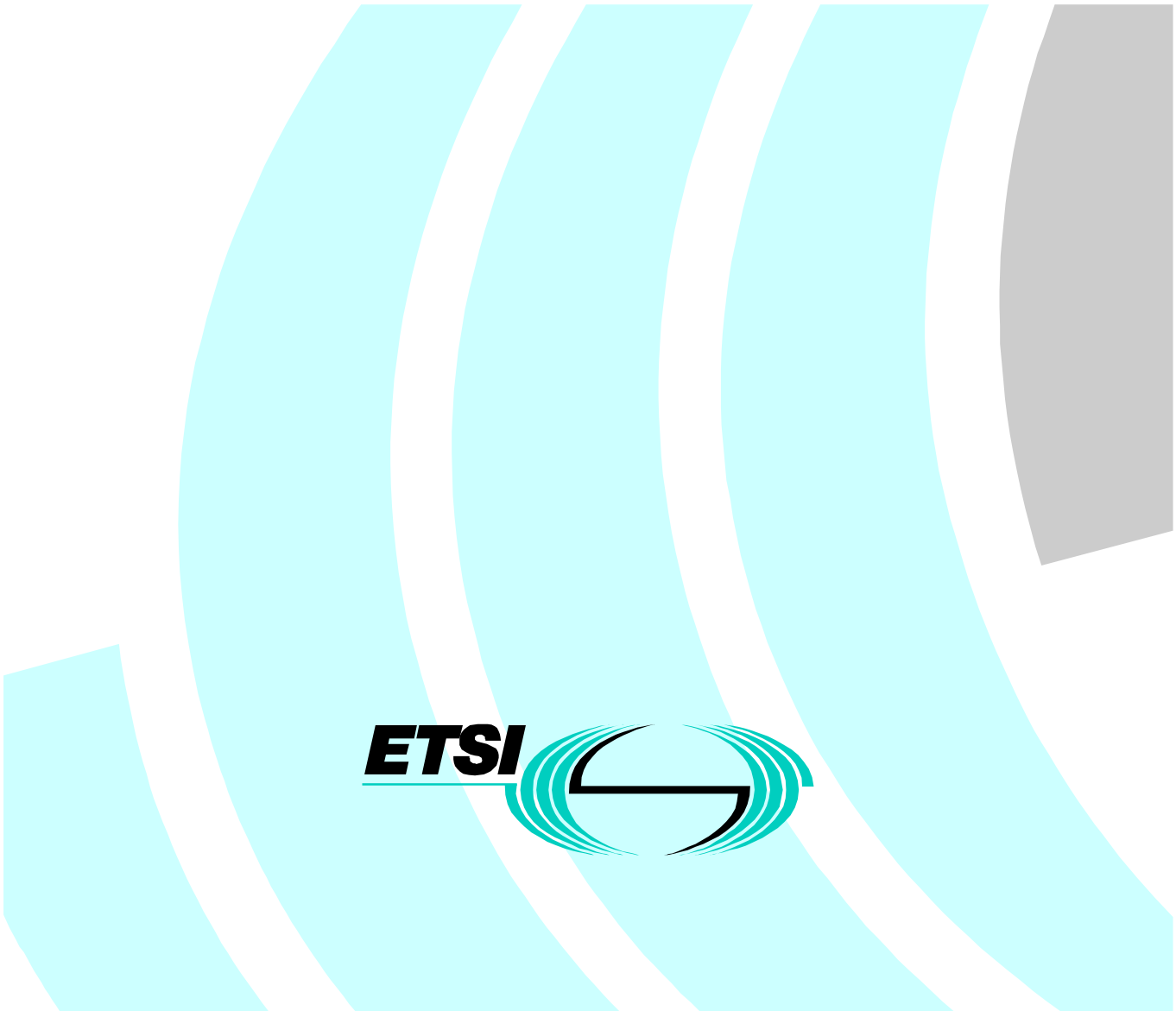
ETSI TS 101 909-5 V1.1.1 (2001-08)

Technical Specification

Access and Terminals (AT);
Digital Broadband Cable Access to the
Public Telecommunications Network;
IP Multimedia Time Critical Services;
Part 5: Dynamic Quality of Service
for the Provision of Real Time Services
over Cable Television Networks using Cable Modems



Reference
DTS/AT-020020-05

Keywords
access, broadband, cable, IP, multimedia, PSTN

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights 7
Foreword 7
Introduction 8
1 Scope 9
2 References 9
3 Definitions and abbreviations 10
3.1 Definitions 10
3.2 Abbreviations 10
4 Void 10
5 Technical Overview 11
5.1 IPCablecom QoS Architecture Requirements 11
5.2 IP QoS Access Network Elements 13
5.2.1 Multimedia Terminal Adapter (MTA) 13
5.2.2 Cable Modem (CM) 14
5.2.3 Access Node (AN) 14
5.2.4 Call Management Server (CMS) and Gate Controller (GC) 14
5.2.5 Record Keeping Server (RKS) 14
5.3 IPCablecom Dynamic QoS Architecture 14
5.4 QoS Interfaces 15
5.5 Framework for IPCablecom QoS 17
5.6 Requirements of Access Network Resource Management 19
5.6.1 Preventing theft of service 19
5.6.2 Two-phase Resource Commitment 19
5.6.3 Segmented Resource Assignment 20
5.6.4 Resource Changes During a Session 20
5.6.5 Dynamic Binding of Resources 20
5.6.6 Dynamic QoS Performance 20
5.6.7 Session Class 21
5.6.8 Intermediate Network Support 21
5.6.9 Backbone QoS Support 21
5.7 Theory of Operation 21
5.7.1 Basic Session Set-up 21
5.7.2 Gate Co-ordination 22
5.7.3 Changing the Packet Classifiers Associated With a Gate 22
5.7.4 Session Resources 23
5.7.5 Admission Control and Session Classes 23
5.7.6 Resource Renegotiations 24
5.7.7 Dynamic Binding of Resources (Re-reserve) 24
5.7.8 Support For Billing 24
5.7.9 Backbone Resource Management 25
5.7.10 Setting the DiffServ Code Point 25
6 MTA to AN Quality-of-Service Protocol (pkt-q3) 25
6.1 RSVP Extensions Overview 26
6.1.1 Segmented Operation 26
6.1.2 Bidirectional Reservations 26
6.1.3 Header Compression, Suppression and VAD 26
6.1.4 Dynamic Binding of Resources 27
6.1.5 Two-Stage Reserve/Commit Process 28
6.1.6 Authentication 28
6.2 RSVP Flowspecs 29
6.3 Definition of Additional RSVP objects 30
6.3.1 Reverse-Rspec 30

6.3.2 Reverse-Session 31
6.3.3 Reverse-Sender-Template 31
6.3.4 Reverse-Sender-Tspec 31
6.3.5 Forward-Rspec 32
6.3.6 Component-Tspec 33
6.3.7 Resource-ID 33
6.3.8 Gate-ID 34
6.3.9 Commit-Entity 34
6.3.10 DClass 34
6.4 Definition of RSVP Messages 34
6.4.1 Message Objects for Upstream Reservation 35
6.4.2 Message Objects for Downstream Reservation 36
6.4.3 Message Objects for Support of Multiple Flowspecs 36
6.5 Reservation Operation 36
6.5.1 Reservation Establishment 37
6.5.2 Reservation Change 39
6.5.3 Reservation Deletion 39
6.5.4 Reservation Maintenance 40
6.6 Definition of Commit Messages 41
6.7 Commit Operations 42

7 Authorization Interface Description (PKT-Q6) 42
7.1 Gates: the Framework for QoS Control 42
7.1.1 Classifier 43
7.1.2 Gate 43
7.1.3 Gate Identification 44
7.1.4 Gate Transition Diagram 45
7.1.5 Gate Co-ordination 46
7.2 COPS Profile for IPCablecom 48
7.3 Gate Control Protocol Message Formats 49
7.3.1 COPS Common Message Format 49
7.3.2 Additional COPS Objects for Gate Control 50
7.3.2.1 Transaction-ID 50
7.3.2.2 Subscriber-ID 51
7.3.2.3 Gate-ID 51
7.3.2.4 Activity-Count 51
7.3.2.5 Gate-spec 52
7.3.2.6 Remote-Gate-Info 53
7.3.2.7 Event-Generation-Info 54
7.3.2.8 Media-Connection-Event-Info 55
7.3.2.9 IPCablecom-Error 55
7.3.2.10 Electronic-Surveillance-Parameters 56
7.3.2.11 Session-Description-Parameters 56
7.3.2.12 Gate-Co-ordination-Port 56
7.3.3 Definition of Gate Control Messages 57
7.4 Gate Control Protocol Operation 58
7.4.1 Initialization Sequence 58
7.4.2 Operation Sequence 59
7.4.3 Procedures for Allocating a new Gate 59
7.4.4 Procedures for Authorizing Resources Through a Gate 60
7.4.5 Procedures for Querying a Gate 61
7.4.6 Procedures for Deleting a Gate 61
7.4.7 Termination Sequence 62

8 Gate-to-Gate Co-ordination Interface (PKT-Q8) 62
8.1 Gate-to-Gate Protocol Messages 63
8.1.1 GATE-OPEN 65
8.1.2 GATE-OPEN-ACK 66
8.1.3 GATE-OPEN-ERR 66
8.1.4 GATE-CLOSE 66
8.1.5 GATE-CLOSE-ACK 66
8.1.6 GATE-CLOSE-ERR 66

8.2 Gate Co-ordination Procedures 67
8.2.1 Example Procedures for end-to-end Gate Co-ordination 67
8.2.2 Example Procedures for Proxied Gate Co-ordination 68

Annex A (normative): Additional requirements for J.112 annex A implementations 70

A.1 Terminology 70

A.2 Mapping of Flowspecs into J.112 QoS parameters 70

A.3 Use of J.112 MAC Primitives 72
A.3.1 Reserving Resources 72
A.3.2 Committing Resources 72
A.3.3 Releasing Resources 73

A.4 Support of Two-phase Resource Allocation 73

A.5 Reservation Maintenance 77

Annex B (normative): Additional requirements for J.112 annex B and annex C implementations 78

B.1 Mapping Flowspecs into J.112 QoS Parameters 78

B.2 J.112 Support for Resource Reservation 79
B.2.1 Two-Phase QoS Reservation/Commit 80
B.2.2 Reservation with Multiple Service Flow Specifications 82
B.2.3 Reservation Maintenance 83
B.2.4 Support for Dynamic Binding of Resources 84
B.2.5 QoS Parameter Mapping for Authorization 84
B.2.6 Automatically-Committed Resources 84

B.3 Use of J.112 MAC Control Service Interface 84
B.3.1 Reservation Establishment 85
B.3.2 Reservation Change 85
B.3.3 Reservation Deletion 86
B.3.4 Mapping RSVP Flowspecs into J.112 QoS Parameters 86
B.3.4.1 Example of Mapping 87
B.3.4.2 Payload Header Suppression and VAD 88

Annex C (normative): Timer Definitions and Values 90

Annex D (informative): Sample mapping of SDP descriptions into RSVP flowspecs 92

Annex E (informative): Sample Protocol Message Exchanges for Basic DCS On-Net to On-Net Call for Standalone MTA 94

E.1 Example Call Flow with J.112 annex A messages 95

E.2 Example Call Flow with J.112 annex B/annex C messages 109

Annex F (informative): Sample Protocol Message Exchanges for Basic NCS On-Net to On-Net Call for Standalone MTA 125

F.1 Example Call Flow with J.112 annex A messages 126

F.2 Example Call Flow with J.112 annex B/annex C messages 139

Annex G (informative): Sample Protocol Message Exchanges for Mid-Call Codec Change 153

G.1 Example Call Flow with J.112 annex A messages 153

G.2 Example Call Flow with J.112 annex B/annex C messages 154

Annex H (informative): Sample Protocol Message Exchanges for Call Hold 165

H.1 Example Call Flow with J.112 annex A messages 165

H.2 Example Call Flow with J.112 annex B/annex C messages 167

Annex I (informative): Sample Protocol Message Exchanges for Call Waiting 171

I.1 Example Call Flow with J.112 annex A messages 171

I.2 Example Call Flow with J.112 annex B/annex C messages 171

Annex J (informative): Sample Protocol Message Exchanges for Basic DCS On-Net to On-Net Call of an Embedded MTA 179

J.1 Example Call Flow with J.112 annex A messages 179

J.2 Example Call Flow with J.112 annex B/annex C messages 190

Annex K (informative): Sample Protocol Message Exchanges for Basic NCS Call for Embedded MTA 203

K.1 Example Call Flow with J.112 annex A messages 203

K.2 Example Call Flow with J.112 annex B/annex C messages 215

Annex L (informative): Theft of Service Scenarios 230

L.1 Scenario No. 1: Customers establishing high QoS Connections themselves 230

L.2 Scenario No. 2: Customers using provisioned QoS for non-voice applications 230

L.3 Scenario No. 3: MTA non-co-operation for billing 231

L.4 Scenario No. 4: MTA altering the destination address in voice packets 231

L.5 Scenario No. 5: Use of half-connections 231

L.6 Scenario No. 6: Early termination leaving a half-connection 231

L.7 Scenario No. 7: Forged Gate Co-ordination messages 232

L.8 Scenario No. 8: Fraud directed against unwanted callers 232

Annex M (informative): COPS (Common Open Policy Service) 233

M.1 COPS Procedures and Principles 233

Annex N (informative): RSVP (Resource Reservation Protocol) 235

N.1 RSVP Procedures and Principles 235

Annex O (informative): TCP Considerations 237

O.1 Requirements 237

O.2 Recommended Changes 237

O.3 TCP Connection Establishment impacting Post-dial Delay 238

O.4 Need Low Latency for packets between the GC and AN, even under loss 239

O.5 Head of Line Blocking 239

O.6 TCP Slow Start 240

O.7 Delaying of packets: Nagle's Algorithm 240

O.8 Non-Blocking Interface 240

Annex P (informative): Bibliography 241

History 242

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://www.etsi.org/ipr).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Access and Terminals (AT).

The present document is part 5 of a multi-part deliverable supporting real-time multimedia services, as identified below:

Part 1: "General";

Part 2: "Architectural framework for the delivery of time critical services over cable Television networks using cable modems";

Part 3: "Audio Codec Requirements for the Provision of Bi-Directional Audio Service over Cable Television Networks using Cable Modems";

Part 4: "Network Call signalling Protocol";

Part 5: "Dynamic Quality of Service for the Provision of Real Time Services over Cable Television Networks using Cable Modems";

Part 6: "Media Terminal Adapter (MTA) device provisioning";

Part 7: "Management Information Base (MIB) Framework";

Part 8: "Media Terminal Adapter (MTA) Management Information Base (MIB)";

Part 9: "Network Call Signalling (NCS) MIB Requirements";

Part 10: "Event Message Requirements for the Provision of Real Time Services over Cable Television Networks using Cable Modems";

Part 11: "Security";

Part 12: "Internet Signalling Transport Protocol";

Part 13: "Trunking Gateway Control Protocol";

Part 14: "Operation System Support".

NOTE 1: The above list is complete for the first version of this Technical Specification (TS) (V1.1.1 2001-08). Additional parts are being proposed and these will be added to the list in future versions.

The present part is part 5 of the above mentioned series of ETSI deliverables and addresses requirements for a client device to obtain access to network resources. In particular, it specifies a comprehensive mechanism for a client device to request a specific Quality of Service from the J.112 network. Extensive examples illustrate the use of the specification. The scope of the present document is to define the QoS Architecture for the "Access" portion of the IPCablecom network, provided to requesting applications on a per-flow basis.

NOTE 2: The choice of a multi-part format for this deliverable is to facilitate maintenance and future enhancements.

NOTE 3: The term **MUST** or **MUST NOT** is used as a convention in the present document part to denote an absolutely mandatory aspect of the specification.

Introduction

The cable industry in Europe and across other Global regions have already deployed broadband cable television hybrid fibre coax (HFC) data networks running the Cable Modem Protocol. The cable industry is in the rapid stages of deploying IP Voice and other time critical multimedia services over these broadband cable television networks.

The cable industry has recognized the urgent need to develop ETSI Technical Specifications aimed at developing interoperable interface specifications and mechanisms for the delivery of end to end advanced real time IP multimedia time critical services over bi-directional broadband cable networks.

IPCablecom is a set of protocols and associated element functional requirements developed to deliver Quality-of-Service (QoS) enhanced secure IP multimedia time critical communications services using packetized data transmission technology to a consumer's home over the broadband cable television Hybrid Fibre/Coaxial (HFC) data network running the Cable Modem protocol. IPCablecom utilizes a network superstructure that overlays the two-way data-ready cable television network. While the initial service offerings in the IPCablecom product line are anticipated to be Packet Voice, the long-term project vision encompasses packet video and a large family of other packet-based services.

