5 Analysis of the Possible Measurement Errors

Errors caused by the measurement of the one-way parameters are not going to be discussed in this thesis because they can be found in the corresponding RFCs. Errors introduced by the proposed metrics have to be discussed in this section. The reason of these errors is the packet loss in the network. When a packet never arrives its destination, its delay might be hidden from the results.

When considering how to calculate the proposed metrics, three means were considered. The first way provides one-way parameter metrics corresponding to each received packet. That means for each packet one metric can be found for it. Then error caused by packet loss can then be easily sorted out before calculating the Group-to-one and One-to-group parameters.

However, for the other two ways, either a mean or the last packet received by the measurement point during a period of time is used to present the interested one-way parameter. If there are any packets lost in the period of time, they will be ignored by the calculation of the multiparty communication parameters. For instance, the calculation of the multiparty communication parameters is done every $T$ seconds. Then for the second way, the mean of the one-way delay in $T$ seconds could be infinity if any packets are lost during that $T$ second and infinity is a valid metric value for the one-way delay metric. Then our Group-to-one and One-to-group mean parameters and variation parameters could be infinity too. This infinity does not mean anything in terms of relative delay for multiparty communication. The conclusion should not be had that during that $T$ seconds, users in the group suffered significantly different delay.

For the third way where only using the latest packet received during a $T$ seconds time slot, if all of the packets were lost during the $T$ seconds, which is quite possible since $T$ could be a very short time, the one-way metric value uses to calculate the multiparty communication parameters will be the latest packet receiver in last $T$ seconds. Clearly, the result will not reflect any information of the network performance during the current $T$ seconds and therefore, it turns out to be an error.

The possible calibration can be done by using more sophisticated way to calculate the multiparty communication parameters. For instance, one can ignore all the one-way metrics with infinity value when calculating the multiparty communication parameters using the second calculation way. One can find out which $T$ seconds suffers from packet loss and do not calculate the multiparty communication parameter for it using the third way. There might be other methods to handle the errors, which are not discussed here. As long as they can avoid leading to the wrong analysis results, they can be implemented in the applications.
Chapter 7. Performance Parameters and Relative QoS for MMC

7.2 QoS Optimisation Algorithm Proposed for Multiparty Multimedia Communications

Despite of the question on how to define and measure the absolute one-to-one QoS parameters for multiparty multimedia communications as discussed in [1] [9] and [7], the relative QoS is another very important factor as discussed in previous sections. The main difference of the multiparty multimedia communication from the one-to-one real-time communications is the concept of a "group". Additional to the one-to-one communication, thanks to IP networks, the multiparty multimedia communication technologies enables communication among a group of people. This new feature introduces the challenge to the existing one-to-one QoS concept. When a group users trying to share the same information from sources, each of them expects the similar QoS in order to compete with others. If some users suffer worse QoS than the others, fairness problems are raised. This issue appears stronger in the real-time heavily interactive multiparty multimedia communication services, e.g. Internet gaming, online real-time contest and bidding, online stock market, etc. Even for those services without heavily interactive behaviours, e.g. the multimedia conference, people may still suffer from this fairness problem especially when the QoS differences between the users are significant. A conferencing service via satellite is a good example. People who are not in the same satellite spot beam with the current source will have fewer chances to get the floor to show their opinion than those located closely to the current source. With awareness of this fairness issue, a QoS optimisation algorithm was proposed to minimize the variance of the QoS among multiparty multimedia communications.

Some research efforts have been put to reduce the relative delay in the interactive multiparty service by adding additional entities to the network. [2] proposed to use delay control entities on both server and user's side to decrease the relative delay while increasing the absolute delay. The proposed algorithm in this thesis was developed based on already existing technologies rather than introducing more complexity to the network. Moreover, the idea to increase the absolute delay is not a good choice in that real-time applications are all vulnerable to absolute delays. The proposed algorithm utilizes the measurement methodologies and parameters to solve this relative QoS issue based on optimizing traffic priorities.

The fundamental of DiffServ [58] is to differentiate services by classifying the traffic. That implies that it is trying to deal with the relative QoS of those traffic classes. For instance, some algorithms manipulate the traffic priorities on routers to let some traffic pass through more quickly while others have to wait for a longer time. So the basic idea is to assign the users who are suffering worse delay a higher priority traffic class to decrease the absolute delay and, in another hand, to improve the relative delay with other users. However, there are more questions to be answered to support this algorithm.
Firstly, what environment the algorithm will be deployed? The trend of the multiparty communication is to utilize the multicast technologies to use the bandwidth more efficiently and enable many-to-many communications. The algorithm proposed in [16] will be the possible solution to enable the different QoS class for individual multicast branch. Also, the communication topology will be centralized server and distributed clients based because to support the fully meshed multicast connections will be typically difficult since the change of the sources will make the delay change much more often and that will trigger much more traffic class modifications. The complexity and the dynamics of the fully meshed multicast might be too high to be efficiently supported in an IP network. However, that is also possible for a future work. The client-server based topology is also highly practical for many multiparty communications mentioned before. Actually, some of them, e.g. online gaming and online stock market, have to be implemented based on this topology. Furthermore, the server-client topology implies single source multicast. SSM will be used to enable the multicast routing. Another point that needs to be mentioned is that multicast only happens when the server acts as a source and sends the same information to all clients. Clients have to use unicast to send data to the server and allow server to update all the remaining clients. This is essential for a central server and distributed client topology.

Secondly, when to trigger the traffic class modification function? This question should be answered by the dynamic measurement that will be executed on both server and client. For the unicast traffic from clients to the server, the server will periodically calculate the current Relative Delay Variation (RDV) value by collecting the one-way delays from all clients currently connected. The RDV is the population standard deviation. It can be denoted by $\delta$ and the calculation should be:

$$\delta = \sqrt{\frac{\sum_{i=1}^{N} (x(i) - \mu)^2}{N}}$$  \hspace{1cm} (7.1)

where $x(i)$ is the delay for connection from the $i^{th}$ client and $\mu$ is the arithmetic mean of these delay (Mean Delay). $N$ is the total number of clients currently connecting with the server. Clearly, RDV mentioned in this algorithm is actually an example of the One-to-group variation parameters proposed in previous sections of this chapter.

The server sets up a threshold value for this RDV, which can be adjusted to satisfy different requirements of different applications. For instance, FPS (First Person Shooter) online games will have much higher requirement on RDV than a videoconference service. If the calculated RDV is bigger than the threshold, it will trigger the delay modification function to upgrade the class of the traffic from the client who is currently suffering from the worst delay to give it higher priority till
Chapter 7. Performance Parameters and Relative QoS for MMC

the threshold is satisfied. And if the threshold cannot be satisfied when the traffic from the client already has the highest priority, the server will degrade the traffic from the client with the shortest delay to a lower priority. At last, if the threshold still cannot be satisfied, that means the network situation is beyond the optimisation limit, therefore, the server can either give up optimisation or notify the client with the worst delay to modify its connection situation.

There is a reason to increase the traffic class to give the client with the worst delay a higher priority rather than decrease the one with best delay. The multimedia interactive applications have high requirement on absolute delay, which can be represented by the Mean Delay $\mu$ in equation (7.1). Therefore, the better choice is to improve the Mean Delay at first by increasing the priority of the client with worst delay unless it cannot less the RDV below the threshold value. When the server manages to lower a client traffic priority, it has to check if the resulted Mean Delay is up another threshold required by the application to avoid the absolute delay requirement being abandoned. This implies the system has an assumption that any violations of the RDV to the threshold are because the network situation becomes worse rather than better. The weak point of this assumption is that the priorities of the traffic from clients will become higher and higher. For instance, the server has A, B and C three connections. In the beginning, the RDV for all of these three connections satisfies the threshold in the server. Connection A suddenly suffers the worst delay and the server upgrades its priority. Later, the network situation for connection A becomes much better and results in a much shorter delay. This rises up a new violation to the RDV threshold. However, the server will not degrade the priority for connection A but upgrade connection C, which currently suffers the worst delay, to satisfy to threshold. If this situation repeats, three connections will be upgraded to highest priority at last before the server managed to degrade them. This is not economic since more network resources will be occupied for this high priority that is not necessary. There are some means to avoid this. One is to set up both the lower bound and upper bound for Mean Delay. So the server will check the upper Mean Delay bound to decide if it should degrade the priority for a connection first rather than always upgrade the priority. Another method is to let the server remember two delays for each connection collected at the last two continuous measurement time points. Before it executes the class modification function, it will have to check the reason of the RDV violation to know whether it's because one connection suffers from a worse delay caused by the dynamic network or another connection benefits from a better delay. This can be implemented by comparing the two delays for each connection to check if they become much worse or better. this can be the future work.

The above method can optimise the relative delay for the unicast connections from clients to server. However, it's just half of the optimisation algorithm for the multiparty communication. The other half of the algorithm has to be considered to optimise the multicast connections from the server to clients. The need of considering the unicast and multicast transmission separately is
based on the fact that two routing algorithms might use different routings and might suffer from
different delays. The idea to optimise the multicast connections is very similar with the part
dealing with unicast ones. The difference is that the delay for each multicast branch has to be
measured on each client rather than on the server. Then all clients should send the measured delay
to the server for the RDV calculation. If the RDV is higher than the pre-set threshold, the class
modification function will be triggered based on the principle described for the unicast
connections.

It is always true for multicast connections that the transient end-to-end parameters, i.e. delay,
jitter, packet loss, can only be measured on the receivers. This is why the optimisation of a
meshed multicast connected system becomes much more difficult. For Any Source Multicast
(ASM), any end user can be either receiver or source or both of them at the same time. It will be
very difficult to decide where to measure the delay and where to calculate the RDV. For Source
Specific Multicast (SSM), each end user can be a receiver in one multicast group and can measure
the delay for the connection from the source to itself. It can also be a receiver for several multicast
groups and has to decide to which source it should send the just measured delay. It can be a
source for another multicast group at the same time and, therefore, has to collect delays measured
by all of its receivers and be responsible to calculate the RDV and trigger the class modification
function. This will make the algorithm very complex in each of the end users involved in the
communication. That's one of the reasons to choose the client-server topology for optimisation.

The algorithm will face another fairness problem caused by multicast. When the signalling tries to
optimise the traffic class for one client, it might affect all the clients connected to the same
multicast tree branch. If the server sends multicast data to more than one client through one edge
router mark point, the traffic class of all of these clients might be changed when our algorithm is
triggered. This is because the edge router marks traffic based on the source and group information
carried by the packets. For multicast, all packets carrying (S, G) information will be marked by
the edge router and optimising traffic class for individual client is difficult. However, on the other
hand, it's very possible that all clients connected to the same multicast tree branch might share the
same network situation, e.g. the congestion that causes the delay. Therefore, the real cost raised
by the algorithm might be much lower than thought. The degree of the extra cost then will depend
on the network situation. There might be means to minimise this extra cost. There are two
possible solutions. One solution is to make the edge router more powerful. After duplicate the
multicast packets, it will mark them based on the outgoing port before it forwards them. A similar
method can be found in [16] that was proposed to satisfy the different QoS requirements of
receivers in the same multicast group. However, this solution cannot completely solve the fairness
problem because it is possible that more than one client attach to same outgoing port of the edge
router. Moreover, the edge router might not duplicate the multicast packet at all. The second
solution to this fairness problem is to switch the connection between the server and the client with worst delay from multicast to unicast. Then our optimisation algorithm will work well for this unicast connection. The draw back is to have a very complex connection situation that includes both multicast and unicast that will result more complicate algorithm accordingly. Furthermore, something has to be done so that the unicast connection may switch back to multicast when necessary, e.g. the congestion causing the bad delay disappears.

This algorithm is not only proposed to optimise the relative delay but can also be used to optimise other relative QoS parameters, i.e. the relative packet loss rate and relative jitter. Thanks to the traffic differentiation in DiffServ, every class of traffic can have several constraints, which can include all QoS parameters that users care for. One should only set the dynamic measurement parameter to the particular one-to-group parameter interested on the server to trigger the traffic class modification functions. For instance, if applications care packet loss rate more than delay, the server will set a threshold for the packet loss ratio variation and dynamically calculate it to trigger the traffic class modification function when it breaks the threshold. The packet loss ratio constraint defined for in proper traffic class will reduce the relative packet loss ratio. The only difference between optimising the relative delay and the relative packet loss ratio is the method used to calculate these two parameters in that delay can have value on any time point while packet loss rate is always for a period of time. However, this thesis are not going to discuss more details about how to implement this algorithm to optimise other parameters rather than the relative delay. It should be pointed out that this algorithm is not only feasible for optimising the relative delay but also can be implemented for more general purposes. The following simulation results will show that as long as the class modification function is triggered, the packet loss ratio will be improved besides the delay due to the class constraint definition.

7.2.1 Simulation for the Proposed QoS Optimisation Algorithm

With the proposed QoS optimization algorithm in mind, a simulation was built up to verify its availability of and testify how much it can improve the relative QoS in a multiparty communication environment. All of the simulation was built using ns 2 (Network Simulator 2) version 2.27 [11]

Ns is a discrete event simulator targeted at networking research. It provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. Ns has always included substantial contributions from other researchers.

Many necessary functions have been added to the ns to enable the proposed QoS optimisation algorithm with the guide from [12] and [13]. They were implemented using C++ code. These changes include creating two new agencies and two new application flows, adding Extreme
distribution function, and modification on the JoBS (Joint Buffer Management and Scheduling) [13] block, etc. Each of these new functions is briefly described in the following sections.

For consistency with previous work, the application flows have been expected to be multiparty multimedia conferencing source streams. However, it has been found that these kinds of stream are not easy to find on the Internet, especially the common characteristics of this application. It is heavily affected by codex used by the multiparty multimedia conference applications. In contrast, the Internet gaming, another typical multiparty communication application is an ideal alternative to provide input as application flows for its common traffic characteristics discovered by researchers in the last few years. It was found that for one kind of game, for instance, the First Person Shooting (FPS) game, many game applications generate traffic with the same distributions that can be easily adopted to the simulation. The second reason that the Internet gaming was chosen as the application flows in the simulation is that both the multiparty multimedia conferencing services and the Internet gaming services have requirement on relative QoS and no matter which of them is chosen for the simulation, it will not effect the proposed relative QoS optimization algorithm.

7.2.1.1 New Application Flows

Application flow is the highest level in the ns simulator. It is equivalent to the application layer in the real networks and it generates traffic and passes it to an agent who is responsible to send packets to the network. To simulate the multiparty communication in a client-server topology, the online game was chosen to be the application for its popularity and the high requirements regarding relative QoS. Two application flows were created to generate traffic, Gameclient and Gameserv. They model the performance of a game client and a game server respectively. For the Gameclient, its packet length has an extreme distribution and its packet interarrival time has a deterministic distribution. The extreme distribution [15] has a CDF as:

\[ F(x) = e^{-\frac{x-a}{b}} \]  

(7.2)

Its PDF is shown as:

\[ f(x) = e^{\frac{x-a}{b}} \frac{e^{-\frac{x-a}{b}}}{b} \]  

(7.3)

A set of random variable with an extreme distribution has to be generated for simulation purpose. As [15] derived, it can be:

\[ x = a - b \ln[-\ln U(0,1)] \]  

(7.4)

where U(0,1) is a uniform distribution bounded by 0 and 1.
For the Gameserv, it has extreme distribution for both packet length and interarrival time [14] [15]. Since the ns does not provide the extreme distribution function originally, it was added into the rng.h as "inline double extreme (double a, double b) {return (a-b*log(-log(uniform()));}" and random.h as "static double extreme(double a, double b) {return rng()->extreme(a, b);}" and enabled by recompiling. The code of the implementation of the two application flows can be found in Appendix A and Appendix B.

7.2.1.2 New Agents

In ns, agents play the transport layer role, which include network layer packet construction and consuming, acknowledging the requirements of applications and so on. Agents are used in the implementation of protocols at various layers. As described in the previous section, signalling will be exchanged in the proposed algorithm between clients and server to notify the corresponding network elements, i.e. the Marker and Demarker in JoBS. Therefore, two new agents are needed to carry out the dynamic measurement and the traffic class modification signalling functions.