The cable industry is a global market and therefore the ETSI standards are developed to align with standards either already developed or under development in other regions. The ETSI Specifications are consistent with the CableLabs/PacketCable set of specifications as published by the SCTE. An agreement has been established between ETSI and SCTE in the US to ensure, where appropriate, that the release of PacketCable and IPCablecom set of specifications are aligned and to avoid unnecessary duplication. The set of IPCablecom ETSI specifications also refers to ITU-SG9 draft and published recommendations relating to IP Cable Communication.

The whole set of multi-part ETSI deliverables to which the present document belongs specify a Cable Communication Service for the delivery of IP Multimedia Time Critical Services over a HFC Broadband Cable Network to the consumers home cable telecom terminal. 'IPCablecom' also refers to the ETSI working group program that shall define and develop these ETSI deliverables.

Many cable television operators are upgrading their facilities to provide two way capability and using this capability to provide high speed IP data services per ITU-T Recommendations J.83 and J.112. These operators now want to expand the capability of this delivery platform to include telephony. The present document is one of a series of documents required to achieve this goal. It provides for the dynamic quality of service needed in many real time applications.

1 Scope

The present set of documents specify IPCablecom, a set of protocols and associated element functional requirements. These have been developed to deliver Quality-of-Service (QoS), enhanced secure IP multimedia time critical communication services, using packetized data transmission technology to a consumer's home over a cable television Hybrid Fibre/Coaxial (HFC) data network.

NOTE 1: IPCablecom set of documents utilize a network superstructure that overlays the two-way data-ready cable television network, e.g. as specified within ES 201 488 and ES 200 800.

While the initial service offerings in the IPCablecom product line are anticipated to be Packet Voice and Packet Video, the long-term project vision encompasses a large family of packet-based services. This may require in the future, not only careful maintenance control, but also an extension of the present set of documents.

NOTE 2: The present set of documents aims for global acceptance and applicability. It is therefore developed in alignment with standards either already existing or under development in other regions and in International Telecommunications Union (ITU).

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

ITU-T Recommendation G.711: "Pulse code modulation (PCM) of voice frequencies".

ITU-T Recommendation G.726: "Extensions of Recommendation G.726 for use with uniform-quantized input and output".

ITU-T Recommendation J.83 (1997): "Digital multi-programme systems for television, sound and data services for cable distribution".

ITU-T Recommendation J.112: "Transmission systems for interactive cable television services".

RFC 1321 (1992): The MD5 Message-Digest Algorithm".

RFC 1890 (1996): "RTP Profile for Audio and Video Conferences with Minimal control".

RFC 2113: "IP Router Alert Option".

RFC 2138 (1997): "Remote Authentication Dial In User Service (RADIUS)".

RFC 2205 (1997): "Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification". (Updated by RFC 2750.)

RFC 2210 (1997): "The Use of RSVP with IETF Integrated Services".

RFC 2327 (1998): "SDP Session Description Protocol".

RFC 2474 (1998): "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Header".

RFC 2543: "Session Initiation Protocol (SIP)".

RFC 2748 (2000): "The COPS (Common Open Policy Service) Protocol".

ETSI ES 200 800: "Digital Video Broadcasting (DVB); DVB interaction channel for Cable TV distribution systems (CATV)".

ETSI ES 201 488: "Data-Over-Cable Service Interface Specifications Radio Frequency Interface Specification".

ETSI TS 101 909-2: "Access and Terminals (AT); Digital Broadband Cable Access to the Public Telecommunications Network; IP Multimedia Time Critical Services; Part 2: Architectural framework for the delivery of time critical services over cable Television networks using cable modems".

ETSI TS 101 909-3: "Access and Terminals (AT); Digital Broadband Cable Access to the Public Telecommunications Network; IP Multimedia Time Critical Services; Part 3: Audio Codec Requirements For The Provision Of Bi-Directional Audio Service Over Cable Television Networks Using Cable Modems".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

Access Node: layer two termination device that terminates the network end of the ITU-T Recommendation J.112 connection

Cable Modem: cable modem is a layer two termination device that terminates the customer end of the J.112 connection

NOTE: It is technology specific. In ITU-T Recommendation J.112, annex A, it is called the INA while in annex B it is the CMTS.

IPCablecom: ETSI working group project that includes an architecture and a series of Technical Specifications that enable the delivery of real time services over the cable television networks using cable modems

J.112 Flow: unidirectional or bidirectional flow of data packets that is subject to MAC-layer signalling and QoS assignment compliant to ITU-T Recommendation J.112

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AN	Access Node
CM	Cable Modem
COPS	Common Open Policy Service
CPE	Customer Premise Equipment
DCS	Distributed Call Signalling
INA	Interactive Network Adapter
IP	Internet Protocol
MTA	Media Terminal Adaptor
NCS	Network-based Call Signalling
PHS	Payload Header Suppression
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RAP	Resource Allocation Protocol
RSVP	Resource reSerVation Protocol
TLV	Type-Length-Value
VAD	Voice Activity Detection

4 Void

5 Technical Overview

Enhanced Quality of Service is required for supporting interactive multimedia applications. Resources may be constrained in segments of the network, requiring allocation of resources in the network. The scope of the present document is to define the Quality of Service Architecture for the "Access" portion of the IPCablecom network. The access portion of the network is defined to be between the Multimedia Terminal Adapter (MTA) and the Access Node (AN), including the J.112 network. The present document also recognizes that per-flow reservations may be required within the customer premises, and the protocols developed herein address this potential need. Although some segments of the backbone network may require resource reservation to provide adequate quality of service, we consider the protocols for backbone resource management to be outside the scope of the present document.

Resources are allocated on the J.112 network for individual flows associated with each session of an application, per subscriber, on an authorized and authenticated basis. A DQoS session, or simply a session, is defined by the present document to be a single bidirectional data flow between two clients. When a multimedia application needs multiple bidirectional data flows (e.g. one for voice and a separate for video), separate DQoS sessions are established for each. Applications may use only half of the session's bidirectional data flow, thereby providing send-only or receive-only services. For example, in a typical voice communications application, a simple communication between two parties is implemented by a single session, while complex, multiparty communications (e.g. "conference calls") are implemented by multiple simultaneous sessions.

Two IPCablecom Call Signalling protocols are being defined - Network-based Call Signalling (TS 101 909-4) and Distributed Call Signalling (RFC 2543 SIP). This Dynamic QoS specification is the underlying QoS framework for both of these call signalling protocols. QoS is allocated for flows associated with a session in concert with the signalling protocol.

The present document introduces the concept of a segment-by-segment QoS framework. It exploits the information available from signalling protocols to perform the QoS assignment on both the "local" segment (on the J.112 network close to the originating party) and the "remote" segment (the J.112 network close to the terminating party). Thus, the present document allows different providers to use the most appropriate mechanisms for the segment that they are managing. Using a concatenation of the segments with QoS, we provide end-to-end QoS assurance for the session.

The Dynamic QoS specification incorporates protocols to enable providers of packet-based voice communications using the IPCablecom framework to use different charging models, including both flat-rate charging as well as usage-based charging. It is the intent of the present document to ensure that enhanced QoS is provided only to authorized and authenticated users. The specific techniques used for authorizing and authenticating a user are beyond the scope of the present document.

This Dynamic QoS specification recognizes the requirements of a commercially viable voice communications service analogous to that offered by means of the public switched telephone network. It is important to ensure that resources are available before the two parties involved in the session are invited to communicate. Thus, resources are reserved before the recipient of the communication is notified that someone is trying to initiate a communication. If there are insufficient resources for a session, then the session is blocked.

The protocols developed in the present document explicitly recognize the need to ensure that there is no potential for fraud or theft of service by endpoints that do not wish to co-operate with the call signalling and QoS signalling protocols with the intent of avoiding being charged for usage. The present document introduces the concept of a two-phase activation for resource reservation (reserve and commit). The two phases allow a provider to both allocate resources only when they are required (when the voice path is cut-through) which may be used for billing. Further, because the second phase to commit resources requires an explicit request from the MTA, it enables the provider to prevent fraud and theft of service.

5.1 IPCablecom QoS Architecture Requirements

The following list presents the QoS requirements for supporting multimedia applications over IPCablecom Networks.

1) Provide IPCablecom accounting for the QoS resources on a per-session basis

It is anticipated that, from a billing perspective, one of the resources that will need to be accounted for is the use of QoS in the J.112 network. Thus, information needs to be identified and tracked that allows reconciliation of the use of the J.112 QoS resource with IPCablecom session activity.

2) Both two-phase (reserve-commit) and single-phase (commit) QoS activation models

Under application control it should be possible to utilize either a two-phase or single-phase QoS activation model. In the two-phase model the application reserves the resource, then later commits it. In the single-phase model both reservation and commitment occur as a single autonomous operation. As in the J.112 model, resources that are reserved but not yet committed are available for temporary assignment to other (e.g. best effort) J.112 Flows. The current specification provides mechanisms for both two-phase and single-phase activation for embedded MTAs, and for two-phase activation for standalone MTAs. Single-phase activation for standalone MTAs is deferred to later releases of the present document.

3) Provide IPCablecom defined policies to control QoS in both the J.112 network and the IP backbone

It should be possible for different types of sessions to have different QoS characteristics. For example, sessions within a single CABLE OPERATOR provider's domain may receive different QoS than sessions outside the domain (e.g. international sessions including links to the PSTN). This dynamic QoS specification may allow a CABLE OPERATOR to provide different QoS for different types of customers (e.g. higher QoS for subscribers of a business service at certain times of the day compared to residential customers), or different types of applications for a single customer.

4) Prevent (minimize) abusive QoS usage

Two types of abusive QoS usage are identified: that which is accurately billed but leads to denying service to others, and that which is not accurately billed and leads to theft of service. Subscriber applications and IPCablecom applications (either embedded or PC-based) may inadvertently or intentionally abuse their QoS privileges (e.g. use of enhanced QoS, which the provider wants limited to voice applications, by an FTP application). Even though the J.112 network is expected to enforce a subscriber's access to QoS, rich packet classification and signalling control mechanisms should exist to keep the subscriber (and the subscriber devices) from fraudulent use of QoS. Admission control procedures should be employed to reduce denial-of-service attacks.

5) Provide admission control mechanisms for both upstream and downstream directions in the J.112 network

Both upstream and downstream QoS should be subject to per-session admission control.

6) Use QoS mechanism of the J.112 MAC layer

It should be possible to police (defined as marking, dropping, or delaying packets) all aspects of QoS defined in the service at the AN using the J.112 QoS mechanisms. Furthermore, it should be possible to support multiple flow mapping models - associate a single IPCablecom session to a single J.112 Flow and multiple IPCablecom sessions to a single J.112 Flow.

7) Policy is enforced by the AN

Ultimate policy control is entrusted to the AN. The philosophy is that any client can make any QoS request, but the AN (or an entity behind the AN) is the only entity entrusted to grant or deny QoS requests.

8) IPCablecom entities must be as unaware as possible of specific J.112 QoS primitives and parameters

For IPCablecom, like any other application that uses the IP-network, the design objective is to minimize the amount of access-link-specific knowledge contained within the application layer. The less access-link knowledge in the application layer, the more applications will be available for development and deployment, and the fewer testing and support-problems will be encountered.

9) Reclamation of QoS resources for dead/stale sessions

It is necessary to reclaim and re-allocate precious QoS resources for sessions that are no longer active, but have not been properly torn down. There should be no resource "leaks" in the J.112 link. For example, if an IPCablecom client module malfunctions in the midst of an IPCablecom session, all J.112 QoS resources used by the session should be released within a reasonable period of time.

10) Dynamic QoS policy changes

It is desirable to dynamically change QoS policies for subscribers. For example, this requirement addresses the ability to change a customer's service level (e.g. upgraded from a "bronze" service to a "gold" service) on-the-fly without resetting the CM.

11) Absolute minimum session set-up latency time and post pick-up delay

The IPCablecom Network should allow for emulation and enhancement of the PSTN experience to the user, and should be equally good, if not better, in session set-up and post pick-up delay metrics.

12) Multiple concurrent sessions

It is desirable to allocate QoS resources (e.g. bandwidth) for not only individual point-to-point sessions, but also for multiple point-to-point sessions (e.g. conference calls, combined audio/video calls).

13) Dynamic adjustment of QoS parameters in the middle of IPCablecom sessions

It should be possible for the IPCablecom service to change QoS mid-session, e.g. network-wide resource adjustments or creation of compatible CODEC parameters (necessitating QoS changes), or user defined feature to vary QoS levels, or detection of fax or modem streams (necessitating change from compressing CODEC to G.711).

14) Support multiple QoS control models

Strong cases can be made for both subscriber-side and network-side initiation of QoS signalling. In subscriber side signalling, an application can initiate its request for QoS immediately when the application believes it needs QoS. Also, subscriber side signalling supports application models that are peer-to-peer. In network-side signalling, implementation of the endpoint application can be completely unaware of QoS (especially in the J.112 network). Network-side signalling supports application models that are client-server (with the server being trusted). It is expected that both models will be present in IPCablecom (and other application) networks. The current specification is for subscriber-side signalling only.

15) Support both embedded-MTA and standalone-MTA QoS signalling

It should be possible to signal QoS from both an embedded-MTA and standalone-MTA. In a standalone MTA the only signalling path supported is that specified herein using RSVP. In an embedded MTA, both RSVP and direct access to the J.112 MAC signalling is possible.

5.2 IP QoS Access Network Elements

The following network elements are employed to support QoS for IPCablecom Networks.

5.2.1 Multimedia Terminal Adapter (MTA)

The IPCablecom network client device (i.e. the MTA) can be one of the following devices. These devices reside at the customer site and are connected through the J.112 channel to the network. All MTAs are assumed to implement some multimedia signalling protocol, such as TS 101 909-4. An MTA may be either a device with a standard two-wire telephone set in the MTA-1 configuration, or may add video input/output capabilities in the MTA-2 configuration. It may have minimal capabilities, or may implement this functionality on a multimedia personal computer, and have all of the capabilities of the PC at its disposal.

From the point of view of QoS, there are two types of MTAs.

1) **Embedded/Integrated MTA**. This is a client multimedia terminal which incorporates a J.112 MAC-layer interface to the J.112 network.

2) **Standalone MTA**. This is a Client that implements the multimedia functionality without incorporating a J.112 MAC-layer interface. The standalone MTA will typically use Ethernet, USB, or IEEE 1394 as the physical interconnect to a CM. The standalone MTA may be connected to a customer network, and use transport facilities of the customer network (possibly including intermediate IP routers) to establish sessions over the J.112 network.

5.2.2 Cable Modem (CM)

This is an IPCablecom network element as defined by J.112. The CM is responsible for classifying, policing and marking packets once the traffic flows are established by the signalling protocols described herein.

5.2.3 Access Node (AN)

The Access Node (AN) is the element of the IPCablecom network that contains centralized functions responsible for processing information flows. The AN acts as a Policy Enforcement Point (PEP) per the IETF Resource Allocation Protocol (RAP) Framework.

The AN implements a "IPCablecom Dynamic QoS Gate" (hereafter called just "Gate") between the J.112 network and an IP Backbone. The Gate is implemented using the packet classification and filtering functions defined in J.112.

The AN may or may not also be configured as an "IS-DS Boundary" entity. An IS-DS Boundary interfaces to an inter-network using the Integrated Services (Intserv) model of QoS control and some other model, e.g. Differentiated Services (Diffserv).

5.2.4 Call Management Server (CMS) and Gate Controller (GC)

The IPCablecom Call Management Server (CMS) entity performs services that permit MTAs to establish Multimedia sessions (including voice communications applications such as "IP telephony" or "VoIP"). A CMS using the Network-Controlled call signalling model implements a Call Agent that directly controls the session, and maintains per-call state. A CMS using the Distributed Call signalling model may serve as a "DCS Proxy" and perform services only during initial session set-up. The term Gate Controller (GC) is used to refer to the portion of either type of CMS that performs the Quality of Service related functions.

In the IPCablecom Dynamic QoS Model, the Gate Controller controls the operation of the Gates implemented on an AN. The GC acts as a Policy Decision Point (PDP) per the IETF Resource Allocation Protocol (RAP) Framework.

5.2.5 Record Keeping Server (RKS)

The Record Keeping Server is a IPCablecom network element that only receives information from IPCablecom elements described in the present document. The RKS can be used as a billing server, diagnostic tool, etc.