Since most of the real-time multiparty communication services use UDP as their transport layer protocol, the existing UDP class provided in ns was inherited and new functions were added to create two agents. One of the agents working on the server was named as Udpserv, and the one on the client Udpclient. Udpserv agent is responsible to update a packet information array that has the latest packet information received for each client. For instance, if N clients communicating with the server, then the array will have N elements and each of them is the information of the latest packet received from one of these N clients. The packet information includes the packet sending time, arrival time, source address traffic class to which it belongs. One timer is associated with the agent and for a certain sampling interval, which is the sampling period T described in the proposed parameters sections. It triggers the RDV calculation procedure, which is the implementation of (7.1), using the packet sending times and arrival times stored in the packet information array. This timer marks the crucial interval that will impact the efficiency of the algorithm. It can be derived using this formula:

$$T_{rdv} = D_{link} \times 2 + T_{tran}$$

(7.5)

where $T_{rdv}$ is the RDV calculation interval, $D_{link}$ is the link delay and $T_{tran}$ is the source transmission interval. Equation (7.5) shows clearly that the RDV calculation interval should be equal to the double of the link delay plus the source transmission interval. This is because that the server has to wait long enough for one signalling procedure finishing before it starts another signalling procedure. It has also to wait for the next data packet arrival after the signalling procedure finish in order to get the modified delay for the next RDV calculation. The $T_{rdv}$ in the simulation was set to 160ms in all simulation scenarios while the link delay $D_{link}$ is 60ms and
the client transmit packet with a deterministic interval time $T_{Tran}$ of 40ms.

The delays and RDV are dumped into a trace file for further results analysis. If the RDV is bigger than a pre-defined threshold, one signalling packet will be sent to the proper client after further processing. If the packet suffering from the worst delay does not belong to the highest priority, a signalling packet with a special class code point of 255 but without data payload will be sent back to the source of this packet. Otherwise, if packet suffering from the worst delay has the highest priority and the Relative Delay Mean is under a pre-defined threshold, a signalling packet will be also sent to the source of the packet with a special class code point of 254. Figure 7-3 and Figure 7-4 shows the flow chart of the class modification function and the packet receiving function respectively.

---

![Figure 7-3: Class modification trigger function flow chart for Udpserv agent](image-url)
Chapter 7. Performance Parameters and Relative QoS for MMC

Figure 7-4: Receiving packet function flow chart for Udpserv agent

_Udpclient_ agent will respond any received packet with special class code point, either 255 or 254. The responding process is to send another signalling packet to the server with a special class code point of 253, corresponding to the class code point of 255, or 252 corresponding to the class code point of 254. The reason to have twice signalling packet transmission is because the _Marker_, the network element that marks the packets priority in JoBS, is directive. It means the _Marker_ that is responsible to mark the packets from one client is only aware of the traffic from the client to the server. So after the server realise that one client should modify its traffic class, it cannot directly notify the _Marker_ responsible for the client. Therefore, the server has to notify the client first and the client will then signal the appropriate _Marker_. The codes of both _Udpclient_ and _Udpserv_ can be found in Appendix C and D. Figure 7-5 shows the flow chart of the packet receiving function for _Udpclient_.

Figure 7-5: Receiving packet function flow chart for Udpcient agent

Additional to these two agents, two other agents were created for the server without the signalling functions. It will be used in a non-QoS optimised environment to provide trace file for comparison with the optimised result.
7.2.1.3 JoBS Modification

The JoBS algorithm in ns is to provide absolute and relative (proportional) loss and delay differentiation independently at each node for classes of traffic [13]. It can classify traffic with 4 priorities. The traffic with a class of 1 have the highest priority and lowest priority if class of 4. For each class, a set of performance requirements are specified to the algorithm as a set of per-class QoS constraints.

JoBS performs scheduling and buffer management in a single pass and dynamically allocates services rates to classes in order to satisfy the delay constraints. If no feasible service rate allocations exist, or if the packet buffer overflows, packets are dropped according to the loss constraints. Ns also provides another DiffServ algorithm besides JoBS. However, its performance constraint for each traffic class is packet dropping rather than packet delay. It is not suitable to our relative delay optimisation algorithm. With JoBS, one can optimise the delays by manipulating traffic classes.

The JoBS consists of three elements that are:

- **JoBS Links** should be established to connect all the core routers and enable them to recognise the priority code point and perform the scheduling and queue management algorithm accordingly.

- **Marker** is responsible to marking the traffic and working as an edge router defined in the DiffServ.

- **Demarker** is used to collect the end-to-end delay statistics.

All packets sent by a node have the traffic class code point of “0” and will be marked by the Marker with a pre-defined traffic class.

Some modifications were made to all of these three elements so they can understand the signalling packets and give them special treatments. All signalling packets with special class code point of 255 and 254 are not going to be marked by Marker but only be directly forwarded by all JoBS elements in order to reserve the special class code points. But they have to be treated as packets with the highest priority in all of three JoBS elements. There are two considerations at this point. The first one is that these signalling packets don’t do anything with the network. They are targeted at the clients and should not affect the JoBS algorithm. Secondly, the network situation should have minimum effect to these signalling packets so that their transmission can be most possibly guaranteed. That’s why they have to be forwarded with the highest priority. Packets with class code point of 253 and 252 have to be detected by the Marker. When a Marker receives a signalling packet with class code point of 253 and the current class for this packet is bigger than 1, it will decrease the class by 1 for all of the following packets from the source of that signalling
packet. It’s important to clarify again that the higher the class code point is, the lower the priority associated with a flow. Therefore, the signalling packet with class code point of 253 can notify the Marker to increase the priority for the flow from its source. And, similarly, the signalling packet with class code point of 252 will trigger the Marker to decrease the priority by 1 for the flow from its source.

Hence, generally speaking, the modifications in ns enabled the JoBS algorithm to change the class of traffic dynamically according to the RDV measured at the server.

### 7.2.2 Simulation Configuration

With all the necessary functions ready, the simulation has to be configured for further results. The simulation configuration in ns has to be done using TCL (Tool Command Language). The “multiparty” implies more than two users are involved in the communication. Three users were set up in the simulation to present this concept while keeping the simulation simple and easy to debug and analysis. Each of the users connects a server via core networks to organize the server/client topology. Edge routers of the core networks would enable the traffic classifications. Several additional sources are also needed in the simulation to generate the background traffic. Thanks to the JoBS TCL examples provided with ns, a simulation was successfully configured with three clients and one server. It also has 6 UDP cross-sources for background traffic. Totally 25 nodes connected to simulate a client-server based network. The following figure shows the layout of the simulation.

In the simulation, 9 nodes, $R_i$ ($i \in [1,9]$), work as routers. $R_1$, $R_3$, $R_4$, $R_6$, $R_7$, and $R_9$ are edge routers as well as Markers and Demarkers. Link delays are configured for the link between these routers so each three of them present a domain. I.e., $R_1$, $R_2$ and $R_3$ present a domain; $R_4$, $R_5$ and $R_6$
present another domain; and the rest three routers present the third domain. All domains have been configured with the same link delay. Each of the three game clients, $G_1$, $G_2$, and $G_3$, are connected with one game server, GS, through one of the three domains via the corresponding edge routers. The 6 UDP cross-sources, $CS_j$ ($j \in [1,6]$), and their 6 destination sinks, $SN_k$ ($k \in [1,6]$), are also connected through three domains as shown in Figure 7-6. All these cross-sources have Pareto On-Off application flow attached to the UDP agent and they will send traffic with Pareto distribution to the simulation network. Links connecting end users and the corresponding edge routers were named as edge link and links between routers were named as core link. Edge links always have bigger bandwidth than the core links like the real networks.

The most important thing is the configuration of the game client, the game server and how to connect them. The Figure 7-7 shows the client and server configuration layout. Each game client has two agents. One is UDP agent and another one is Udpc1ient. A Gameclient application flow is attached to the Udpc1ient agent. The game server has two agents as well. One is UDP and another one is Udpserv. A Gameserv application flow is attached to both UDP and Udpserv agents. The connections between those four agents are as shown in Figure 7-7. The client application flow is responsible for sending modelled game client data through the Udpc1ient to the Udpserv agent on the server, while the server application flow generates the modelled game server traffic and sends it to the UDP agent at each client node via its UDP agent. The algorithm signalling will be carried out between the Udpc1ient and Udpserv agents.