5.3 IPCablecom Dynamic QoS Architecture

The IPCablecom QoS architecture is based upon J.112, IETF RSVP, and IETF Integrated Services Guaranteed QoS.

Specifically, the IPCablecom QoS architecture uses the protocol as defined in J.112 within the cable television network. These messages support static and dynamic installation of packet classifiers (i.e. Filter-Specs) and flow scheduling (i.e. flow specs) mechanisms to deliver enhanced quality of service. J.112 QoS is based upon the objects which describe traffic and flow specifications, similar to the TSPEC and RSPEC objects as defined in the IETF Resource reSerVation Protocol (RSVP). This allows QoS resource reservations to be defined on a per flow basis.

In the J.112 QoS architecture, J.112 Flows are considered to be either unidirectional or bidirectional. In each direction, J.112 Flows are subject to the operations shown below.

The CM, where traffic enters the QoS enabled J.112 network, is responsible for:

- Classification of IP traffic into J.112 Flows based on defined filter specifications.
- Performing traffic shaping and policing as required by the flow specification.
- Maintaining state for active flows.
- Altering the TOS field in the upstream IP headers based on the network operator's policy.
- Obtaining the required J.112 QoS from the AN.
- Applying J.112 QoS mechanisms appropriately.

The AN is responsible for:

- Providing the required QoS to the CM based upon policy configuration.
- Allocating upstream bandwidth in accordance to CM requests and network QoS policies.
- Classifying each arriving packet from the network side interface and assigning it to a QoS level based on defined filter specifications.
- Policing the TOS field in received packets from the J.112 network to enforce TOS field settings per network operator policy.
- Altering the TOS field in the downstream IP headers based on the network operator's policy.
- Performing traffic shaping and policing as required by the flow specification.
- Forwarding downstream packets to the J.112 network using the assigned QoS.
- Forwarding upstream packets to the backbone network devices using the assigned QoS.
- Maintaining state for active flows.

The backbone network may either utilize IETF Integrated Services based mechanisms or use IETF Differentiated Services mechanisms. In a Diffserv backbone, network routers forward a packet, providing the appropriate IETF QoS, based on the setting of the TOS field. In a Diffserv backbone, no per-flow state is required in the core network devices.

5.4 QoS Interfaces

Quality of service signalling interfaces are defined between many of the components of the IPCablecom network as shown in figure 1. Signalling involves communication of QoS requirements at the application layer (e.g. SDP parameters), network layer (e.g. RSVP), and at the data-link layer (e.g. J.112 QoS). Also, the requirement for policy enforcement and system linkages between the OSS subscriber provisioning, admission control within the managed IP backbone, and admission control within the J.112 network creates the need for additional interfaces between components in the IPCablecom network.

An expanded explanation of QoS architecture framework is contained in the IPCablecom Architecture Framework - J.arch, and is shown in figure 1.

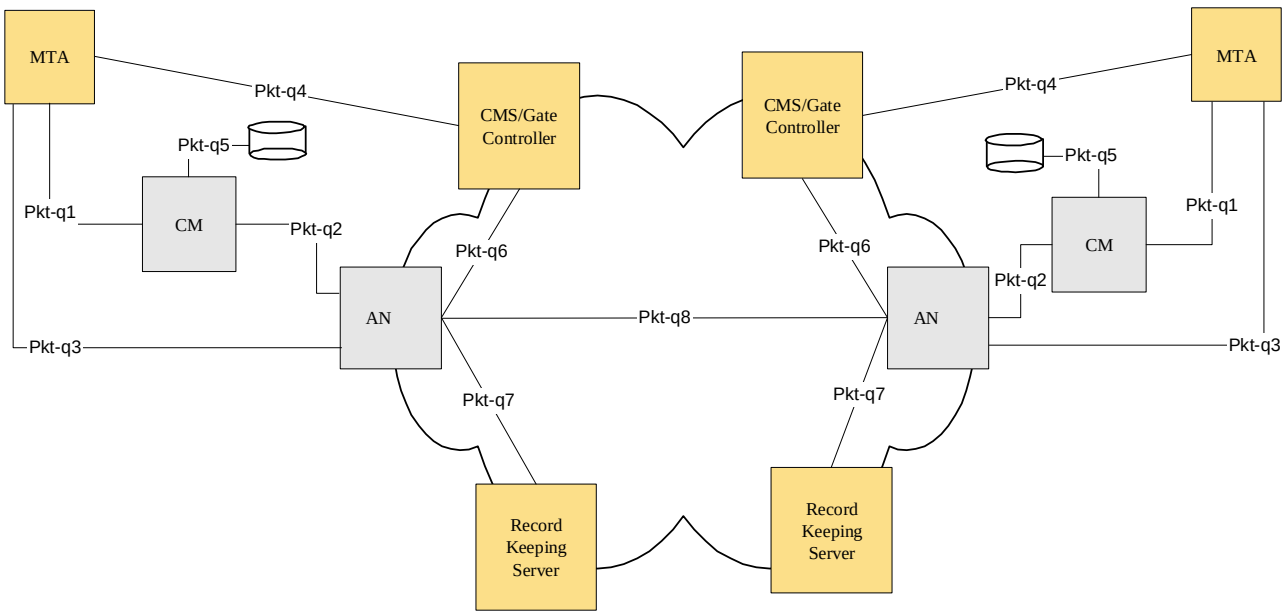


Figure 1: QoS Signalling Interfaces in IPCablecom Network

Interfaces pkt-q1 through pkt-q8 are available for controlling and processing QoS. Not all interfaces are used in all configurations and protocol variations. All but the pkt-q5 interface are utilized by DQoS. The following table briefly identifies each interface and how each interface is used in this Dynamic QoS Specification (DQoS). Two alternatives are shown for the present document: first a general interface that is applicable to either embedded or standalone MTAs; and second, an optional interface that is available only to embedded MTAs.

Table 1: DQoS Interfaces

Interface	Description	DQoS Embedded/ Standalone MTA	DQoS Embedded MTA (optional)
pkt-q1	MTA - CM	N/A	J.112 MAC-layer interface
pkt-q2	CM - AN	J.112 QoS, AN-initiated	J.112 QoS, CM-initiated
pkt-q3	MTA - AN	RSVP+	N/A
pkt-q4	MTA - GC/CMS	NCS/DCS	NCS/DCS
pkt-q5	CM - Provisioning Server	N/A	N/A
pkt-q6	GC - AN	Gate Management	Gate Management
pkt-q7	AN - RKS	Billing	Billing
pkt-q8	AN - AN	Gate Management	Gate Management

pkt-q1: Interface between the MTA and CM

This interface is only defined for the embedded MTA. The interface decomposes into three sub-interfaces:

- Control: used to manage J.112 Flows and their associated QoS traffic parameters and classification rules.
- Synchronization: used to synchronize packetization and scheduling for minimizing latency and jitter.
- Transport: used to process packets in the media stream and perform appropriate per-packet QoS processing.

This interface is conceptually defined in ITU-T Recommendation J.112. For standalone MTAs no instance of this interface is defined.

pkt-q2: J.112 QoS Interface between CM and AN

This is the J.112 QoS interface (control, scheduling and transport). Control functions can be initiated from either the CM or the AN. However the AN is the final policy arbiter and granter of resources by performing admission control for the J.112 network. This interface is defined in ITU-T Recommendation J.112.

pkt-q3: Network Layer Interface between the MTA and AN

The interface is used to request bandwidth, and QoS in terms of delay using standard RSVP and extensions specified herein. As a result of message exchanges between the MTA and AN, J.112 Flows are activated using AN-originated signalling on interface pkt-q2.

pkt-q4: Application Layer signalling between GC/CMS and MTA

Many parameters are signalled across this interface such as the media stream, IP addresses, port numbers, and the selection of Codec and packetization characteristics. DCS and NCS are two examples of application layer signalling.

pkt-q5: Signalling from the J.112/IPCablecom Provisioning to the CM

This interface is not utilized for QoS signalling in DQoS.

pkt-q6: Interface between the GC/CMS and AN

This interface is used to manage the dynamic Gates for media stream sessions. This interface enables the IPCablecom network to request and authorize QoS. With respect to admission and authorization, in the context of IPCablecom, a trust relationship must exist between the GC/CMS and AN.

pkt-q7: AN to Record Keeping Server

This interface is used by the AN to signal to the RKS all changes in session authorization and usage.

pkt-q8: AN-to-AN interface

This interface is used for co-ordination of resources (Gates) between the AN of the local MTA and the AN of the remote MTA. The AN is responsible for the allocation and policing of QoS resources in the J.112 network it is managing.

5.5 Framework for IPCablecom QoS

In order to justify its costs to the end user, a commercial multimedia service (e.g. voice communications capability) may require a high level of transport and signalling performance, including:

- Low delay - end-to-end packet delay needs to be small enough that it does not interfere with normal multimedia interactions. For normal telephony service using the PSTN, the ITU recommends no greater than 300 ms roundtrip delay (see note). Given that the end-to-end backbone propagation delay may absorb a significant amount of this delay budget, it is important to control delay on the access channel, at least for long-distance calls.

NOTE: ITU-T Recommendation G.114 states that a one-way delay of 150 ms is acceptable for most user applications. However, highly interactive voice and data applications may experience degradation even when delays are below 150 ms. Therefore any increase in processing delay (even on connections with transmission times well below 150 ms) should be discouraged unless there are clear service and application benefits.

- Low packet loss - packet loss needs to be small enough so that voice quality or performance of fax and voiceband modems is not perceptibly impaired. While loss concealment algorithms can be used to reproduce intelligible speech even with high loss rates, the resulting performance cannot be considered to be adequate as a replacement for existing circuit-switched telephone service. Loss requirements for acceptable voiceband modem performance are even more stringent than those for voice.

- Short post-dial delay - the delay between the user signalling a connection request and receiving positive confirmation from the network needs to be short enough that users do not perceive a difference from the post-dial delay they are accustomed to in the circuit switched network, or believe that the network has failed. This is of the order of one second.

- Short post pick-up delay - the delay between a user picking up a ringing phone and the voice path being cut through needs to be short enough so that the "hello" is not clipped. This should be less than a few hundred milliseconds (ideally less than 100 ms).

A key contribution of the Dynamic QoS framework is a recognition of the need for co-ordination between signalling, which controls access to application specific services, and resource management, which controls access to network-layer resources. This co-ordination provides a number of critical functions. It ensures that users are authenticated and authorized before receiving access to the enhanced QoS associated with the service. It ensures that network resources are available end-to-end before alerting the destination MTA. Finally, it ensures that the use of resources is properly accounted for, consistent with the conventions of traditional voice-grade telephone service (to which some IPCablecom services are similar from a customer perspective) in which charging occurs only after the party receiving a communication picks up.

In order to support the above requirements, the QoS protocols assure that all resources are committed to all transport segments before the signalling protocols cause alerting of the destination. Likewise, during tear down of a session, the QoS protocols include measures to assure that all resources dedicated exclusively to the session are released. Without this co-ordination between the two directions of data flows, it would be possible for users to thwart the QoS controls and obtain free service. For example, if the paying client terminates the session, but the non-paying does not, a "half channel" remains that can be used to fraudulently transfer data in one direction. The QoS protocols approximate the "all or nothing" transaction semantics for session creation and destruction.

It is desired that the mechanisms used to implement the session be based on existing standards and practices, and also that the results of this work be usable to support alternative call models. These desires have led to the use of the IETF Real Time Protocol (RTP) to carry multimedia data, carried over the IETF User Datagram Protocol (UDP). In-band signalling to set up Quality of Service is carried out using a superset of the IETF Resource reSerVation Protocol (RSVP).

The QoS architecture should provide support for new emerging applications that are dependent on multicast data delivery. Although this is not a strict requirement in the QoS architecture, providing support for multicast will enable the future development of a rich set of multimedia applications. We have not yet examined whether the resource management enhancements introduced here will support multicast seamlessly or not.

For purposes of managing Quality of Service, the bearer channel for a session is managed as three distinct segments: the access network for the originating side of the session, a backbone network, and the access network for the terminating side of the session. J.112 network resources are managed on the basis of J.112 Flows, using the mechanisms defined in J.112. Backbone resources may be managed either per-flow or, more likely, through an aggregated quality of service mechanism. Management of backbone resources is outside the scope of the present document.

Figure 2 graphically shows this model. The present document accommodates a customer environment where a standalone MTA may be connected to the CM via a network of links and standard RSVP-capable routers.

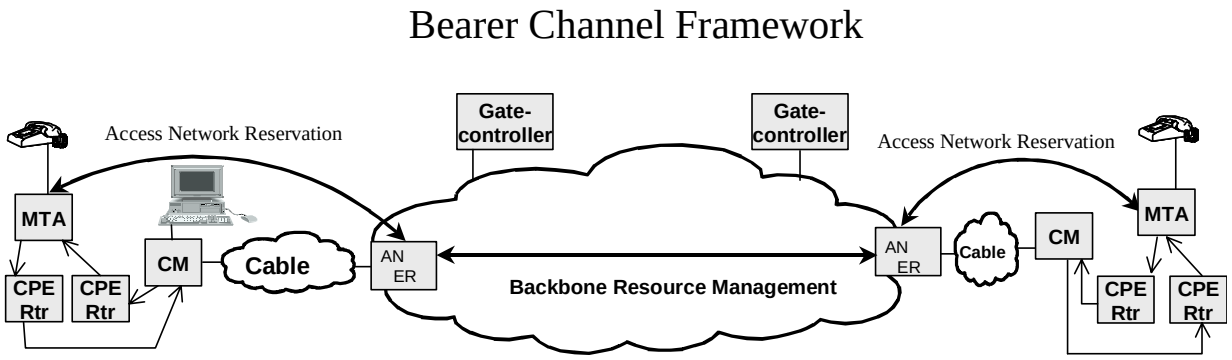


Figure 2: Session Framework

A QoS-defined construct called a *gate* provides a control point for the connection of access networks to high quality backbone service. A gate is implemented by an AN and consists of a packet classifier, a traffic policer, and an interface to an entity that gathers statistics and events (all of these components exist in the J.112 network). A gate can ensure that only those sessions that have been authorized by the service provider receive high quality service. Gates are managed selectively for a flow. For IPCablecom-based voice communications service, they are opened for individual calls. Opening a gate involves an admission control check that is performed when a resource management request is received from the client for an individual session, and it may involve resource reservation in the network for the session if necessary. The upstream packet filter in the gate allows a flow of packets to receive enhanced QoS for a session from a specific IP source address and port number to a specific IP destination address and port number. The downstream packet filter in the gate allows a flow of packets to receive enhanced QoS for a session from a specific IP source address to a specific IP destination address and port number.

A Gate is a logical entity that resides in an AN. A GateID is associated with an individual session and is meaningful at the Gate; the GateID is an identifier that is locally unique at the AN, and is assigned by that AN. A Gate is unidirectional in nature. If a Gate is "Closed", then data going upstream/downstream on the J.112 access network may either be dropped or provided best-effort service. The choice of dropping packets or serving them on a best-effort basis is a policy choice of the provider.

The gate controller is responsible for the policy decision of when and whether the gate should be opened. A gate is established in advance of a resource management request. This allows the policy function, which is at the gate controller, to be "stateless" in that it does not need to know the state of sessions that are already in progress.

While the gate controls the QoS-guaranteed stream, other flows, such as RTCP or signalling messages, are not policed by the gate. These latter flows may be transported on different J.112 Flows in the J.112 network, such as a dedicated signalling link.

5.6 Requirements of Access Network Resource Management

Providing voice communications service over IP networks with the same level of quality as is available over the PSTN imposes bounds on loss and delay metrics for voice packets and requires active resource management in both the access and backbone networks. The service provider needs to be able to control access to network resources, in order to ensure that adequate capacity is available on an end-to-end basis, even under unusual or overload conditions. The service provider may seek additional revenue for providing a voice communications service with these enhanced quality characteristics (i.e. quality beyond that obtained with a "best-effort" service). The mechanisms provided herein for managed access to enhanced QoS enable the service provider to ensure that access is provided only to authorized and authenticated users on a session-by-session basis and there is no theft of that service.

Clients of the service signal their traffic and performance parameters to the "gate" at the network edge, where the network makes an admission control decision based on both resource availability as well as policy information associated with the gate.

In J.112 networks capacity is limited and it is necessary to do resource management on a per-flow basis. In the backbone there may be several alternatives, ranging from per-flow per-hop admission control to coarse-grained resource provisioning. The present document deals only with access network QoS, and is agnostic about backbone network QoS schemes.