![Figure 7-7: configuration of client and server](image)

### 7.2.3 Simulation Results and Analysis

To have a complete study on the proposed QoS optimization algorithm, the simulation was organized into three scenarios. Each of these scenarios has a goal to help testify and study the algorithm with a particular consideration. For the convenience of description, the following content of the simulation results and analysis will be organized into three parts according to those scenarios.
Chapter 7. Performance Parameters and Relative QoS for MMC

Scenario 1: Lightly loaded network

This scenario tried to confirm the feasibility of the proposed algorithm in a lightly loaded network environment with a sudden heavy source striking one of the domains. The heavy source will cause congestion in the domain, which will result in worse delay to the client that connects the server through this domain. Thus, ideally, the algorithm will be triggered and optimize the relative delay. The lightly loaded network environment means the utilities of all edge links and core links are lower than 100% and the queuing delay can be ignored in this case. Therefore, the end-to-end delay equal to the pre-defined link delay.

To prove that this algorithm can improve the RDV, the simulation runs with and without the QoS optimization algorithm. For convenience, they will be called QoS simulation and Non-QoS simulation.

In both simulations, each game client generated 1750 packets. No class modification happened to client G2 and G3 in this scenario. Client G1 sent out 389 class 4 packets, 2 class 3 packets, 2 class 1 packets and 1360 class 1 packets. It indicated that G1 stayed at traffic class 3 and class 4 for only a very short of time before its traffic was assigned the highest priority. The calculation of the RDV was executed 1397 times to monitor the relative delay situation. These calculations resulted in totally 6 signalling packets exchanged between client G1 and the server GS in Figure 7-6 These signalling packets successfully triggered 3 class modification procedures for the traffic from G1 as described in the previous paragraphs. The overhead of this algorithm in this scenario is 6 packets out of 1750 packet, which is around 3%. However, this figure could be dramatically different for various network situations. The more dynamic the network is regarding to the RDV threshold, the more overhead it will be. Since the signalling packet doesn’t have any data payload, the overhead will also depends on the throughput of the application. The proposed algorithm successfully improved the RDV in terms of the mean value from 1.8500e-2s in a best-effort network to 6.7032e-4s in an optimized network. A closer look of the simulation result in terms of the dynamic RDV against time is shown in Figure 7-8.

The solid black curve, Non-QoS RDV in the legend, presents the RDV measured in the Non-QoS simulation and the blue dash curve, QoS RDV in the legend, presents the RDV in the QoS simulation. In the beginning of the simulation, the network was lightly utilized and all three clients, G1, G2, G3, suffer from almost the same delay that is the configured link delay. In this phase, the optimisation algorithm keeps on monitoring the RDV at the server without further behaviour that results in the solid black curve and the blue dash curve overlapped. At the 15th second, a new cross-source CS7 becomes active and generates traffic to overload the connection, C_{R1-R3}, between the edge router R1 and R3. It stops at 25th second. This new background traffic causes the client G1 suffering bigger delay due to the increased queuing time, and it finally
triggers the class modification procedure. The procedure was triggered three times and the class of the traffic from G1 was decreased from 4 to 1 at last, which means its priority increased from the lowest to the highest to satisfy the predefined RDV threshold at the server. Figure 7-8 shows clearly that the QoS simulation had a much lower RDV than the Non-QoS simulation during the 15th second to the 25th second. Therefore, the proposed QoS algorithm successfully proved that it could improve the RDV in the situation of scenario 1.

Then, a further question emerges as how much this algorithm can improve RDV. The answer is quite depends how much effect the new cross-source can bring to the network. Actually, it depends on how big the queuing delay is increased for the client G1. If the queuing buffer of the edge router R1 is $L$ and it service rate is $\lambda$, the queuing delay for $G_1$ could be up to $\frac{L}{\lambda}$ theoretically. If the RDV threshold, $D$, can always be satisfied by the algorithm, the benefit can be $(\frac{L}{\lambda}) - D$. That means the bigger dynamic the network is, the more benefit it can gain from the QoS optimisation algorithm.

Scenario 2: Effects of traffic class constraints

As found in the Scenario 1 that totally 6 signalling packets are exchanged between the client $G_1$ and the server to change the class of the traffic from $G_1$ 3 times from class 4 to class 1. Two considerations regarding the traffic class constraints were raised from this result. One is that the traffic from $G_1$ had the highest priority after the QoS optimisation algorithm was triggered. It implies the most cost the user of $G_1$ has to use. Does he really need that high priority to ensure the RDV under the threshold? Figure 7-8 shows that the QoS RDV is much lower than the Non-QoS RDV. It was found that the big difference only caused by the last class modification. That means when the traffic class was modified from 4 to 3 and from 3 to 2, the RDV didn’t satisfy the

![Figure 7-8: RDV simulation result for scenario 1](image-url)
threshold. Then the final change of the traffic class from 2 to 1 was the key change and it makes
the RDV finally much lower than the threshold. Therefore, the first two class changes wasted time
and the finally class change gave the traffic too high priority that consumed more resources than it
really needed. This analysis result showed that the gap between each pair of adjacent classes
could cause time waste and resource over consuming.

Another consideration regarding the traffic class constraints is that it can reduce the number of the
signalling packets using less class modification and, hence, decrease the overhead of the
algorithm.

These two considerations about scenario 1 were taken into account in this scenario. This scenario
was trying to investigate if the algorithm can work more efficiently regarding the resource over
consuming and the overhead by optimise the class constraints. As described in section 7.2.1.3
(JoBS Modification), the relative delay constraint in JoBS was expressed in terms of proportions.
In scenario 1, it was configured that each traffic class has a delay as 4 times long as the previous
one. That means if class 1 only has absolute delay constraints \(d\), class 2 will have absolute delay
\(4d\). Class 3 will have absolute delay \(16d\), and class 4 will have absolute delay \(64d\). It can be seen
that the absolute delay gap between class 3 and class 4 become much bigger than the gap between
class 1 and class 2. This resulted in that the absolute delay of class 4 became unacceptably bigger
than class 1. If the absolute delay constraint of class 1 is not small enough, the following classes’
absolute delay will be too high for the network transmission. The RDV was not lower than the
threshold in the scenario 1 after the first two class modifications because even the absolute delay
constraints of the traffic class 2, caused by the corresponding relative delay constraints, is not
lower enough to affect the RDV. In another hand, the absolute delay constraints of the highest
priority class cannot be defined too low in order to avoid that one traffic class occupy all of the
network resources. Therefore, the delay constraint gaps between each pair of the adjacent classes
have to be adjusted.

In this scenario, the traffic class relative delay constraints were modified so that if traffic class 1
has an absolute delay constraint \(d\), class 2 will have \(2d\) absolute delay constraint. Class 3 and class
4 will have absolute delay constraint \(4d\) and \(16d\).

The simulation results show clearly that this time it had only one class modification with only two
signalling packets exchanging between client \(G_i\) and the server within the simulation duration. It
means after adjusting the relative delay constraints of the traffic classes, it successfully shortened
the time needed for tuning the RDV under the threshold and the overhead of the algorithm was
reduced from 6 signalling packets to 2 signalling packets. It proved the analysis that the QoS
optimisation algorithm can work more efficiently with an optimised traffic class definition. The
mean RDV between the 15\(^{th}\) second and the 25\(^{th}\) was improved from \(1.7600e-2s\) in a best-effort
network to 9.7689e-4s in an optimized network. Comparing with the result of scenario 1, it can be seen that the mean of the QoS RDV in this scenario is bigger. That also implies that traffic from G1 to server occupied less network resource in this scenario. There are more details of how this happened in Figure 7-9.

In Figure 7-9, the red dashed curve presents the QoS RDV and the black solid curve is the Non-QoS RDV. If comparing this figure and Figure 7-8, it can be found that during the overload period, the QoS optimisation algorithm produced higher QoS RDV in this scenario, which is also below the RDV threshold, than it did in scenario 1. This implies that in this scenario the traffic from G1 needs less resource in the same network than it did in scenario 1 to satisfy the RDV requirement. Therefore, it was proved that well-defined traffic classes could save network resources within the proposed QoS optimisation algorithm.

Scenario 3: Heavily loaded network

The previous two scenarios proved that the algorithm could efficiently improve RDV with light transmission overhead in a lightly loaded network with well-defined traffic class constraints. The aim was to investigate if it could work in a heavily loaded network in this scenario with a competitive cross-source.

All of the six cross-sources, CSj (j e [1,6]), were configured to generate traffic heavy enough to introduce long queuing delay for all the three clients and trigger the traffic class modification procedure. And, again, a new cross-source, CS7, was connected to the network and began to generate traffic at 15th second with the highest priority after the simulation started. It stopped at the 25th second after 10 seconds raising congestion at the edge router R1. It was expecting that the algorithm can realize that only increasing the traffic priorities will not able to solve the RDV problem in this scenario and it will try to approach an optimisation by decrease priorities of some
traffic.

<table>
<thead>
<tr>
<th></th>
<th>Class 1 packets No.</th>
<th>Class 2 packets No.</th>
<th>Class 3 packets No.</th>
<th>Class 4 packets No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client G₁</td>
<td>1710</td>
<td>5</td>
<td>9</td>
<td>29</td>
</tr>
<tr>
<td>Client G₂</td>
<td>1058</td>
<td>326</td>
<td>354</td>
<td>17</td>
</tr>
<tr>
<td>Client G₃</td>
<td>841</td>
<td>824</td>
<td>74</td>
<td>21</td>
</tr>
</tbody>
</table>

Table 7-1: Simulation results for scenario 3

Simulation results showed that the traffic from each client was distributed in all traffic classes adjusted by the proposed QoS optimisation algorithm in a heavily loaded network environment. Table 7-1 shows the packet distribution in each traffic class for each client. Totally 16 class modifications procedures were triggered during the simulation time with 33 signalling packets exchanged between clients and the server. These numbers are much bigger than the ones in scenario 1 and 2, where the network was lightly loaded. It implies that in a heavily loaded network, the dynamics has more impact on the RDV.

One error can be found in Table 7-1: that the number of signalling packets is 33 rather than 32, the double of the class modification times. This error was caused by the large packet delay in the heavily loaded network. The server didn’t receive packets with updated priority after one RDV calculation period, please refer to equation (7.5), due to the large network delay. Therefore, it made the wrong decision that another class modification procedure should be triggered and sent out an extra signalling packet to the corresponding client. The client found its traffic already had the highest priority and ignored this signalling packet. Thus, it had totally 1 extra signalling packet. It can be easily removed by setting a longer RDV period by adding an estimate factor to the right side of the equation (7.5).

It also showed that the algorithm successfully adapted traffic from all clients to the right traffic classes and resulted in a much better mean RDV from 2.8500e-2s in a best-effort network to 2.6000e-3s in an optimized network. Figure 7-10 shows the RDV result curves for both cases with and without the algorithm. It can be clearly seen in Figure 7-10 that right from the beginning to the very last of the simulation, the optimisation algorithm made the QoS RDV lower than the Non-QoS RDV. It’s different in the lightly loaded network scenarios, where the algorithm mainly contributed to the optimised QoS RDV during the period when CS₇ was generating traffic. That means the heavily loaded network provide a worse environment for services with high RDV requirement.
Figure 7-10: RDV simulation result for Scenario 3

As said in the beginning of this scenario, it was expected that the priority decrease procedure happened during the simulation when the algorithm found it necessary. Please refer to section 7.2.1.2 for details of the trigger conditions of this procedure. Table 7-2: shows the details of the priority manipulation in Scenario 3.

<table>
<thead>
<tr>
<th></th>
<th>No. of Priority Increase</th>
<th>No. of Priority Decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client G₁</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Client G₂</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Client G₃</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td>12</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 7-2: Priority manipulations in Scenario 3

Table 7-2: shows that totally 12 priority increase procedures were successfully executed during the simulation as well as 4 priority decrease procedures. In this simulation, increase of priority means decrease of the traffic class number. All the traffic decrease procedures happened between 15th second and 25th second when the traffic with highest priority generated by the CS. That means during that period of time, the RDV threshold cannot be satisfied even though all traffic from clients have the highest priority. Then the algorithm managed to lower the priority of the traffic with the lowest absolute delay to decrease the RDV. And clearly the connection between the server and Client G₃ was mostly affected in this case.

Moreover, this algorithm has extra achievement besides the optimised RDV in this scenario. Traffic between G₁ and the server suffered from packet loss during the simulation and the proposed algorithm reduced the number of packet loss from 173 to 53. This is because the traffic constraints include the packet loss rate upper bound and the higher priority, the lower it is. This constraint gave us the additionally benefit in the simulation regarding the packet loss. Therefore,
it can be said that with the well-defined traffic class constraints, the proposed algorithm cannot only improve the RDV but also the RLV (Relative Loss Variation). This is proof that adaptively adjusting the traffic priority can optimise more relative QoS parameters beside the RDV. Although the algorithm implemented here is specifically for optimisation of the RDV, it can achieve optimisations on more relative QoS parameters as long as the relevant traffic class constraints are defined.

However, the packet loss did cause problems in our simulation because the proposed algorithm works based on the end-to-end delay measurement with the potential assumption that packet can arrive the server to provide transient network information. The packet loss, especially the packet with the highest priority loss, led the wrong information to the proposed algorithm and result the wrong decision. Therefore, there might be some unnecessary class modifications happened during the period when CS7 was active. For instance, at time $T_1$, the algorithm found that the RDV threshold is violated and then sent a signalling packet to increase the priority of Client $G_i$. The procedure has been successfully finished but the following several packets with the higher priority from $G_i$ are lost due to the congestion. At time $T_2$, the algorithm will calculate the RDV using the last arrival packet from $G_i$, which is the one actually arriving before the priority was changed. Thus, the calculation will, again, violate the RDV threshold and another priority increase procedure will be triggered, which might be not necessary. If the network between $G_i$ and the server are seriously congested, the server might keep on sending priority increase signalling packet to the network and that will enlarge the overhead of the algorithm. Moreover, if the highest priority traffic caused the packet loss, even the signalling packet can not be successfully exchanged between clients and servers. The algorithm will loss its functionality. Therefore, it will need an admission control algorithm to make sure no overload for the highest priority traffic.

The packet loss also affects the result of the non-QoS simulation because the Non-QoS RDV in Figure 7-10: and the mean RDV were calculated based on the received packet as well. Therefore, they are not true. The packet dropping was because that the router couldn’t allocate the service rate required by the class delay constraint. So the Non-QoS RDV curve was expected to have a large raise above the delay constraint defined for the lowest priority traffic between 15th second and 25th second. Therefore, it can be said that the proposed algorithm can provide even better RDV than shown in Figure 7-10: .

After the simulation of the proposed algorithm, it gave strong evidence that the adaptive class optimisation concept should be applied to a more general scale in the DiffServ networks. Assigning a fixed priority to any services might cost more resources than it actually needs for certain QoS guarantee. For instance, one does not need to assign his traffic a high priority when the network is lightly loaded since there are enough resources available for it. Another drawback of the fixed priority is that it might not provide the QoS guarantee the traffic desired when the
network becomes busy. For instance, as seen in the proposed simulation scenario 3, when there is heavy traffic with the same priority or even higher priorities, a fixed priority will provide poor QoS guarantee in the resource competition. Thus, an adaptive priority assign algorithm with the consideration of the end-to-end or multi-to-multi QoS requirement might work more efficiently in the framework of DiffServ. This will be our future study direction.

7.3 Summary

This chapter firstly proposed a group of new parameters and their metrics for multiparty multimedia communications to easily and accurately measure the network performance as well as the relative QoS. Then an optimization algorithm was proposed to improve the relative QoS based on the measurements of these parameters.

All IPPM QoS parameters are defined for one-to-one connections. They did not consider the case of group communications, or multiparty communications as called in this thesis. Two main reasons were discussed in this chapter. One is that it can not use end-to-end QoS parameter to efficiently and accurately describe the performance of a group. Another reason is directly using end-to-end metrics cannot present the relative QoS required by multiparty communications.

To overcome this gap, two new sets of QoS parameters and their metrics were proposed for multiparty communications derived from the existing one-to-one QoS parameters. They are One-to-group and Group-to-one parameters. One-to-group parameters were defined to measure the QoS in the view of a group user. Group-to-one parameters were defined to measure the QoS in the view of one multiparty communication user with respect to the fact that this user is receiving from more than one source in the group. Each of them contains two subsets parameters that are (algorithm) mean and variation. Each of two subgroups consists of two parameters corresponding to the one-way delay and one-way jitter parameter defined in IPPM. The parameter metrics were defined following the rules of IPPM working group, which includes metric name, metric parameter and methodologies. In the methodologies part, three possible measurement means were discussed and compared.

It also pointed out the needs of two groups of parameters. It is both because of the metric parameter definitions and the different transporting technologies used for multiparty communications.

This chapter then discussed the possible measurement errors that might be resulted in the measurement of those proposed parameters. It is because when a packet never arrive its destination, its delay might be hidden from the result. Solutions were described that can be deployed to overcome these errors caused by the packet loss in terms of different measurement
methodologies.