This architecture aims to provide a high degree of generality with the intention of enabling new services and future evolution of network architectures. This goal leads to several requirements for a viable QoS architecture, described in the following clauses.

5.6.1 Preventing theft of service

The network resources dedicated to the session are protected from misuse, including:

- Authorization and Security - ensuring that users are authenticated and authorized before receiving access to the enhanced QoS associated with the voice communications service. The CMS/Gate Controller involved in call signalling is trusted to perform these checks and is the only entity which is trusted to create a new gate in an AN. The CMS/GC acts as a policy decision point from the perspective of QoS management.
- Resource control - ensuring that the use of resources is properly accounted for, consistent with the conventions of providers that are part of the PSTN in which charging occurs only after the called party picks up. This includes prevention of utilization of reserved resources for purposes other than the session to which they are assigned. This is achieved through the use of gates and co-ordination between gates, which bind together address filtering mechanisms with resource reservations.

Since this service may be billed on a per-use basis, there is a significant risk of fraud and service theft. The architecture enables the provider to charge for quality of service. Thus, it prevents theft of service scenarios, several of which are described in annex L.

Theft of service scenarios are addressed in this and other documents. They motivate some of the components of the QoS and Call Signalling architectures and protocols.

5.6.2 Two-phase Resource Commitment

A two-phase protocol for resource commitment is essential to a commercial-grade voice communications service, for two reasons unique to the requirements associated with such a service. First, it ensures that resources are available before signalling the party at the far end that a communication is incoming. Secondly, it ensures that usage recording and billing are not started until the far end picks up, which is also the point at which voice may be cut-through. These properties are provided by conventional telephony signalling protocols; we simply wish to emulate the same semantics here. Also, if bandwidth is allocated before the far end picks up, a theft of service becomes possible. Requiring the endpoints to explicitly send a commitment message ensures that usage recording is based on knowledge of the endpoint and its explicit action.

This framework also supports entities, such as announcement servers and PSTN gateways, that need the voice to be cut through after the first phase of the resource management protocol.

5.6.3 Segmented Resource Assignment

The Dynamic QoS Architecture partitions resource management into distinct access and backbone segments. Segmented resource assignment is beneficial for two reasons:

- It allows for different bandwidth provisioning and signalling mechanisms for originator's network, far-end network, and backbone network.
- It allows for resource-poor segments to maintain per-flow reservations and carefully manage resource usage. At the same time, when backbone segments have sufficient resources to manage resources more coarsely, it allows the backbone to avoid keeping per-flow state, and thus enhance scalability.

When the backbone does not require explicit per-flow signalling (such as with a Diffserv backbone), it reduces the time taken to set up a session (minimize post-dial delay) and avoids impacting the voice cut-through time (minimize post-pick-up delay).

It potentially reduces the amount of reservation state that is be stored if the remote client is a PSTN gateway.

After the first phase of call signalling, both clients have completed capability negotiation and know what resources are needed end-to-end. Clients send resource management messages using RSVP that may be interpreted hop-by-hop over the local (i.e. user) and access networks (or, optionally for embedded clients, the J.112 MAC-layer Interface). The AN maps the resource management messages to the resource management protocol used over the backbone (e.g. IETF diffserv). It also maps the resource management message to the resource management protocol used over the access link (i.e. J.112).

5.6.4 Resource Changes During a Session

It is possible to change the resources allocated for a session during the life of the session. This facilitates mid-session changes such as switching from a low-rate voice codec to G.711 when modem tones are detected, and the addition of video data to a session that starts as voice only.

5.6.5 Dynamic Binding of Resources

Dynamic binding of resources (re-reserve) is a requirement to enable efficient use of resources when services such as call waiting are invoked. Abstractly, re-reserving takes bandwidth allocated for a session between a VoIP host and a client and reallocates that same bandwidth to a session with a different client.

It is important to understand the potential danger in de-allocating the session bandwidth, then making a new request for allocation of the new bandwidth. There is a risk of another client using the last remaining bandwidth between the two steps, leaving the original session without an assured quality path. The one-step re-reserve mechanism avoids this, as the bandwidth is not made available to other clients.

5.6.6 Dynamic QoS Performance

QoS messaging takes place in real time while callers wait for services to be activated or changed. Thus, the protocol needs to be fast. The number of messages is minimized, especially the number of messages which transits the backbone, and the number of upstream J.112 messages. On the J.112 network, where there is no possibility that forward and reverse paths will be different, this protocol adds several new objects to RSVP, which enables the AN to reduce latency by acting as a proxy for the far-end client.

RSVP messages, J.112 management messages, and call signalling messages (collectively referred to as signalling messages) are all transported over the J.112 network on a best effort basis. If the CM is also supporting data services, best effort service may be unable to provide the low latency needed for signalling messages. In this situation, the CM MAY be provisioned with a separate J.112 Flow, with enhanced QoS, to carry signalling traffic. This separate J.112 Flow is provisioned in the same manner as other J.112 media streams, and MAY include classifiers such that its presence is transparent to the MTA.

5.6.7 Session Class

Resources may be reserved for different types of services and each service may in turn define different classes of services for its sessions. QoS reservations for sessions designated by the service provider to be of higher priority (e.g. emergency calls) suffer a lower likelihood of blocking than normal sessions. The determination of what session class to assign to a session is performed by the service provider, and is a policy that is exercised by the originating Call Agent/Gate Controller complex at the time the initial session request (e.g. first stage INVITE in the case of SIP RFC 2543) is made.

5.6.8 Intermediate Network Support

The architecture should not prohibit intermediate networks between the MTA or Multimedia host and the CM (e.g. customer network). Although the intermediate network may not fall under the CABLE OPERATOR's administrative domain or responsibility, allocation of bandwidth in the CABLE OPERATOR's J.112 network is possible when an intermediate network exists. It is also desirable to present a solution that transparently allows for the reservation of resources on the intermediate network.

5.6.9 Backbone QoS Support

It is possible that some mechanism for explicitly managing backbone resources will be necessary. The scope of the present document is QoS over the J.112 network, but the architecture provides open, sufficiently general interfaces that are compatible with many of the known backbone QoS mechanisms.

5.7 Theory of Operation

5.7.1 Basic Session Set-up

Resource reservation is partitioned into separate Reserve and Commit phases. At the end of the first phase, resources are reserved but are not yet available to the MTA. At the end of the second phase, resources are made available to the MTA and usage recording is started so that the user can be billed for usage.

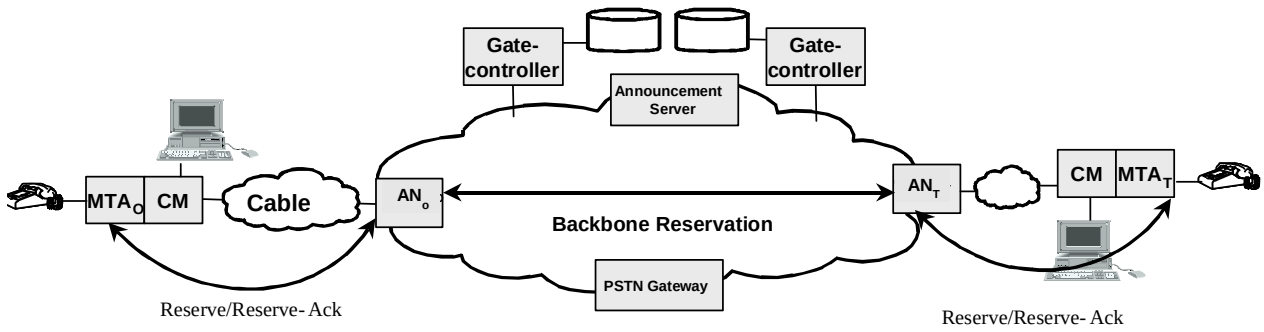


Figure 3: Resource Management Phase 1

Figure 3 shows the first phase of the resource management protocol for a Multimedia application. In this description, subscripts "O" and "T" designate the originating and terminating points of the call. The MTA can be either a standalone VoIP host or an embedded MTA; the latter is shown in figure 3. MTA_O and MTA_T request resource reservation (PATH message in RSVP, or J.112 message in the optional interface for embedded clients) to AN_O and AN_T respectively. AN_O and AN_T perform an admission control check for resource availability (initiating signalling for resource reservation in the backbone if necessary) and send a reply to the respective MTAs. In the RSVP framework, the RESV message from the AN (where the gate resides) is the acknowledgment to the MTA.

Figure 4 shows the second phase. After determining that resources are available, MTA_O sends a RING message to MTA_T instructing it to start ringing the phone. MTA_T sends a RINGING indication to MTA_O indicating both that resources are available and that the RING message was received. When the called party picks up the phone, MTA_T sends an ANSWERED message to MTA_O and a COMMIT message to AN_T. When MTA_O receives the ANSWERED message, MTA_O sends a COMMIT message to AN_O. The COMMIT messages cause resources to be allocated for the call in the J.112 networks. The arrival of the COMMIT messages at AN_T and AN_O causes them to open their gates, and also starts accounting for resource usage. To prevent some theft of service scenarios, the ANs co-ordinate the opening of the gates by exchanging GATE-OPEN messages.

The RING, RINGING, and ANSWERED messages shown in this figure and in the above description are logical equivalents to the call signalling messages exchanged by TS 101 909-4 and SIP RFC 2543.

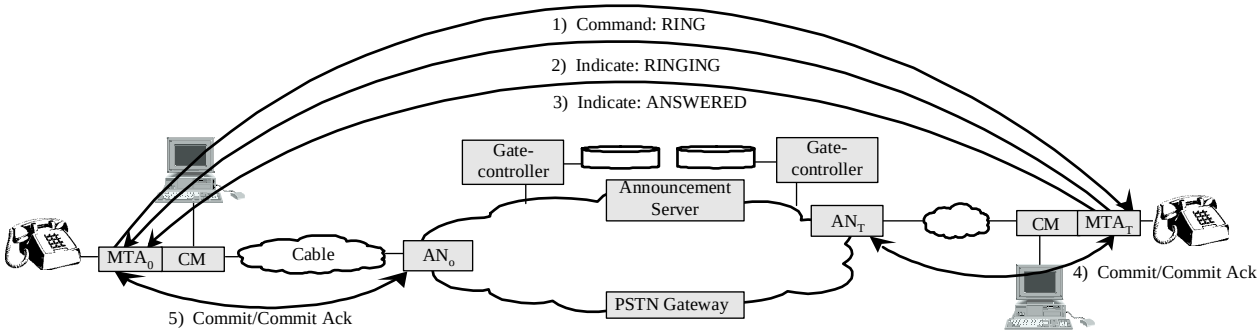


Figure 4: Resource Management Phase 2

5.7.2 Gate Co-ordination

QoS signalling leads to the creation of a gate at each AN associated with a client involved in the session. Each gate maintains usage data for the session and controls whether the packets generated by the associated client receive access to enhanced QoS. Gate co-ordination is needed to prevent fraud and theft of service in situations where a malfunctioning or modified client does not issue the expected signalling messages. It is essential that protocol mechanisms are robust against abuse (see note). A gate co-ordination protocol ensures that:

- A potential for one-way session establishment without billing is avoided. Because the clients may have adequate intelligence and are not trusted, one can envisage the clients establishing two one-way sessions to provide the users with an adequate interactive voice communication channel. Gate co-ordination prevents such sessions being established without the provider being able to charge for them.

- The resources reserved and committed by the two clients are consistent with the results of capability negotiation. If only one client pays for a session, it is important that the resources that are reserved and used are consistent with the expectations of the payer. Gate co-ordination prevents a malicious session recipient from defining session characteristics that will result in an unexpectedly high charge to the originator.

- The gates open and close virtually simultaneously (i.e. within a few hundred milliseconds of each other). Gate co-ordination assures that billing data at the two ends of the session is consistent so that the cost of the session does not depend on which end is paying for it.

NOTE: Several theft of service scenarios are described in annex L.

5.7.3 Changing the Packet Classifiers Associated With a Gate

Once a pair of gates is set up, clients can communicate over the network with enhanced QoS. Several features needed for a commercial voice communications service involve changing the clients involved in a session, for example when a session is transferred or redirected, or during three-way calling. This requires the packet classifiers associated with a gate to be modified to reflect the address of the new client. In addition, changing the endpoints involved in a session may affect how the session is billed. As a result, gates include addressing information for origination and termination points.

5.7.4 Session Resources

The relationship between different categories of resources, authorized, reserved, and committed, is shown in figure 5. A set of resources is represented by an *n*-dimensional space (shown here as two-dimensional) where *n* is the number of parameters (e.g. bandwidth, burst size, jitter, classifiers) needed to describe the resources. The exact procedures for comparing *n*-dimensional resource vectors are given in J.112.

When a session is first established, DQoS protocols authorize the use of some maximum amount of resources, indicated by the outer oval, specifying the authorized resources. When a client makes a reservation for a session, it reserves a certain amount of resources, which are not greater than those for which it has been authorized. When the session is ready to proceed, the client commits to some amount of resources, which are not more than the reserved resources. In many common cases, the committed and reserved resources will be equal. The committed resources represent resources that are currently in use by the active session, whereas reserved resources represent those that are tied up by the client and have been removed from the pool for admission control purposes, but which are not necessarily being used by the client.

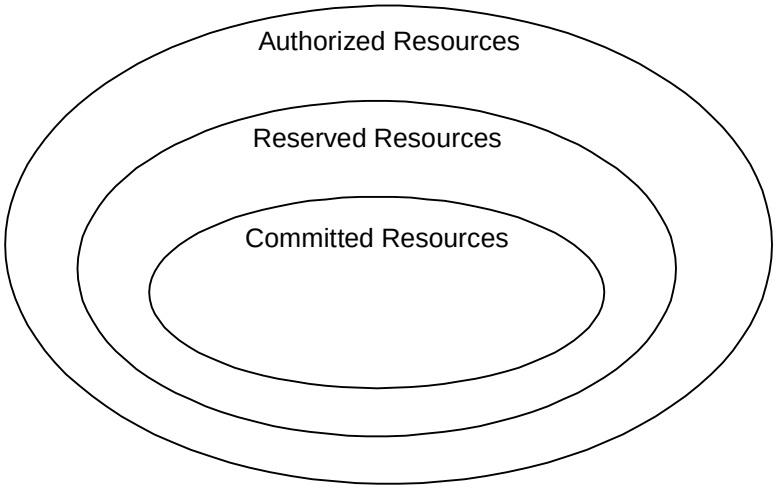


Figure 5: Authorized, Reserved and Committed Resources

Authorizations only affect future resource reservation requests. Resources that have been reserved prior to an authorization change are not affected.

Resources that have been reserved but not committed are available to the system for short-term uses only, such as handling of best-effort data. These resources are not available for other reservations (i.e. overbooking is not allowed). The maximum portion of the available resources that can be reserved at once is a policy decision by the AN, and outside the scope of DQoS.

Excess resources reserved above those committed are released unless the client explicitly requests they be kept through periodic reservation refresh operations. Maintaining such a condition for long periods of time is discouraged, as it reduces the overall capacity of the system. However, there are situations (e.g. call waiting service, where the call on hold requires resources beyond those needed for the active call) where excess reservations are necessary.

5.7.5 Admission Control and Session Classes

It is envisaged that the Gate at the AN may use one or more session classes for resources reserved from an MTA. Session classes define provisionable admission control policies, or their parameters. It is expected that the provider would provision the necessary parameters and/or the alternative admission control policies in the AN and in the Gate Controller. For instance, a session class for normal voice communications, and an overlapping session class for emergency calls could be defined to allow the allocation of up to, respectively, 50 % and 70 % of the total resources to these classes of calls, and leaving the remainder 30-50 % of the total bandwidth available to other, possibly lower priority, services. Session classes may furthermore enable pre-emption of already reserved resources, in which case the policy for such pre-emption would be provisionable by the service provider. When the Authorized Envelope is communicated to the Gate at the AN by the Gate Controller in the Gate-Set message, the Gate Controller includes adequate information to indicate which session class should apply when the corresponding RESERVE request is processed.

5.7.6 Resource Renegotiations

Several of the supported session features require renegotiations of the QoS parameters associated with a session during the lifetime of the session. For example, clients might start communicating using a low-bit-rate audio codec. They can subsequently switch to a higher bit-rate codec or add a video stream, as long as the requested QoS is within the authorized envelope and there is available bandwidth on the network. The use of an authorized QoS envelope that is pre-authorized by the Gate Controller acting as the policy decision point gives clients the flexibility to renegotiate QoS with the network without requiring subsequent Gate Controller involvement. This essentially means that use of resources up to the limits of the envelope is pre-authorized but NOT pre-reserved. Successful allocation of resources within the authorized envelope requires an admission control decision, and is not guaranteed. Subsequent to admission control, the resources are reserved for the flow, although the actual usage of the resources is permitted only after the Commit phase of the Resource Reservation protocol completes. However, no admission control decision is required at the time of commitment of resources. Each change in commitment of resources within the limits of the admission control decision does not require a further reservation. All reservation requests that pass admission control MUST fit within the authorization envelope.