Group communication introduces the challenge to the existing end-to-end QoS concept. When a group users trying to share the same information from sources, each of them expects the similar QoS in order to compete with others. If some users suffer worse QoS than the others, the fairness problems are raised. The proposed algorithm is to utilize the measurement methodologies and proposed parameters to solve this relative QoS issue based on the knowledge of DiffServ. The basic idea is to assign the users who are suffering worse delay a higher priority traffic class to decrease the absolute delay and, in another hand, improve the relative delay with other users.

The communication topology used with the algorithm will be a centralized server and distributed clients based because to support the fully meshed multicast connections will be typically difficult due to the change of the sources will make the delay change much more often and that will trigger much more traffic class modifications. SSM is used to enable the multicast routing in this topology. The dynamic measurement that was executed on both server and client is the trigger of the traffic class modification function. Relative Delay Variation (RDV) is periodically calculated at the server as an example of the One-to-group parameters. If the calculated RDV is bigger than the threshold, it will trigger the traffic class modification function to upgrade the class of the traffic from the client who is currently suffering from the worst delay to give it higher priority till the threshold is satisfied.

The proposed algorithm is not only proposed to optimise the relative delay but can also be used to optimise other relative QoS parameters, i.e. the relative packet loss ratio and relative jitter. With the awareness that every class of traffic can have several constraints, which can include all QoS parameters that users care for, it should only set the dynamic measurement parameter to the particular one-to-group parameter interested at the server to trigger the traffic class modification functions.

Simulations have been carried out to verify the proposed algorithm. The results showed that the proposed algorithm can provide less relative delay variation in both heavily and lightly loaded networks. With well-defined traffic class constraints, the simulation shows that very small transmission overhead will be required by the proposed algorithm to achieve better relative delay variation in a lightly loaded network. The algorithm can provide more benefit on relative delay variation in a heavily loaded network as well as the improved packet loss situation when relevant class constraints are defined.
8 Conclusions and Future Works

Multiparty multimedia communications is playing a more and more important role in the IP network communications. With the support of satellite global coverage and on board processing ability, one can provide effective multiparty multimedia communication services in the next generation Internet. This thesis describes research on multiparty multimedia communications over GEO satellite with focus on design and evaluation of an IP conferencing over satellite system. It developed a hybrid routing architecture using both unicast and multicast routing technologies for the conferencing system to efficiently provide the IP conferencing service over a GEO satellite. A multiple MCU conference model was proposed to accompany this routing architecture for the conferencing service to avoid media data sending over multiple satellite hops.

With the network traffic measurement technologies studied, this thesis described how a test regime was developed to evaluate the designed IP conferencing over satellite system. Furthermore, the relative QoS requirement of the multiparty multimedia communications has been pointed out after the measurement and evaluation. A set of parameters and their metrics have been proposed to present this requirement. One relative QoS optimization algorithm has been proposed using those parameters and it was verified using simulation with positive results.

8.1 Conclusions

Multiparty multimedia communications are developed based on the VoIP technologies, which have reviewed in the beginning of this thesis to build up the solid ground for the later research work on the multiparty multimedia conferencing services over GEO satellite. These VoIP technologies include signalling protocols, such as H.323 and SIP, Real-time Transport Protocols, the RTP, and its accompany control protocol RTCP.

A constraint for a model that is suitable for a multiparty multimedia conferencing system over satellite has been identified, which is that the model should not require multiple satellite hops for delivering the media data. A new conference model, multiple media servers or multiple MCU model, was proposed under this constraint to support the efficient and flexible conferencing over satellite. In this model, end users send multimedia streams to the MCUs, which collect the streams, manipulate them and generate multicast flows to send to other users through the satellite network. This model minimizes the bandwidth in comparison to the unicast only conference and simplifies the end user terminals’ requirements.
Chapter 8. Conclusions and Future Works

Associated with the multiple MCU conference model, a hybrid routing architecture was proposed, which consisted of both unicast and multicast routing technologies. This routing architecture adopts inter-domain multicast technologies that are originally designed for terrestrial networks, i.e. the PIM-SM/MSDP/MBGP suite, into a satellite centralized integration network. It takes advantage of the MCUs to enable both unicast users and multicast users to join the conferencing services in order to be compatible to the current unicast-mainly Internet and to make it easy to migrate to a full multicast enabled Internet in the future.

Traffic measurement technologies have been studied to measure and evaluate the designed conference system. A measurement regime developed for this purpose has validated all technologies proposed. All the functionalities and the delays introduced by those technologies were tested in several different test scenarios on a demonstrator. The objective test designed with a set of parameters as well as the corresponding metrics and measurement methodologies was carried out. The result showed that the conference system can provide less than 300ms end-to-end delay including 25ms MCU delay as well as around 0.27% end-to-end packet lost ratio and less than 9.1ms end-to-end jitter.

The study shows that all IPPM QoS parameters were defined for one-to-one connection, which can not be used to efficiently and accurately describe the performance of group communications or multiparty multimedia communications as it is called in this thesis. Directly using end-to-end metrics cannot provide the relative QoS required by multiparty multimedia communications. Two new sets of relative QoS parameters and their metrics have been proposed in this thesis for multiparty multimedia communications. They are One-to-group parameters, which are defined to measure the QoS in the view of a group user and Group-to-one parameters, which are defined to measure the QoS from the view point of one multiparty communication user with respect to the fact that this user is receiving from more than one sources in the group. These new proposed parameters have been documented into an Internet Draft (ID) and adopted as a working item in IPPM working group in IETF.

Using the proposed multiparty multimedia communications relative QoS parameters, a new algorithm was proposed to optimize the relative QoS for multiparty multimedia communications. It allows clients to adapt the appropriate traffic priority based on the dynamically measurement relative QoS parameters in order to optimize the relative QoS. The simulation showed that the proposed algorithm can provide less relative delay variation in both heavily and lightly loaded networks. With well-defined traffic class constraints, the simulation showed that very small transmission overhead would be required by the proposed algorithm to achieving better relative delay variation in a lightly loaded network. The algorithm can provide more benefit on relative delay variation in a heavily loaded network as well as improved the packet loss situation when relevant class constraints are defined. It showed that this algorithm can also optimise other
relative QoS parameters, i.e. the relative packet loss ratio and relative jitter.

8.2 Discussion and Future Work

The new relative QoS parameters and their metrics proposed in this thesis, have been submitted to the IETF as an individual Internet Draft (ID) [67]. In the last IETF meeting in Paris in August 2005, this ID was presented in the IPPM working group session. The IPPM showed lots of interest and decided to pick this ID up as a working group study item to attract more research attention to make it a complete concept for traffic measurement of the multiparty multimedia communications. Some work is being carried out in the IPPM on the composition and decomposition metrics [68] that has a close relationship with the multiparty metrics being proposed in this thesis. Research on these proposed parameters and metrics will continue with collaboration on the composition and decomposition metrics in the IPPM working group and try to contribute to standardization.

There can be some extensions from the relative QoS optimization algorithm proposed in this thesis, which will lead to future research. For instance, the class modification functions can be more sophisticated to decide if the priority of the traffic needs to be increased or the other one should be decreased. By using both signalling to exchange the network situation between clients and the server and the relative delay measurement, the priority of traffic might be decreased when the network is no more congested.

The adaptive class optimisation concept in the proposed algorithm can be extended to a more general case. Assigning a fixed priority to any services might cost more resources than it actually needs for a certain QoS guarantee. For instance, one does not need assign his traffic a high priority when the network is lightly loaded since there are enough resources available for it. Another drawback of the fixed priority is that it might not provide the QoS guarantee the traffic desired when the network becomes busy. For instance, as seen in the simulation scenario 3, when there is heavy traffic with the same priority or even higher priorities, a fixed priority will provide poor QoS guarantee in the resource competition. Thus, an adaptive priority assignment algorithm with the consideration of the end-to-end or multi-to-multi QoS requirement could work more efficiently. This is another future study direction.
References


[14] Johannes Farber, "Network Game Traffic Modelling", First Workshop on Network and
References

System Support for Games (NetGames 2002), Braunschweig, Germany, April 16-17, 2002.


References


[49] “Universal Mobile Telecommunications System (UMTS); Selection procedures for the choice of radio transmission technologies of the UMTS (UMTS 30.03 version 3.2.0),” TR 101 112, v3.2.0 (1998-04), Universal Mobile Telecommunications System, European Telecommunications Standards Institute, pp. 34-35.


References


December 2004.

Appendix A: Codes of Client Application

Flow

File name: Game.h
File type: Header file
File codes:

```c
#ifndef ns_game_h
#define ns_game_h

#include "timer-handler.h"
#include "app.h"
#include "config.h"
#include "packet.h"
#include "address.h"

class UdpAgent;
class TfrcAgent;
class GameApp;

class GameAppTimer : public TimerHandler {
    public:
        GameAppTimer(GameApp* t) : TimerHandler(), t_(t) {};
        inline virtual void expire(Event*);
    protected:
        GameApp* t_;
};

class GameApp : public Application {
    public:
        GameApp();
        void timeout();
    protected:
        void start();
        void stop();
        inline double next();
        double interval_;
        int running_;
        GameAppTimer timer_;
};

#endif
```

File name: Game.cc
File type: C++ file
File codes:

```c
#include "random.h"
#include "udp.h"
#include "ip.h"
#include "game.h"

static class GameAppClass : public TclClass {
    public:
        GameAppClass() : TclClass("Application/Game") {
            TclObject* create(int, const char*const*) {
                return (new GameApp);
            }
        }
}
```

160
GameApp::GameApp() : running_(0), timer_(this)
{
    bind("interval_", &interval_);
}

void GameAppTimer::expire(Event*)
{
    t_->timeout();
}

void GameApp::start()
{
    running_ = 1;
    double t = next();
    timer_.sched(t);
}

void GameApp::stop()
{
    running_ = 0;
}

void GameApp::timeout()
{
    if (running_)
    {
        /* call the TCP advance method to send packet with extreme
distribution*/
        agent_->sendmsg((int)Random::extreme(80, 5.7));
        /* reschedule the timer */
        double t = next();
        timer_.resched(t);
    }
}

double GameApp::next()
{
    /* for game client, interarrival time is deterministic at 40ms which can
be input using tcl by user*/
    return (interval_);
}
Appendix B: Codes of Server Application Flow

File name: Gameserv.h
File type: header file
File codes:

```c
#ifndef ns_gameserv_h
#define ns_gameserv_h

#include "timer-handler.h"
#include "app.h"
#include "config.h"
#include "math.h"
#include "packet.h"
#include "address.h"
#include "agent.h"
#include "tclcl.h"
#include "ip.h"

class UdpAgent;
class TfrcAgent;
class GameservApp;
class GameservAppTimer;

class GameservAppTimer : public TimerHandler {
public:
   GameservAppTimer(GameservApp* t) : TimerHandler(), t_(t) {}  
   inline virtual void expire(Event*);
protected:
   GameservApp* t_;  
};

class GameservApp : public Application {
public:
   GameservApp();
   void timeout();
   Agent* agent_;
protected:
   void start();
   void stop();
   inline double next();

   double interval_;  
   int running_;  
   GameservAppTimer timer_;  
};
```

File name: Gameserv.cc
File type: cc file
File codes:

```c
#include "random.h"
#include "udpserv.h"
#include "gameserv.h"

```
#include "ip.h"
#include "tclcl.h"
#include "packet.h"
#include "address.h"

static class GameservAppClass : public TclClass {
public:
    GameservAppClass() : TclClass("Application/Gameserv") {
        TclObject* create(int, const char* const*) {
            return (new GameservApp);
        }
    } class_app_gameserv;

    // the GameservApp initializes all timers and the running status
    GameservApp::GameservApp() : running_(0), timer_(this) {
    }

    void GameservAppTimer::expire(Event*) {
        t_ -> timeout();
    }

    void GameservApp::start() {
        running_ = 1;
        /* set up the timers for sending next packet*/
        double t = next();
        timer_.sched(t);
    }

    void GameservApp::stop() {
        running_ = 0;
    }

    void GameservApp::timeout() {
        if (running_) [
            /* call the UDP advance method to send packet with extreme distribution*/
            agent_ -> sendmsg((int) Random::extreme(120, 36));
            /* reschedule the timer */
            double t = next();
            timer_.resched(t);
        ]
    }

    double GameservApp::next() {
        /* for game server, interarrival time is extremely distributed with parameter (55, 6)*/
        return Random::extreme(55, 6) * 0.001;
    }
}
Appendix C: Codes of Udpclient Agent

File name: Udpclient.h
File type: header file
File codes:

```c
#ifndef ns_udpclient_h
#define ns_udpclient_h
#include "timer-handler.h"
#include "agent.h"
#include "tclcl.h"
#include "packet.h"
#include "trafgen.h"
#include "address.h"
#include "ip.h"
#include "config.h"
#include "math.h"
#include "udp.h"

class UdpAgent;
class TfrcAgent;
class UdpclientAgent;

class UdpclientAgent : public UdpAgent {
public:
    UdpclientAgent();
    virtual void recv(Packet*, Handler*);
};
#endif
```

File name: Udpclient.cc
File type: cc file
File codes:

```c
#include "random.h"
#include "udpclient.h"
/*counts for signalling pkts sent by client. used to check the signalling pkt
lost.*/
int inc_count_=0;
int dec_count_=0;

static class UdpclientAgentClass : public TclClass {
public:
    UdpclientAgentClass() : TclClass("Agent/Udpclient") {}
    TclObject* create(int, const char*const*) {
        return (new UdpclientAgent());
    }
} class_udpclient;

//the UdpservAgent initializes the delay calculation timer and the running
status
UdpclientAgent::UdpclientAgent() : UdpAgent(PT_UDP) {
}
```
/*ack the priority modification trigger packets from the gameserv*/
void UdpclientAgent::recv(Packet* pkt, Handler*)
{
    // Access the IP header for the received packet
    hdr_ip* hdrip = hdr_ip::access(pkt);

    // if the packet have a special prio_, reflect it to update the Marker for client
    if (hdrip->prio_==255)
    {
        ns_addr_t addtmp;
        addtmp=hdrip->src_;  // src_ and dst_ inter-change
        hdrip->src_=hdrip->dst_;  
        hdrip->dst_=addtmp;
        hdrip->prio_=253;
        send(pkt,0);
        inc_count_++;
        printf("totally %d pkt with 253 priority sent from %d\n",
               inc_count_,hdrip->src_.addr_);
    }
    else if (hdrip->prio_==254)
    {
        ns_addr_t addtmp;
        addtmp=hdrip->src_;  // src_ and dst_ inter-change
        hdrip->src_=hdrip->dst_;  
        hdrip->dst_=addtmp;
        hdrip->prio_=252;
        send(pkt,0);
        dec_count_++;
        printf("totally %d pkt with 252 priority sent from %d\n",
               dec_count_,hdrip->src_.addr_);
    }
}

Appendix D: Codes of Udpserv Agent

File name: Udpserv.h
File type: h file
File codes:

```c
#ifndef ns_udpserv_h
#define ns_udpserv_h

#include "timer-handler.h"
#include "agent.h"
#include "tclcl.h"
#endif

#include "packet.h"
#include "trafgen.h"
#include "address.h"
#include "ip.h"
#include "config.h"
#include "math.h"
#include "udp.h"

class UdpAgent;
class TfrcAgent;
class UdpservAgent;
class GameservDelTimer;
/*establish a new timer class for delay calculation*/
class GameservDelTimer : public TimerHandler {
  public:
    GameservDelTimer(UdpservAgent* a) : TimerHandler(), a_(a){}
    inline virtual void expire(Event *);
  protected:
    /*this UdpservAgent instance is needed coz the Timerhandler function of
    sched(double delay) is gonna be run with the UdpservAgent::start()
    as delay_timer_sched(t) where the t is the delay calculation period*/
    UdpservAgent* a_;}

class UdpservAgent : public UdpAgent {
  public:
    UdpservAgent();
    //virtual int command(int argc, const char*const* argv);
  virtual void recv(Packet*, Handler*);
  void timeout(); //trigger the QOS algorithm
  //dump the delays for each packet
  void dump(double*, int, double);

  protected: //add the GameservDelTimer instance to assossiated with the
              UdpservAgent
    GameservDelTimer delay_timer_;
  
};

/* define stucture that will store the packet source address and the sending
time and arrival time. */
struct pktinfo {
    ns_addr_t pktsrc_; //packet source address
    double stime; //packet send time
    double atime; //packet arrival time
    int prio_; //packet class
};

void pktinfcp (pktinfo *, pktinfo *); //struct copy function
```
void pktinfini (pktinfo *, int ); //struct array initialization function