5.7.7 Dynamic Binding of Resources (Re-reserve)

The Dynamic QoS Architecture recognizes that there may be a need to share resources across multiple sessions, especially when resources are in short supply. In particular, when using the call-waiting feature in telephony-like applications, the client may be involved in two simultaneous sessions, but will be active in only one conversation at a time. It is feasible in this case to share the network-layer resources (in particular, on the access link) between the two conversations. Therefore, this architecture allows a set of network layer resources (such as a bandwidth reservation) to be explicitly identified, and allows one or more gates to be associated with those resources. Signalling primitives allow the resources associated with a gate to be *shared* with another gate at the same AN. This improves the efficiency with which resources in the J.112 network are utilized.

When switching back and forth between two sessions in a call-waiting scenario, a client needs to keep enough resources reserved to accommodate either of the sessions, which in general may not need the same amount of resources. Thus, the re-commit operation may change the committed resources. However, the reserved resources do not change in this case, as the client should not have to go through admission control when switching back to the other session.

Whereas the committed resources are always associated with the current active session (and its corresponding IP flow), the reserved resources may be bound to different flows and different gates at different times. A handle, called a resource ID, is used to identify a set of reserved resources for the purpose of binding a flow to those resources.

5.7.8 Support For Billing

QoS signalling can be used to support a broad range of billing models, based on only a stream of event records from the AN. Since the gate is in the data path, and since it participates in resource management interactions with a client, resource usage accounting is done by the gate. The gate in the AN is the appropriate place to do resource accounting, since the AN is directly involved in managing resources provided to a client. It is also important to do usage accounting in the AN to cope with client failures. If a client that is involved in an active session crashes, the AN MUST detect this and stop usage accounting for the session. This can be accomplished using soft state through a resource management refresh message (by having RSVP-PATH messages periodically transmitted for an active session), by monitoring the flow of packets along the data path for continuous-media applications, or by other mechanisms (such as station maintenance) performed by the AN. In addition, since the gate retains state for flows that have been authorized by a service-specific Gate Controller, it is used to hold service-specific information related to charging, such as the account number of the subscriber that will pay for the session. The policy function in the Gate Controller thus becomes stateless.

The support required in the AN is to generate and transmit an event message to a record keeping server on every change to the QoS, as authorized and specified by a gate. Opaque data provided by the Gate Controller that may be relevant to the record keeping server may also be included in the message. Requirements for handling of event records are contained in other Operations Support specifications.

5.7.9 Backbone Resource Management

When an AN receives a resource reservation message from an MTA, it first verifies that adequate upstream and downstream bandwidth is available over the access channel using locally available scheduling information. If this check is successful, the AN can either generate a new backbone resource reservation message, or forward towards the backbone a modified version of the resource reservation message received from the MTA. The AN performs any backbone-technology-specific mapping of the resource reservation that is needed. This enables the architecture to accommodate different backbone technologies, at the service provider's choosing. The specific mechanisms for reserving backbone QoS are outside the scope of the present document.

A bidirectional model is used for resource reservation in the J.112 network where the routing is symmetric. A unidirectional model is used for resource reservation in the backbone, which allows routing asymmetries. Thus, when MTA_O makes a reservation with the AN, it knows two things: that it has adequate bandwidth in both directions over the J.112 network, and that it has adequate bandwidth over the backbone networks for the MTA_O to MTA_T flow. Thus, MTA_O knows that resources are available end-to-end in both directions once it gets a reply from MTA_T.

5.7.10 Setting the DiffServ Code Point

This architecture also allows for the use of a Differentiated Services backbone, where there is adequate bandwidth to carry voice conversations, but access to this bandwidth is on a controlled basis. Access to the bandwidth and differentiated treatment is provided to packets with the appropriate encoding of bits in the field of the IP header specified for Differentiated Service. This is called the Diffserv code point (DSCP). The DS field maintains backward compatibility with the present uses of the IP Precedence bits of the IPv4 TOS byte (RFC 2474). It is desirable to be able to set the Diffserv code point of packets that are about to enter the provider backbone from the AN. Since resources consumed by these packets in the backbone may depend heavily on this marking, this architecture provides control of the marking to network entities. This allows the network and service provider the control on use of the enhanced QoS rather than trusting the MTA. The provider can configure policies in the AN that determine how to set the DSCP for flows that pass through the AN. Such policies are sent to the AN in the gate set-up protocol from the CMS/GC.

For implementation efficiency, we pass the information to the MTA about the appropriate DSCP for it to use on a given session. This is done with the IETF-proposed DCLASS object in RSVP. The AN still needs to police received packets to ensure that correct DSCP is being used and that the volume of packets in a given class is within authorized bounds.

6 MTA to AN Quality-of-Service Protocol (pkt-q3)

To meet the requirements described previously, RSVP and the IETF's Integrated Services architecture RFC 2210 is used as a basis for the signalling mechanism for providing local QoS. RSVP, as currently specified, needs some additional enhancements to meet the requirements of the Dynamic QoS architecture.

RSVP and the Integrated Services architecture specify QoS parameters in generic terms that are independent of the underlying layer 2 technology. It is necessary to specify a means of mapping those general traffic specifications into specific J.112 Flow specifications. Such mappings exist for other layer 2 protocols (e.g. ATM, IEEE 802.XX LANs); this clause describes mappings for J.112 networks.

The Dynamic QoS Architecture uses a superset of RSVP with the following differences:

- Since resource reservations are independently initiated for each J.112 network (segmented resource allocation model), the present document does not depend on resource management messages propagating end-to-end.
- The resource management exchange between the MTA and AN reserves resources in *both* directions over the local area (i.e. customer-operated) and J.112 networks. This allows the AN to act as a proxy for the far endpoint, with the benefit of minimizing the number of messages required for resource management in bandwidth-constrained J.112 networks, and reduces the post-dial and post-pick-up delay.
- In the local area (i.e. customer-operated) portion of the network, existing RSVP-capable routers may be present. In this environment, unidirectional reservations are required. To enable these two functions (bidirectional reservations on the J.112 network and unidirectional reservations inside the customer-premises), an enhanced PATH message is issued by the MTA to the Gate.
- Ability to bind a single set of resources to a group of multiple reservations, based on information from the MTA that only one reservation in the group will be active at any given time.

- Support for the two-phase resource activation facility available in J.112, giving the ability to guarantee resources are available before ringing of the far-end phone. The RSVP exchange with the AN performs the first stage, the admission control, and the MTA sends a separate message to the AN to perform the activation.

The Dynamic Quality of Service operation does not address standard RSVP, which may or may not be supported. Regardless, standard RSVP messages will not trigger the DQoS operations specified in this document.

6.1 RSVP Extensions Overview

6.1.1 Segmented Operation

As defined in RFC 2205, RSVP is intended to run between a pair of hosts. However, the IPCablecom QoS model requires the signalling to be done in a segmented manner, where one segment is between an MTA and an AN. This clause illustrates how RSVP can support a segmented model.

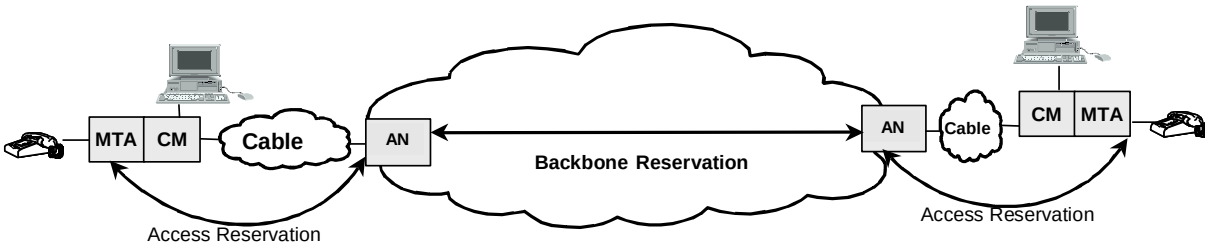


Figure 6: Segmented Signalling Model

In the segmented model, an MTA communicates with the AN. In addition to the simple scenario pictured in figure 6, the present document allows for more complex scenarios, such as when there is a customer network between the client and the CM, which may include a variety of network elements, including RSVP-capable switches or routers. The presence of a customer network means that the solution works even if the client and the AN are not immediately adjacent at the IP layer. The customer network may provide multiple paths between the client and the CM, leading to the possibility of asymmetric routes in this network.

The AN intercepts RSVP messages sent from the originating MTA to the MTA on the terminating side of the session to implement the Segmented model. This minimizes changes to RSVP, by keeping the destination address of the PATH messages the same as the destination address of the data.

6.1.2 Bidirectional Reservations

Traditional RSVP makes unidirectional reservations. PATH messages flow in the same direction as data, and RESV messages flow in the opposite direction. To make a bidirectional reservation, it is necessary to add new RSVP objects to define both directions. The AN responds to the request by establishing reservations on both directions of the J.112 link. If there are RSVP-capable routers between the originating MTA and the CM, then the AN initiates a PATH message that appears to have come from the remote client.

6.1.3 Header Compression, Suppression and VAD

If the AN and CM are configured to perform header compression or suppression, then the bandwidth that is needed for a J.112 Flow may be reduced. It is necessary for the client to convey to the AN the fact that compression or suppression may be applied prior to the installation of a reservation to ensure that appropriate bandwidth is reserved. The general solution to this problem is described in Integrated Services in the Presence of Compressible Flows [draft-davie-intserv-compress-02].

The MTA adds a parameter (Compression_Hint) described in [draft-davie-intserv-compress-02] to the Sender-TSpec that identifies the type(s) of header compression or suppression that might be applied to the data. The Compression_Hint parameter contains a Hint field that advertises the type(s) of compression or suppression that is (are) possible, as well as whether the sender is using UDP or IP checksums and/or IP-Ident; if these are not used, these fields may also be compressed or suppressed. If any field in the IP header is not being compressed or suppressed, then the IP checksum MUST NOT be compressed or suppressed.

To signal header suppression to the J.112 network, the AN uses the data provided by the Hint field of the Compression_Hint parameter to indicate the scheme of header suppression that will be performed on this J.112 Flow. This information is used to reduce the effective rate and depth of the token bucket supplied by the MTA. If header suppression is not supported on a link, the Compression_Hint parameter is ignored and the full TSpec is used.

When performing header suppression on a J.112 link, it is also necessary to communicate the *contents* of the header that will be suppressed to the AN in advance of the first data packet's transmission so that the suppression context can be established at the CM and the AN. This information may be delivered by the RSVP message that is used to establish the reservation or through MAC-layer messages sent ahead of the first data packet. Since PATH messages are processed by any intermediate hops between the client and the AN, an arriving PATH message will contain the same TTL value as data packets, provided PATH messages and data packets have the same initial TTL when sent by the MTA. The AN thus may use the contents of the PATH to learn the values of the fields that will be suppressed. The AN uses J.112 MAC-messaging to convey to the CM the fact that suppression should be used for a particular flow, and instructs it to suppress appropriate fields given the presence or absence of UDP checksums.

The AN also may instruct the CM to suppress the IP Identification field. This field is used only when fragmentation occurs. Since this field changes with every packet, its value can neither be conveyed using RSVP nor using MAC messages. The question of whether to suppress it or not depends on whether the packet might be fragmented later. There is no need for the MTA to convey any information to the AN regarding the suppressibility of this field; the AN may decide to suppress it or not based on a local policy.

The same basic approach enables support of Voice Activity Detection (VAD). An AN may use different scheduling algorithms for flows that are using VAD, and thus needs to know which flows may be treated with VAD. The compressibility object carried in the Tspec MUST contain a value which indicates that the data flow for which this reservation is being requested may be treated with VAD (i.e. it has not already undergone silence detection at the MTA, and it is voice, not fax or data).

6.1.4 Dynamic Binding of Resources

The Dynamic QoS model requires the ability to dynamically modify the binding of resources to flows. For example, to provide Call Waiting, it may be desirable to hold enough resources for only one session in place over the J.112 network, and to switch the allocation of those resources from one caller to another. While this capability has been suggested for RSVP in the past, it was not included in RSVP version 1.

In RSVP, the "handle" on a set of reserved resources is the Session-Object. Since the Session contains the destination address of the flow, reallocation of resources to a flow with a different destination address would require a change in the Session-Object. Changing the source address of the flow could be accomplished using a new Filterspec in the RESV message.

To accommodate this functionality, a Resource-ID object is added to RSVP messages. Routers, which understand this object, will attempt to use the resources associated with that ID. The Resource-ID object is an opaque identifier generated by the node that has control of the resources, i.e. the AN in this case.

This is illustrated in figure 7. When an MTA issues a reservation request for a new flow, it indicates to the AN that this session is willing to share resources for this new gate (Gate 2) with a previously created gate (Gate 1) by including the ResourceID in the request. As long as the QoS requested for the new gate can be satisfied with a bandwidth allocation equal to or less than the existing gate, no new bandwidth is reserved in the J.112 network. However, bandwidth may need to be reserved in the backbone network depending on the end-to-end path taken by the new session. Access to the shared reservation occurs in a mutually exclusive manner: a MTA has to issue a commit message to indicate to the AN which flow is currently active, and that commit explicitly removes the committed resources for the other. In the call waiting example, the client sends a commit message to the AN to identify the currently active flow when the user switches between sessions.

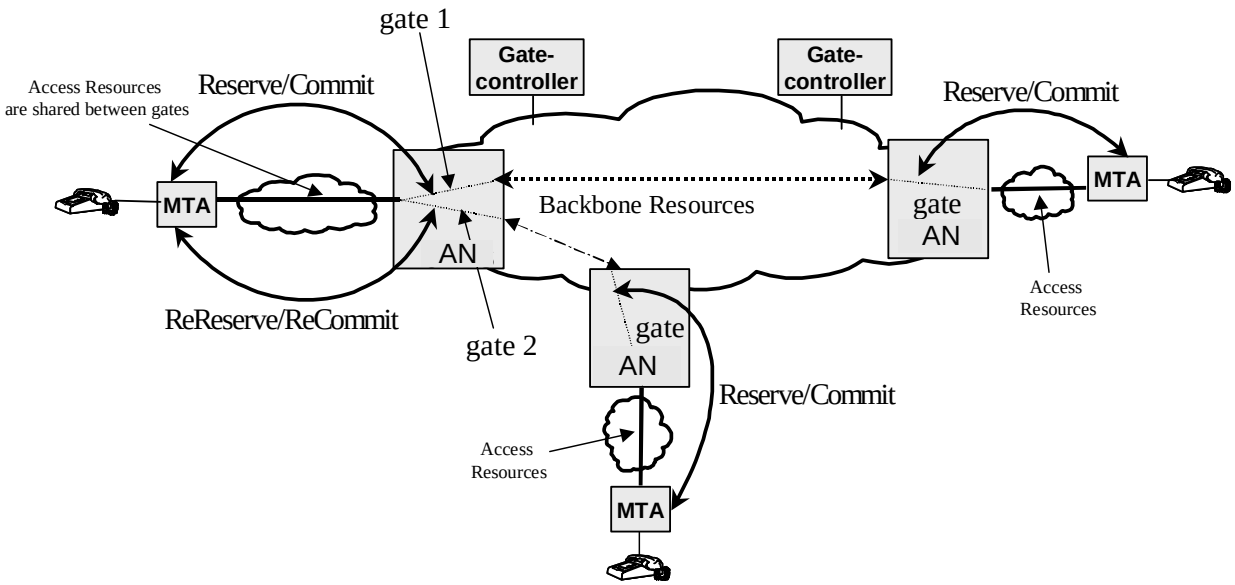


Figure 7: Sharing of Resource Reservations across Gates

In the segmented model, the AN includes the Resource-ID in the first RESV message that it sends to the MTA. The MTA may include the Resource-ID in subsequent messages that apply to the resources in question. Most importantly, if the MTA wishes to establish a new session, and reuse the resources of an existing session, it includes the Resource-ID associated with the old session in the PATH message it sends to the AN. A PATH message that contains the Resource-ID of a currently allocated set of resources adds a new binding between a flow (as identified in the Session and Sender-Template objects) and those resources. It may optionally change the amount of resources allocated by the inclusion of Tspecs and Rspecs which differ from those previously received by the AN for this set of resources. This may include the addition of a new set of Tspecs and Rspecs to accommodate multiple codecs as described in clause 6.2.1.

RSVP allows reservations to vary in size over time. A reservation that is no greater than the one currently installed (i.e. does not require an increased level of resources in any dimension for either direction of the session) MUST NOT fail admission control. The same rule applies when using the Resource-ID object. If the amount of resource requested in the new reservation is no greater than previously installed, the reservation MUST NOT fail admission control.

A router that does not understand this new object (e.g. in the customer network) will simply try to install what appears to be a new reservation without reusing previously allocated resources. Since it is unlikely that there is less bandwidth in the home than on the J.112 network, this is unlikely to be a problem. The old reservation will time out if it is not refreshed. In the event that resource scarcity is a problem in the customer network, it would be necessary to upgrade the routers in the home to support this new object. Note that attempting to install reservations on the customer network is worthwhile even if bandwidth is relatively abundant there, as a reservation provides devices in the customer network with the necessary information to isolate specific flows from excessive delay and jitter that they would otherwise experience if simply mixed with best effort traffic (or reserved flows of widely different traffic characteristics) in a common queue.

6.1.5 Two-Stage Reserve/Commit Process

A significant aspect of the IPCablecom Dynamic QoS model is that reservation is a two-phase process, with a Commit phase following the Reserve phase. RSVP is used to cover the Reserve phase, so the AN does not actually provide the resources until the second stage of the process.

Because the commit phase only involves a MTA and a local gate, it is a unicast message from the MTA to the AN. The MTA learns the gate-ID from the call signalling protocol.

6.1.6 Authentication

The provider is able to ensure that parties do not reserve unauthorized network resources. RSVP provides a number of mechanisms to do this, such as RSVP Integrity-Objects and policy data contained in other RSVP messages. The Dynamic QoS specification includes a GateID as policy data, which MUST be included in the RSVP-PATH messages.

6.2 RSVP Flowspecs

The IETF Integrated Services architecture uses general purpose (layer 2 independent) descriptions of the traffic characteristics and resource requirements of a flow. The description of the traffic is known as a TSpec, the resource requirements are contained in an RSpec, and the combination of these is known as a Flowspec. In order to reserve resources on a specific layer 2 medium such as a J.112 network, it is necessary to define a mapping from the layer 2 independent Flowspec to specific layer 2 parameters. Mappings for a variety of other technologies (ATM, 802.3 LANs, etc.) have already been defined.

Other specifications (e.g. the IPCablecom CODEC specification J.acf) contain the mapping requirements of higher-layer service descriptions (e.g. SDP as used in VoIP applications) into Flowspecs. Annex A and annex B of this ITU-T Recommendation specifies how the AN and MTA MUST map Flowspecs to J.112 layer 2 parameters.

Integrated Services currently defines two types of service, controlled load and guaranteed, the latter being the more suitable for latency sensitive applications. When making a reservation for guaranteed service, the Flowspec contains:

Tspec

Bucket depth (b) - bytes
bucket rate (r) - bytes/second
peak rate (p) - bytes/second
min policed unit (m) - bytes
maximum datagram size (M) - bytes

RSpec

reserved rate (R) - bytes/second
slack term (S) - microseconds

The TSpec terms are mostly self-explanatory. (r,b) specifies a token bucket that the traffic conforms to, p is the peak rate at which the source will send, and M is the maximum packet size (including IP and higher layer headers) that will be generated by the source. The minimum policed unit, m, is usually the smallest packet size that the source will generate; if the source sends a smaller packet, it will count as a packet of size m for the purposes of policing.

To understand the RSpec, it is helpful to understand how delay is calculated in an Integrated Services environment. The maximum end-to-end delay experienced by a packet receiving Guaranteed service is:

$$\text{Delay} = b/R + C_{tot}/R + D_{tot}$$

where b and R are as defined above, and C_{tot} and D_{tot} are accumulated "error terms" provided by the network elements along the path, which describe their deviation from "ideal" behaviour.

The rate R provided in the RSpec is the amount of bandwidth allocated to the flow. It MUST be greater than or equal to r from the TSpec for the above delay bound to hold. Thus, a flow's delay bound is completely determined by the choice of R; the reason to use a value of R greater than r would be to reduce the delay experienced by the flow.

Since it is not permissible to set $R < r$, a node making a reservation may perform the above calculation and determine that the delay bound is tighter than needed. In such a case, the node may set R=r and set S to a non-zero value. The value of S would be chosen such that

$$\text{Desired delay bound} = S + b/R + C_{tot}/R + D_{tot}$$

Guaranteed Service does not attempt to bound jitter any more than is implied by the delay bound. In general, minimum delay that a packet might experience is the speed of light delay, and the maximum is the delay bound given above; the maximum jitter is the difference between these two. Thus jitter may be controlled by suitable choice of R and S.

There are various situations in which a reservation needs to cover a range of possible flowspecs. For example, for some applications it is desirable to create a reservation, which can handle a switch from one codec to another mid-session without having to pass through admission control at each switch-over time.

In cases such as this, the MTA MUST generate multiple Tspecs. The second and later Tspecs MUST be marked as Component Tspecs (see clause 6.3.7), and contain the Flowspec parameters for an individual codec. The first Tspec MUST be formed by, for each component in the flow description, taking the maximum resource usage of any of the following Component Tspecs. This is referred to as the Least-Upper-Bound (LUB). With the LUB placed in a standard RSVP Tspec, any router not familiar with these extensions will allocate sufficient (and possibly more than sufficient) resources to carry any of the alternatives.

Simply taking the least upper bound of two flowspecs causes some loss of information. For example, suppose codec A is G.726 -24 kbit/s at 20 ms packets, which requires a Tspec of:

bucket depth (b) = 100 bytes
bucket rate (r) = 5 000 bytes/s
peak rate (p) = 5 000 bytes/s
min policed unit (m) = 100 bytes
maximum datagram size (M) = 100 bytes

while codec B is G.726 -40 kbit/s at 10 ms packets, which requires a Tspec of:

bucket depth (b) = 90 bytes
bucket rate (r) = 9 000 bytes/s
peak rate (p) = 9 000 bytes/s
min policed unit (m) = 90 bytes
maximum datagram size (M) = 90 bytes

Looking first at Codec A, we conclude that it needs a grant to transport IP packets of size 100 bytes every 20 ms (M/r = 0,02 s) , while Codec B requires a grant to deliver 90 byte packets every 10 ms. However, the least upper bound of the two Tspecs is:

bucket depth (b) = 100 bytes
bucket rate (r) = 9 000 bytes/s
peak rate (p) = 9 000 bytes/s
min policed unit (m) = 100 bytes
maximum datagram size (M) = 100 bytes

which leads to giving a grant for 100 bytes every 11,1 ms (M/r = 100/9), which is not appropriate for either of the sessions. For this reason, when making a reservation that will need to cover two or more different flowspecs, each component flowspec MUST be included in the appropriate RSVP messages.

6.3 Definition of Additional RSVP objects

Several new RSVP objects MUST be added to the original PATH message sent by the MTA. All new objects have a class-number with the high order two bits set, which means that RSVP nodes that do not recognize these objects should forward them without modification. This clause defines the formats of the various new objects that are to be carried in RSVP messages. All objects use the TLV encoding scheme of RSVP RFC 2205.

6.3.1 Reverse-Rspec

Reverse-Rspec object: Class = 226, C-type = 1

130 (h)	0 (i)	2 (j)
Rate [R] (32-bit IEEE floating point number)		
Slack Term [S] (32-bit integer)		
h)	Parameter ID, parameter 130 (Guaranteed Service RSpec)	
i)	Parameter 130 flags (none set)	
j)	Parameter 130 length, 2 words not including parameter header	

See RFC 2210 for explanation of fields.

The Reverse-RSpec applies to data sent by the MTA, i.e. upstream in the J.112 network. It is included in the PATH message sent by the MTA, and is turned into the Forward-RSpec object in the RESV message generated by the AN in its role as proxy for the remote endpoint.

6.3.2 Reverse-Session

IPv4 Reverse Session object: Class = 226, C-Type = 2

IPv4 Destination Address (4 bytes)		
Protocol ID	Flags	Destination Port

The Reverse-Session object describes the destination information of the data stream to be received by the MTA, i.e. downstream in the J.112 network. It becomes the Session Object in the PATH message generated by the AN in its role as proxy for the remote endpoint.

6.3.3 Reverse-Sender-Template

IPv4 Reverse-Sender-Template object: Class = 226, C-Type = 3

IPv4 Source Address (4 bytes)		
Reserved	Reserved	Source Port

The Reverse-Sender-Template describes the source information of the data stream to be received by the MTA, i.e. downstream in the J.112 network. It becomes the Sender-Template object in the PATH message generated by the AN in its role as proxy for the remote endpoint.

6.3.4 Reverse-Sender-Tspec

Reverse-Sender-Tspec object: Class = 226, C-Type = 4. Same fields as Sender-Tspec described in Integrated Services in the Presence of Compressible Flows [draft-davie-intserv-compress-02].

0 (a)	Reserved	10 (b)
1 (c)	0 Reserved	9 (d)
127 (e)	0 (f)	5 (g)
Token Bucket Rate [r] (32-bit IEEE floating point number)		
Token Bucket Size [b] (32-bit IEEE floating point number)		
Peak Data Rate [p] (32-bit IEEE floating point number)		
Minimum Policed Unit [m] (32-bit integer)		
Maximum Packet Size [M] (32-bit integer)		
126 (h)	flags (i)	2 (j)
Hint (assigned number) (k)		
Compression factor (32-bit integer) (l)		

(a) Message format version number (0)
(b) Overall length (10 words not including header)
(c) Service header, service number 1 (default/global information)
(d) Length of service 1 data, 9 words not including header
(e) Parameter ID, parameter 127 (Token_Bucket_TSpec)
(f) Parameter 127 flags (none set)
(g) Parameter 127 length, 5 words not including header
(h) Parameter ID, parameter 126 (Compression_Hint)
(i) Parameter 126 flags (none set)
(j) Parameter 126 length, 2 words not including header
(k) Hint value defined for J.112 header suppression (TBD)
- 0x????0001 Do not suppress UDP checksum AND do not suppress IP-Ident field nor IP-Checksum field
- 0x????0002 Do not suppress UDP checksum AND suppress IP-Ident and IP-Checksum field
- 0x????0003 Suppress UDP checksum AND do not suppress IP-Ident nor IP-Checksum field
- 0x????0004 Suppress UDP checksum AND suppress IP-Ident field and IP-Checksum field
- ???? TBD IANA number assignment for IPCablecom

(l) Compression factor value - the percentage reduction in packet size as a result of using J.112 header suppression.
NOTE: This varies depending on the CODEC used. See RFC 2210 for explanation of fields.

The Reverse-Sender-TSpec describes the data flow to be sent by the MTA, i.e. upstream in the J.112 network. It becomes the Sender-TSpec object in the PATH message generated by the AN in its role as proxy for the remote endpoint.

6.3.5 Forward-Rspec

Forward-Rspec object, Class = 226, C-type = 5. Same fields as Reverse-Rspec.

130 (h)	0 (i)	2 (j)
Rate [R] (32-bit IEEE floating point number)		
Slack Term [S] (32-bit integer)		

The Forward-Rspec applies to data flowing toward the MTA, i.e. downstream in the J.112 network. This object appears in a PATH message sent by the MTA, and the contents are incorporated into the Flowspec object in the returned RESV message.

6.3.6 Component-Tspec

Component-Tspec object: Class = 226, C-type = 6. Same fields as Sender-Tspec defined in Integrated Services in the Presence of Compressible Flows [draft-davie-intserv-compress-02].

0 (a) Reserved		10 (b)
1 (c)	0 Reserved	9 (d)
127 (e)	0 (f)	5 (g)
Token Bucket Rate [r] (32-bit IEEE floating point number)		
Token Bucket Size [b] (32-bit IEEE floating point number)		
Peak Data Rate (p) (32-bit IEEE floating point number)		
Minimum Policed Unit [m] (32-bit integer)		
Maximum Packet Size [M] (32-bit integer)		
126 (h)	flags (i)	2 (j)
Hint (assigned number) (k)		
Compression factor (32-bit integer) (l)		

(a) Message format version number (0)
(b) Overall length (10 words not including header)
(c) Service header, service number 1 (default/global information)
(d) Length of service 1 data, 9 words not including header
(e) Parameter ID, parameter 127 (Token_Bucket_TSpec)
(f) Parameter 127 flags (none set)
(g) Parameter 127 length, 5 words not including header
(h) Parameter ID, parameter 126 (Compression_Hint)
(i) Parameter 126 flags (none set)
(j) Parameter 126 length, 2 words not including header
(k) Hint value defined for J.112 header suppression (TBD)

0x????0001	Do not suppress UDP checksum AND do not suppress IP-Ident field nor IP-Checksum field
0x????0002	Do not suppress UDP checksum AND suppress IP-Ident and IP-Checksum field
0x????0003	Suppress UDP checksum AND do not suppress IP-Ident nor IP-Checksum field
0x????0004	Suppress UDP checksum AND suppress IP-Ident field and IP-Checksum field
????	TBD IANA number assignment for IPCablecom

(l) Compression factor value - the percentage reduction in packet size as a result of using J.112 header suppression. Note this varies depending on the CODEC used.

6.3.7 Resource-ID

Resource-ID object: Class = 226, C-type = 7.

Resource ID (32-bit integer)

The Resource-ID object is returned in a RESV message to the MTA, and contains the identifier used for future resource changes. It is also included in PATH messages sent by the MTA in requests to share resources across multiple sessions.

6.3.8 Gate-ID

Gate-ID object: Class = 226, C-type = 8.

Gate ID (32-bit integer)

The Gate-ID object is included in PATH messages from the MTA to identify the proper resource authorization at the AN.

6.3.9 Commit-Entity

IPv4 Commit-Entity object: Class = 226, C-type = 9.

IPv4 Destination Address (4 bytes)	
Reserved	Destination Port

The Commit-Entity object is returned in a RESV message from the AN, and indicates the destination address and port number to which the MTA is to send the COMMIT message.

6.3.10 DClass

DClass object: Class = 225, C-Type = 1

Unused	Unused	Unused	DSCP

The DClass object is returned in a RESV message from the AN, and indicates the DSCP that SHOULD be used by the MTA when sending data packets over this reservation to the AN. The use of the DClass object is described in Use and Format of the DCLASS Object with RSVP Signalling [draft-bernet-dclass-01].

6.4 Definition of RSVP Messages

This clause defines the enhanced RSVP messages that MUST be generated by the MTA and MUST be supported by the AN.

RSVP messages MUST be sent as "raw" IP datagrams with protocol number 46. The RSVP-PATH message MUST be sent with the RouterAlert option RFC 2113 in the IP header. Each RSVP message MUST occupy exactly one IP datagram.

All RSVP messages MUST consist of a Common Header, followed by a variable number of variable-length objects. The Common Header MUST be as follows:

Version	Flags	Message Type	RSVP Checksum
Sent-TTL		(Reserved)	RSVP Message Length

Values of each field MUST be as specified in RFC 2205.

Each object MUST consist of one or more 32-bit words with a one-word header, of the following format:

Length in bytes	Class-Number	C-Type
Object Contents …		

Values of each field MUST be as specified in RFC 2205.

The format of the RSVP-PATH message and RSVP-RESV message compliant with the present document MUST contain the following objects (items in italics are defined in the present document, all others in RFC 2205 and/or RFC 2210). For objects not defined in the present document, the object ordering rules MUST be followed according to RFC 2205. There is no ordering requirement for the <Resource-ID>, <Gate-ID>, and <Commit-Entity> objects. <Reverse-Rspec> and <Downstream-Flowspec> MUST follow the <Sender-Tspec> object. If <Component-Item> is included in the message, <Component-Item> MUST appear in the PATH message after the <Sender-Tspec><Reverse-Rspec><Downstream-Flowspec> triple. Objects defined in <Downstream-Flowspec> and <Component-Item> MUST follow the order shown in their BNF below:

```
<PATH-Message> ::= <Common-Header> [<Integrity-Object>]

                        <Session-Object> <RSVP-Hop> <Time-Values>
                        [<Policy-Data> …] <Sender-Template>
                        <Sender-Tspec> <Reverse-Rspec>
                        <Downstream-Flowspec> [<Resource-ID>]
                        <Gate-ID> [<Component-Item> …]

      <Downstream-Flowspec> ::= <Reverse-Session> <Reverse-Sender-Template>
                        <Reverse-Sender-Tspec><Forward-Rspec>

      <Component-Item> ::= <Component-Tspec> <Reverse-Rspec>
                        <Downstream-Flowspec>
<RESV-Message> ::= <Common-Header> [<Integrity-Object>]

                        <Session-Object> <RSVP-Hop> [<DClass>]
                        <Time-Values> [<RESV-Confirm>] [<Scope>]
                        [<Policy-Data> …] <Resource-ID>
                        <Commit-Entity> <Style> <Flowspec>
                        <Filter-Spec>
```

The various components of these messages are described in the following clauses.