#include "random.h"
#include "udpserv.h"
#include "stdio.h"
define MAXUSER 6
#define MAXDELVAR 0.015
#define MAXDELMEAN 0.2 //mean delay upper bound
#define MAXCLASS 4 //lowest priority (highest class)
define MINCLASS 1 //highest priority (lowest class)

class UdpservAgentClass :  public TclClass {
public:
UdpservAgentClass() :  TclClass("Agent/Udpserv")  {}  
TclObject* create(int, const char*const*) { return (new UdpservAgent());  }
}
class_udpserv;

pktinfo pktinf_[MAXUSER]; //totally 6 user connect to the server

pktinfo del_inf; //used to calculate the packet delay
int N_node_=0; //number of node that has send the packets

bool lock_pktinf_=false; //a flag to prevent the recv() function update the pktinf_ array

//when the array is needed for the delay calculation

int pkt_counter=0; //received pkt counter, used to trigger the calculation timer upon the arrival of the first packet

FILE *temchk;
//signalling pkt count from the server.
int inc_count=0;
int dec_count=0;

void UdpservAgent::dump (double* delays_, int N_node_, double delvar){

FILE *trace_;
trace_=fopen ("rel_del_ser.trc", "a");
for (int j=0; j<N_node_; j++)
fprintf (trace_, "%f ", delays_[j]);
fprintf (trace_, "%fn", delvar);
fclose (trace_);
}

void pktinfini (pktinfo * infarr, int max)
{
for (int m=0; m<max;m++) {
infarr[m].pktsrc_.addr_=-1;
infarr[m].pktsrc_.port_=-1;
}
//the UdpservAgent initializes the delay calculation timer and the running status
UdpservAgent::UdpservAgent() : UdpAgent(PT_UDP), delay_timer_(this)
{
void pktinfini (pktinf_, MAXUSER);

void pktinfcp (pktinfo* infl, pktinfo* inf2)
{
    inf2->pktsrc_=infl->pktsrc_;  
    inf2->stime=infl->stime;
    inf2->atime=infl->atime;
    inf2->prio_=infl->prio_;  
}

//trigger the delay variance calculation
void GameservDelTimer::expire(Event*)
{
    a_->timeout();
}

void UdpservAgent::timeout()
{
    lock_pktinf_=true; //lock the pktinf_ array

    /*add the code to calculate the delay variance*/
    double delays_[MAXUSER]; //delay array
    double delays_var=0;  //delay variance
    double delays_mean=0;  //delay mean
    double delays_sum=0;  //delay sum
    double d_0=0;
    if(N_node_>1)
    {
        for (int i=0; i<N_node_; i++)
        {
            delays_[i]=(pktinf_[i].atime-pktinf_[i].stime);
        }
        //calculate the mean first
        for (int j=0; j<N_node_; j++)
            delays_sum+=delays_[j];
        delays_mean=delays_sum/N_node_;
        //the delay variance
        for (int k=0; k<N_node_; k++)
            d_0=(delays_[k]-delays_mean)*(delays_[k]-delays_mean);
        delays_var=sqrt(d_0/N_node_);
        //find the source with the biggest delay and put it in pktinf_[0]
        //and the one with smallest delay in pktinf_[N_node_-1]
        pktinfo tmp;
        for(int l=0; l<N_node_-1; l++)
        {
            for(int m=l+1; m<N_node_; m++)
            {
                if((pktinf_[l].atime-pktinf_[l].stime)<(pktinf_[m].atime-
                    pktinf_[m].stime))
                    {
                        pktinfcp(&pktinf_[l],&tmp);
                        pktinfcp(&pktinf_[m],&pktinf_[l]);
                        pktinfcp(&tmp,&pktinf_[m]);
                    }
            }
        }
    }

    //print the relative delay into a file
    dump(delays_,N_node_,delays_var);
    /*if the delay var is bigger than the max value */
if (delays_var>MAXDELVAR)
{
    //decrease the class (increase the priority) of the packet with biggest
delay if it doesn't havethe lowest class
    if(pktinf_[0].prio_>MINCLASS)
    {
        // create signalling packet
        Packet* sigpkt = allocpkt();
        hdr_ip* ipsig = hdr_ip::access(sigpkt);
        double m_del=pktinf_[0].atime-pktinf_[0].stime;
        printf("violet delay: %f, source: %d, prio: %d\n",
                m_del,pktinf_[0].pktsrc_.addr_, pktinf_[0].prio_);
        //printf("a special packet with prio of *255* was sent-------
    \n");
        ipsig->prio_=255; /*consider the duplex link between the gameserv and the
        client, this prio_ of 255 only inform the client to send one pkt with prio_ of
        253 to increase the priority*/
        // add the dst address and send it
        ipsig->dst_ = pktinf_[0].pktsrc_;
        send(sigpkt, 0);
        inc_count++;
        printf("totally %d pkt with 255 priority sent\n", inc_count);
    }
    //decrease the class of the packet with smallest delay if it doesn't have
    the lowest class
    else if (pktinf_[N_node_-1].prio_<MAXCLASS && delays_mean<MAXDELMEAN)
    {
        // create signalling packet
        Packet* sigpkt = allocpkt();
        hdr_ip* ipsig = hdr_ip::access(sigpkt);
        double m_dels=pktinf_[N_node_-1].atime-pktinf_[N_node_-1].stime;
        printf("violet delay: %f, source: %d, prio: %d\n",
                m_dels,pktinf_[N_node_-1].pktsrc_.addr_, pktinf_[N_node_-1].prio_);
        //printf("a special packet with prio of *254* was sent-------
    \n");
        ipsig->prio_=254; /*consider the duplex link between the gameserv and the
        client, his prio_ of 254 only inform the client to send one pkt with prio_ of
        252 to decrease the priority*/
        // add the dst address and send it
        ipsig->dst_ = pktinf_[N_node_-1].pktsrc_;
        send(sigpkt, 0);
        dec_count++;
        printf("totally %d pkt with 254 priority sent\n", dec_count);
    }
}
}
lock_pktinf_=false; //unlock the pktinf_ array so that the recv() function can
update it
/*reschedule the time, the delay variance calculation period will be 120ms, the
double of the link delay + client transmission interval*/
double dt=0.16;
delay_timer_.resched(dt);
}
/*store the latest packet info for each source*/
void UdpservAgent::recv(Packet* pkt, Handler*)
{
    //ignore the signalling packet
    hdr_ip* hdrip = hdr_ip::access(pkt);
    hdr_cmn* hdrcmn = HDR_CMN(pkt);
    if( (hdrip->prio_>1 && hdrip->prio_<4) ||
        (!lock_pktinf_) //update the pktinf_ array if it's not locked
Appendix D

{  
//record the pkt information to the pktinf_ array
bool in_arr_=false; //indicate if this packet source is in the pktinf_ array

//replace the last packet info from the same source
for (int i=0; i<MAXUSER; i++)
{
  if (hdrip->src_.addr_==pktinf_[i].pktsrc_.addr_ &&
      hdrip->src_.port_==pktinf_[i].pktsrc_.port_)
  {
    pktinf_[i].stime=hdrcmn->ts_arr_;  
    pktinf_[i].atime=Scheduler::instance().clock();  
    pktinf_[i].prio_=hdrip->prio_;  
    in_arr_=true;
  }
}

//if there is not src in the array match this new coming pkt. add its src into the array
if (!in_arr_)
{
  pktinf_[N_node_].pktsrc_.addr_=hdrip->src_.addr_;  
  pktinf_[N_node_].pktsrc_.port_=hdrip->src_.port_;  
  pktinf_[N_node_].stime=hdrcmn->ts_arr_;  
  pktinf_[N_node_].atime=Scheduler::instance().clock();  
  pktinf_[N_node_].prio_=hdrip->prio_;  
  N_node_++;
}

pkt_counter++;  

// when receive the first packet, set the delay calculation timer
if (pkt_counter==1)
{
  double dt=0.05; //the predefined delay variance calculation period will be 50ms  
  delay_timer_.sched(dt);
}

printf("pkt received by server: %d, node number is: %d 
",
        pkt_counter,N_node_);
printf("Node %d: send time is: %f, arrival time is: %f 
",
        hdrip->src_.addr_,hdrcmn->ts_arr_,
        Scheduler::instance().clock());
}  

}