6.4.1 Message Objects for Upstream Reservation

A standard RSVP-PATH message contains, at a minimum, the following objects:

```
<Session> <RSVP-Hop> <Time-Values> <Sender-Template> <Sender-Tspec>
```

However, in the segmented model, it is necessary to get all the information to the AN that would enable it to make a bidirectional reservation on the J.112 link. It is also necessary to enable it to send an RSVP-RESV towards the MTA. A standard RSVP RESV message contains, at a minimum, the following objects:

```
<Session> <RSVP-Hop> <Time-Values> <Style> <Flowspec> <Filter-Spec>
```

The AN MUST generate such a message towards the MTA after receiving an RSVP-PATH message from the MTA. The only object here that is not derivable from the RSVP-PATH or local information is the Flowspec. The Filter-Spec, which consists of the IP address and source port to be used by the MTA, is derived from the Sender-Template in the PATH. Almost everything in the Flowspec can be derived from the Sender-Tspec in the PATH message. The exceptions to this are the values of R (reserved rate) and S (slack), which together constitute the RSpec. Thus, the MTA provides a suitable RSpec, containing R and S for guaranteed service, which is encoded as in RFC 2210. This is enclosed in a Reverse-RSpec object, which is described in clause 6.3.2.

6.4.2 Message Objects for Downstream Reservation

The MTA MUST provide enough information to allow the AN to construct an RSVP-PATH message for the downstream data flow given that it has just received an RSVP-PATH message for the upstream data flow. This means the MTA MUST provide the following objects that relate to the downstream (AN->MTA) data flow:

```
<Session> <Sender-Template> <Sender-Tspec>
```

These objects have their normal RSVP definitions, and apply to the simplex data stream that will flow from the far endpoint to the MTA. In the RSVP-PATH message sent by the MTA, they are assigned new object codes (as noted above) and new names (Reverse-session, Reverse-sender-template, Reverse-Sender-Tspec). The Reverse-Session-Object MUST contain the IP address of the MTA, the protocol type, and the port (if applicable) on which it will receive data for this flow. The Reverse-Sender-Template MUST contain the IP address of the far endpoint, or zeroes if the source is intended as a wildcard. The Reverse-Sender-Template MUST contain the port number, if applicable and known, otherwise zero. The Reverse-Sender-TSpec MUST contain the TSpec information that describes the data flow from the far endpoint. The AN MUST use its own address as the RSVP-Hop and choose a value for Time-Values that indicates how often it will refresh the RSVP-PATH message. Even if the AN does not need to generate the RSVP-PATH message to send it to the MTA, this information is necessary to enable it to establish a reservation and create packet classifiers in the downstream direction.

Given the information described above, the one additional piece of information that the AN requires to make a reservation in the downstream direction is an RSpec. Again, it is assigned a new object number and name, Forward-Rspec. It contains the same information elements and is encoded in the same way as a conventional RSpec.

Note that a Forward-Rspec applies to data that is flowing towards the MTA, which means that it is sent by the MTA in the same direction as the RSVP-RESV that would normally carry this information. It is provided in the RSVP-PATH message simply as an optimization to reduce set-up latency. A Reverse-RSpec is sent by the MTA in the opposite direction to the RSVP-RESV that would normally carry this information.

6.4.3 Message Objects for Support of Multiple Flowspecs

To accommodate the multiple codec situation described in clause 6.2.1, a PATH message may need to carry multiple Tspecs and Rspecs. At the same time, RSVP-capable devices between the MTA and the AN need to receive the least upper bound Tspec and Rspec. Thus, in the case where resources are being reserved with the goal of accommodating multiple codecs, a standard Tspec or Rspec object carried in an RSVP message should contain the least upper-bound of the resources required. Additional Tspecs and Rspecs may be included in the PATH message, using new object types that will be ignored by standard RSVP devices. Since all the objects describing the Downstream-flowspec and Reverse-Rspec will be ignored by standard RSVP, the only new object needed is a Component-Tspec object that MAY be carried in the RSVP-PATH message. There may be two or more such objects in a RSVP-PATH message, in addition to the standard Tspec that is required to carry the least upper bound of all the components, and which will be used by devices in the customer network. The interpretation of each Component-Tspec object is that the resources reserved on the J.112 link are suitable to accommodate any flow matching one of these Tspecs.

Similarly, there MAY be multiple Reverse-Rspecs, Reverse-Session, Reverse-Sender-Template, Reverse-Sender-Tspec, and Forward-Rspecs objects. Since it is necessary to be able to correctly identify which combination of forward and reverse parameters need to be accommodated at one time, the order of these objects in the RSVP-PATH message is important. The order given above, in clause 6.4, is REQUIRED.

6.5 Reservation Operation

This clause describes the required behaviour of the MTA and AN to co-operatively perform resource reservations.

For the purposes of this discussion, the endpoint that is in direct communication with the AN is referred to as the client, and the other endpoint of the session is referred to as the far endpoint. We make no assumptions about what types of devices these might be (gateways, PCs, embedded clients). We assume that the client uses RSVP to communicate QoS requests to the AN, and we make no assumptions about the capabilities of the far endpoint. The data flow from client to AN is referred to as upstream, and the flow from AN to client is referred to as downstream.

6.5.1 Reservation Establishment

RSVP operation under the segmented model is as follows:

The client MUST send a RSVP-PATH message towards the far endpoint of the session, which MUST be intercepted by the AN. This initiates the process of reserving both upstream and downstream bandwidth. The RSVP-PATH MUST carry information about both upstream (i.e. Reverse-RSpec) and downstream (i.e. Reverse-Sender-TSpec, Forward-RSpec) resource requirements in the case where reservations are required in both directions.

The AN MUST verify that the amount of resources requested is within the authorized amount for this session and that it has sufficient local resources to accommodate the reservation. It then reserves upstream and downstream resources and MUST perform the J.112 MAC-level messaging to allocate appropriate resources on the J.112 link.

The AN MUST establish classifiers for the upstream and downstream flows. The upstream classifier MUST contain the client's source IP address and port number from the Sender Template object. The upstream classifier MUST contain the protocol type, destination IP address and port number from the Session Object. If the Reverse-Sender-Template Object is present, and contains an address other than 0.0.0.0, then the downstream classifier MUST contain this address as the source IP address. If the Reverse-Sender-Template is present, and contains a port number other than 0, then the downstream classifier MUST contain this value as the source port. The downstream classifier MUST contain the protocol type, destination IP address, and port number from the Reverse Session Object.

The AN MUST perform any backbone resource reservation necessary, based on the provisioned algorithm defined for the particular backbone configuration.

If the access and backbone reservations succeed, the AN MUST send a RSVP-RESV to the client. The contents of the RSVP-RESV MUST be derived from the RSVP-PATH: the Session-Object is copied from the RSVP-PATH, Style is set to Fixed-Filter, Flowspec is formed from the Sender-Tspec and Forward-Rspec, Filter-Spec is set from the Sender-Template, and the Resource-ID is generated, containing the Resource-ID assigned to the allocated resources. The Commit-Entity Object MUST be included, and contain the address of the AN and port number on which the AN will accept the COMMIT message (as described in clause 6.6). The DCLASS object SHOULD be included and value set based on the Diffserv Code Point field of the gate.

If the address of the previous hop differs from the Source Address of the RSVP-PATH message, then the AN MUST generate a RSVP-PATH for downstream reservations. The contents of the RSVP-PATH MUST be derived from the RSVP-PATH received from the client. The Session-Object MUST be obtained from the Reverse-Session-Object in the RSVP-PATH message. If the address contained in the Reverse-Sender-Template is 0.0.0.0, or the port number is 0, then the Sender-Tspec and Sender-Template are not sent in the RSVP-PATH. Otherwise, the Sender-Tspec is obtained from the Reverse-Sender-Tspec, the Forward-Rspec is obtained from the Reverse-Rspec, and the Sender-Template is obtained from the Reverse-Sender-Template. The Resource-ID object is generated, and contains the Resource-ID assigned to the allocated resources. The MTA MAY use the Reverse-Sender-Tspec it sent in the RSVP-PATH message in calculating the Filter-Spec returned in its RSVP-RESV reply, or MAY generate a Wildcard-Filter reply. On receipt of the RSVP-RESV message, the client knows that necessary resources have been reserved. At this point, in the case of a successful reservation, the client knows that it has a reservation in both directions, and can proceed with the call signalling to ring the phone at the far end.

If the reservation does not succeed, the AN MUST send a RSVP-PATH-ERR message to the client, indicating why the reservation failed (e.g. lack of authorization, insufficient resources, etc.). If the reservation failed for policy reasons, the RSVP-PATH-ERR message MUST contain a RSVP-ERROR-SPEC object with the following Error Codes and Error Values:

- Error Code = 2 (Policy Control Failure), Error Value = 3 (Generic Policy Rejection) is returned if the RSVP-PATH did not contain a Gate-ID object or the Gate-ID object did not match any gates known to the AN.
- Error Code = 1 (Admission Control Failure), Error Value = 2 (Requested bandwidth unavailable) is returned if the RSVP-PATH was rejected because there was no more resources that are available for the priority level of the gate. In these cases, the MTA MAY take special action indicating the specific error to the user. If the RSVP-PATH failed for non-policy reasons, it MUST contain a RSVP-ERROR-SPEC object with an Error Code and Error Value as defined in Appendix B of RFC 2205.

The sender of a RSVP-PATH (MTA or AN) is responsible for reliably installing the reservation. When the sender transmits a RSVP-PATH, it MUST receive an RSVP-RESV or RSVP-PATH-ERR message within a configured timeout interval of Timer T3 (see annex C).

Whenever an MTA or AN transmits an RSVP message that requires an acknowledgement, the sender MUST include an RSVP-MESSAGE-ID object in that message, and the ACK_Desired flag of the RSVP-MESSAGE-ID object MUST be set. The MTA and AN MUST set the Refresh-Reduction-Capable flag in the common header of every RSVP message. When the MTA or AN receives an RSVP message with an RSVP-MESSAGE-ID object, it MUST respond with an RSVP message that contains an RSVP-MESSAGE-ACK or RSVP-MESSAGE-NACK object. The RSVP-MESSAGE-(N)ACK object MAY be piggy-backed onto standard RSVP messages, but MAY be transmitted in an RSVP-ACK message if the receiver of the RSVP-MESSAGE-ID object has no other RSVP message to send at the time. For example, the AN SHOULD NOT delay processing of a received RSVP-PATH message, but if it chooses to delay, it MUST reply immediately with an RSVP-ACK message, to be followed by a RSVP-RESV message later.

RSVP-ACK messages carry one or more RSVP-MESSAGE-(N)ACK objects. They MUST NOT contain any other RSVP objects except an optional RSVP-INTEGRITY object. When included, an RSVP-MESSAGE-(N)ACK object MUST be the first object in the message, unless an RSVP-INTEGRITY object is present (in which case, the RSVP-MESSAGE-(N)ACK object MUST immediately follow the RSVP-INTEGRITY object). The MTA or AN MAY use RSVP-INTEGRITY objects.

The use of RSVP-MESSAGE-ID and RSVP-MESSAGE-(N)ACK objects can be used to ensure reliable RSVP message delivery in the face of network loss. Since the MTA or AN sets the ACK_Desired flag, it MUST retransmit unacknowledged messages at a more rapid interval than the standard RSVP refresh interval until the message is acknowledged or until an interval of Timer T3 (see annex C) elapses. A rapid retransmit rate based on well-known exponential back-off functions MUST be used. An initial retransmit timeout of Timer T6 (see annex C) MUST be used, with a power of 2 back-off. The rapid retransmit process ends when either an RSVP-MESSAGE-(N)ACK object is received or Timer T3 expires. If RSVP-PATH sender does not receive a RSVP-RESV, RSVP-PATH-ERROR, or RSVP-MESSAGE-(N)ACK before the next retransmit, it MUST assume either its original RSVP-PATH or the response from the other end was lost and re-sends the RSVP-PATH. Since all RSVP messages are idempotent, no duplications of reservations will occur.

In IPCablecom, only RSVP-PATH messages MUST include RSVP-MESSAGE-ID objects with the ACK_Desired flag set. RSVP-MESSAGE-ID objects MAY be used in other RSVP messages.

RSVP-MESSAGE-IDs are used on a per-RSVP-hop basis. Each RSVP-capable hop in the path that supports refresh reduction does its own fast retransmit until it sees an acknowledgement from the next upstream node. So if a standalone MTA behind an RSVP-capable CM receives an RSVP-MESSAGE-ACK object from the CM for an RSVP-PATH and the CM is waiting for an RSVP-MESSAGE-ACK from the AN for the RSVP-PATH, the CM will do the fast retransmit while the standalone MTA waits for its normal (30 s) RSVP-PATH refresh Timer To expire. (The MTA no longer does fast retransmit because it got an acknowledgement.) If an RSVP-capable CM gives up its fast retransmit, it will send back an RSVP-PATH-ERROR to the standalone MTA. This way, retransmits do not affect the entire path, just the loss-prone hops.

Reliable message delivery for RSVP messages is defined in RSVP Refresh Overhead Reduction Extensions [Draft-ietf-rsvp-refresh-reduct-02].

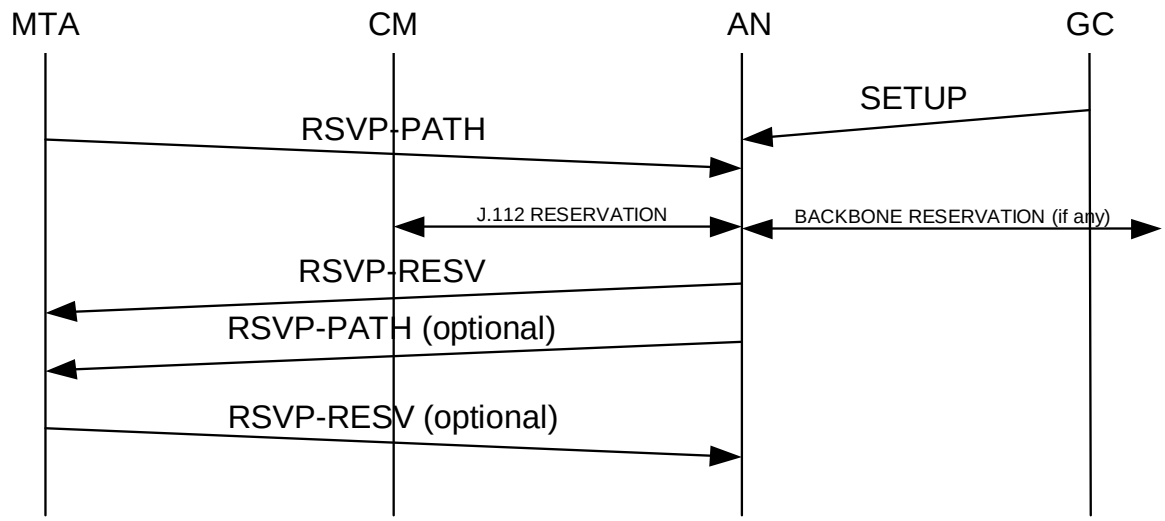


Figure 8: Reservation Establishment

The AN MUST enforce upstream packet classification filters for J.112 Flows. That is, the AN MUST discard upstream packets, which do not match the set of upstream packet classifiers for the J.112 Flow. Upstream packet classification filtering is an optional AN requirement in J.112 networks . The present document requires its implementation for J.112 Flows used to carry IPCablecom media streams. If an AN chooses to enforce upstream classification filters only on the J.112 Flows, and not on other flows, it is an AN vendor-specific decision as to how the particular J.112 Flows are determined.

6.5.2 Reservation Change

In addition to establishing a reservation for some amount of resources, it may be necessary to change the resources allocated. Resource usage may need to be increased or decreased. RSVP handles changes in resource usage by changes in the FLOWSPEC object of a RSVP-RESV message and/or a change in the Sender-Tspec in a RSVP-PATH message. A reservation change MUST follow the same series of steps as the establishment of a new reservation. Admission control SHOULD always succeed for a session, which is changing its resource requirements in a way that does not cause an increase in any dimension relative to the resources previously reserved. Because resources are described by multi-dimensional vectors, a change in reservation that increased resources in one direction and decreased them in another MUST pass through admission control. Note that in order to pass admission control, the resources MUST be within the amount of authorized resources for the session and also within the amount of resources available to the AN.

In the event that an existing reservation is pre-empted because a session with a higher priority gate must be established in the presence of insufficient bandwidth, then the AN MUST send a RSVP-PATH-ERR and/or PATH-RESV-ERR message for the session that is being pre-empted. This message SHOULD be sent as soon as possible. In response, the MTA SHOULD tear down the reservation and MAY notify the user of the pre-emption (e.g. play a special tone to phone user). The RSVP-PATH-ERR (or RSVP-RESV-ERR) message in this case MUST contain a RSVP-ERROR-SPEC object with an Error Code of 2 (Policy Control Failure) and an Error Value of 5 (Flow was pre-empted).

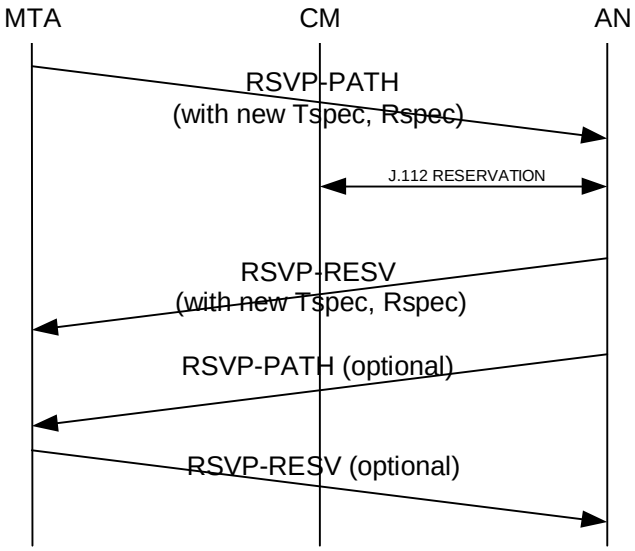


Figure 9: Reservation Change

6.5.3 Reservation Deletion

RSVP provides two messages for the explicit removal of Path and Reservation state, the RSVP-PATH-TEAR and RSVP-RESV-TEAR messages. To delete a reservation at the AN, the MTA SHOULD send a RSVP-PATH-TEAR message. To delete a reservation from RSVP-capable devices between the MTA and the AN, the MTA MAY send a RSVP-RESV-TEAR message. The format of these messages MUST be in accordance with RFC 2205, and MUST include both the Session-Object and Sender-Template to enable the AN to identify the proper gate.

If Path state and Reservation state are not periodically refreshed, they MUST time out. This is appropriate when a MTA crashes, for example. More details of refresh mechanisms appear in clause 6.5.4.

The AN MUST respond to a received RSVP-PATH-TEAR by sending a RSVP-RESV-TEAR to the MTA. The format of these messages MUST be as given in RFC 2205.

RSVP version 1 provides no means to ensure the reliable delivery of RSVP-PATH-TEAR and RSVP-RESV-TEAR messages, on the assumption that the state which they aim to delete will time out anyway. However, to avoid any delay in teardown (which causes short-term resource wastage and may cause overbilling), the message reliability extension to RSVP described in [Draft-ietf-mpls-lsp-tunnel-00] may be used.

6.5.4 Reservation Maintenance

RSVP has a soft-state model, in that reservation state times out if not periodically refreshed. This characteristic is retained in the segmented model described here. Since the entire reservation process in this model is initiated by the MTA, the MTA MUST periodically refresh all RSVP state information. The MTA MUST send RSVP-PATH messages as described in clause 6.5.1 within the time interval given by the AN in the RSVP-RESV Time-Values Object. The AN MUST generate RSVP-RESV messages towards the MTA on receipt of the RSVP-PATH (and a RSVP-PATH message as well if RSVP capable nodes have been detected as described in clause 6.5.1). This retains the soft state nature of RSVP, which retains its resiliency in the face of routing changes and node failures.

The MTA (or AN) MAY also implement RSVP Summary Refresh as another way to conserve upstream bandwidth when refreshing reservation state. It allows RSVP-capable nodes to "compress" their Path (or Resv) states for multiple reservations into single message. RSVP Refresh Overhead Reduction Extensions [Draft-ietf-rsvp-refresh-reduct-02] describes summary refresh as follows:

"The summary refresh extension enables the refreshing of RSVP state without the transmission of standard Path or Resv messages. The benefits of the described extension are that it reduces the amount of information that must be transmitted and processed in order to maintain RSVP state synchronization. Importantly, the described extension preserves RSVP's ability to handle non-RSVP next hops and to adjust to changes in routing. This extension cannot be used with Path or Resv messages that contain any change from previously transmitted messages, i.e. are trigger messages.

The summary refresh extension builds on the previously defined MESSAGE_ID extension. Only state that was previously advertised in Path and Resv messages containing MESSAGE_ID objects can be refreshed via the summary refresh extension.

The summary refresh extension uses the objects and the ACK message previously defined as part of the MESSAGE_ID extension, and a new Srefresh message. The new message carries a list of Message_Identifier fields corresponding to the Path and Resv trigger messages that established the state. The Message_Identifier fields are carried in one of three Srefresh related objects. The three objects are the MESSAGE_ID LIST object, the MESSAGE_ID SRC_LIST object, and the MESSAGE_ID MCAST_LIST object.

The MESSAGE_ID LIST object is used to refresh all Resv state, and Path state of unicast sessions. It is made up of a list of Message_Identifier fields that were originally advertised in MESSAGE_ID objects. The other two objects are used to refresh Path state of multicast sessions. A node receiving a summary refresh for multicast path state will at times need source and group information. These two objects provide this information. The objects differ in the information they contain and how they are sent. Both carry Message_Identifier fields and corresponding source IP addresses. The MESSAGE_ID SRC_LIST is sent in messages addressed to the session's multicast IP address. The MESSAGE_ID MCAST_LIST object adds the group address and is sent in messages addressed to the RSVP next hop.

The MESSAGE_ID MCAST_LIST is normally used on point-to-point links.

An RSVP node receiving an Srefresh message, matches each listed Message_Identifier field with installed Path or Resv state. All matching state is updated as if a normal RSVP refresh message has been received. If matching state cannot be found, then the Srefresh message sender is notified via a refresh NACK.

A refresh NACK is sent via the MESSAGE_ID_NACK object. As described in the previous clause, the rules for sending a MESSAGE_ID_NACK object are the same as for sending a MESSAGE_ID_ACK object. This includes sending MESSAGE_ID_NACK object both piggy-backed in unrelated RSVP messages or in RSVP ACK messages.

Complete details on how summary refresh works can be found in section 5 of RSVP Refresh Overhead Reduction Extensions [Draft-ietf-rsvp-refresh-reduct-02].

6.6 Definition of Commit Messages

This clause defines the Commit messages that MUST be generated by the MTA and MUST be supported by the AN.

Commit messages MUST be sent as UDP/IP datagrams with protocol number 17 (UDP). Each Commit message MUST occupy exactly one UDP/IP datagram. The destination IP address and port number in the UDP header MUST be as specified from the Commit-Entity Object returned in the RSVP-RESV message. The source port number MUST be the port on which the MTA will accept the acknowledgement message.

The Commit messages MUST consist of a Common-Header, followed by a variable number of variable-length objects. The Common Header MUST be as follows:

Version	Flags	Message Type	Message Checksum
Sent-TTL		(Reserved)	Message Length

Values of each field MUST be as specified in RFC 2205. Message types MUST be as follows:

COMMIT	240
COMMIT-ACK	241
COMMIT-ERR	242

Each object MUST consist of one or more 32-bit words, with a one-word header of the following format:

Length in bytes	Class-Number	C-Type
Object Contents …		

Values of each field MUST be as specified in RFC 2205.

The format of the COMMIT message and COMMIT-ACK message compliant with the present document MUST be as follows (items in italics are defined in the present document in clause 6.3, all others in RFC 2205 and/or RFC 2210):

<COMMIT-Message> ::= <Common-Header> <Session>
<Sender-Template> <*Gate-ID*>
[<Flowspec>] [<*Downstream_Flowspec*>]

<COMMIT-ACK-Message> ::= <Common-Header> <Session>
<Sender-Template><*Gate-ID*>

<COMMIT-ERR-Message> ::= <Common-Header> <Session>
<Sender-Template><*Gate-ID*><Error-Spec>

The Session and Sender-Template objects identify the sender and destination IP addresses and ports, and MUST be present. The Committed resources MAY be less than the total reserved resources (especially in a call-waiting or codec-change scenario), so that a Commit message MAY also contain a <Flowspec> object for each direction of the session. This provides a mechanism by which the size of the committed resources can be modified up or down as long as the amount of resources committed does not exceed the reserved resources. Note that a set of resources MAY be put on hold (frozen) by lowering the committed resources to zero while leaving the reserved resources in place. If either flowspec is omitted, the AN MUST set the amount of committed resource in that direction to equal to the amount of reserved resources.

6.7 Commit Operations

A significant aspect of the Dynamic QoS model is that reservation is a two-phase process, with a Commit phase following the Reserve phase. Clause 6.5 above described the Reserve phase, while this clause describes the Commit Phase and its relationship to the Reserve phase.

A conformant AN MUST perform all admission control and resource allocation functions on receipt of the original RSVP-PATH message, but MUST NOT allow access to those resources by the MTA until a COMMIT message is received, unless told otherwise in the GATE-SET parameters.

To perform a COMMIT the MTA MUST send a unicast message to the AN. This is desirable because the Commit phase only involves a MTA and a gate. The MTA learns the AN address and port number from the Commit-Entity object in the RSVP-RESV message.

Note that a COMMIT message differs in an important way from a standard RSVP message. It is sent directly from the MTA to the AN rather than hop-by-hop as an RSVP message would be. However, it contains objects that are syntactically the same as RSVP objects.

The AN MUST verify the value of Gate-ID, and verify the contents of the Session and Sender-Template objects match the previous reservation with the same value of Gate-ID, and that the Reverse-Session and Reverse-Sender-Template, if present, match the previous reservation with the same value of Gate-ID. The AN MUST acknowledge the receipt of a COMMIT with a COMMIT-ACK message or a COMMIT-ERR message.

When an MTA does not receive the acknowledgement within a timeout interval of Timer T4 (see annex C), the MTA MUST resend the COMMIT, up to a limit of seven attempts.

If the MTA desires to change the amount of committed resources within the reserved envelope, another COMMIT/COMMIT-ACK sequence is REQUIRED.

If the MTA desires to change the amount of reserved resources, then the RSVP-PATH/RSVP-RESV exchange MUST be repeated.

7 Authorization Interface Description (PKT-Q6)

This clause describes the interfaces between the AN and Gate Controller for purposes of authorizing the MTA to receive high Quality of Service. Signalling is required between the Gate Controller and AN to support gate management and IPCablecom QoS Admission Control Service. In addition, accurate subscriber billing requires the AN to indicate actual "committed" QoS resource usage on a per session basis. This clause describes the use of the COPS protocol to transport IPCablecom QoS defined messages between the Gate Controller and AN.

7.1 Gates: the Framework for QoS Control

A IPCablecom Dynamic QoS "Gate" is a policy control entity implemented at the AN to control access to enhanced QoS Services of a J.112 network by a single IP flow. Gates are unidirectional, in that a single gate controls access to a flow in either the upstream or downstream direction. Gates enable the creation of J.112 Flow Classifiers, which in turn control the routing of packets to J.112 Flows.

While a Gate also has a N-tuple just like a Classifier, it is not identical to a Classifier. The AN MUST set-up the Gate when a flow is authorized, until explicitly disabled to terminate the authorization for a flow. A J.112 Classifier MAY be set up and associated with a Gate. A Gate MAY exist before and after the Classifier it authorizes exists. A Gate MAY be considered to be associated with exactly zero, one, or two Classifiers.

An AN conforming to this Recommendation MUST NOT dynamically create a Classifier with a J.112 MAC message exchange unless it is authorized to do so by the existence of a Gate for that Classifier. An identifier, called the GateID is associated with Gates. The GateID, locally administered by the AN where the Gates exists, MAY be associated with one or more unidirectional Gates. For a point-to-point session, typically two unidirectional Gates exist, associated with a single GateID. In addition, J.112 Classifiers exist for each unidirectional flow that is established.

7.1.1 Classifier

A classifier is a six-tuple:

- Direction (Upstream/Downstream).
- Protocol.
- Source IP.
- Destination IP.
- Destination Port.
- Source Port.

If there is an upstream and an associated (part of the same session) downstream flow, then there MUST exist separate classifiers for the upstream flow and the downstream flow. The Classifier is updated by the RSVP message for the reservation performed for the upstream and downstream flows. The session data flow MUST match the classifier to receive the Quality of Service associated with the RSVP reservation. Future reservations can change the classifier.

7.1.2 Gate

A Gate is associated with a unidirectional flow, and comprises the following:

- Gate-ID.
- Prototype Classifier.
- Various flag bits described below.
- Authorized Envelope (Flow Spec).
- Reserved Envelope (Flow Spec).
- Resource-ID.

The GateID (described below) is a local 32-bit identifier that is allocated from the local space at the AN where the Gate resides. Up to two gates MAY share the same Gate-ID. Typically, a Gate-ID will identify a single upstream flow and a single downstream flow, and correspond to a single multimedia session. [This does not preclude bidirectional implementations, however.]

The Prototype Classifier consists of the same six elements as a Classifier, as described above. The Source IP is the IP address (as seen at the AN) of the originator of the flow. In the case of an upstream Gate on the J.112 channel, the Source IP is the IP address of the local MTA. For the downstream flow, the Source IP address is the IP address of the remote MTA. For selected parameters of a Gate's prototype classifier, a wild card is allowed. In Multimedia call signalling, the source UDP Port is not signalled, so its value is not considered to be part of a Gate's information.

The Source Port MAY be wild-carded, to support both IPCablecom Call Signalling Protocols (DCS and TS 101 909-4). If the Source Port is wild-carded, its value in the Gate parameters will be zero.

The Source IP address MAY be wild-carded, to support the TS 101 909-4 Call Signalling Protocol. If the Source IP address is wild-carded, its value in the Gate parameters will be zero.

The Auto-Commit flag, when set, causes resources to be committed immediately upon reservation. For a telephony application, this will typically be used for the downstream gate at the originator of a call when the destination is a PSTN gateway. When the originating MTA makes the resource reservation, the downstream flow is enabled so that remote ringback, call progress tones, and announcements may be heard by the call originator. See clause 7.1.4 for further description.

The Commit-Not-Allowed flag, when set, causes the AN to ignore any COMMIT messages for this gate. This facility may be used by a Gate Controller when the remote endpoint address is not yet known, and therefore specified as a wildcard in the prototype classifier. In such an application, the Gate Controller typically updates the gate's prototype classifier prior to the MTA issuing the COMMIT message; use of this flag prevents various theft-of-service scenarios.

The Authorized and Reserved Envelopes are RSVP Flow Specs (both T-Spec and R-Spec) as described in the earlier clauses.

A reservation request for resources (as specified in the PATH message or equivalent J.112 MAC message) MUST be checked against what has been authorized for the Gate-ID associated with the direction for the resource request. The resources authorized are specified in the Authorized envelope. Also checked is the wild-card in the Gate for particular entries.

The Resource-ID is a local 32-bit identifier that is allocated from the local space at the AN where the Gate resides. Any number of gates MAY share a resource-ID, and therefore share a common set of resources, with the restriction that only one of these gates in each direction have resources committed.

7.1.3 Gate Identification

A GateID is a unique identifier that is locally allocated by the AN where the Gate resides. The GateID is a 32-bit identifier. A GateID MAY be associated with one or more Gates. In both the TS 101 909-4 and DCS call signalling protocols, a Gate-ID is associated with each call leg, and consists of a single upstream gate and a single downstream gate.

A Gate-ID MUST be associated with the following information:

- One or two Gates, which MUST be one of the following combinations:
- Single upstream gate.
- Single downstream gate.
- Single upstream gate and a single downstream gate [typically this would be a bidirectional implementation].
- Gate Co-ordination information.
- Address: Port of the remote AN (or other entity) with which to co-ordinate resource allocation for this set of gates.
- Gate-ID assigned at the remote AN (or other entity) for the remote set of gates.
- Security key for communication with the remote AN (or other entity).
- No-Gate-Co-ordination flag, which when set, causes the AN to skip the gate co-ordination, i.e. not require receipt of a Gate-Open message from the remote AN (or other entity).
- No-Gate-Open flag, which when set, causes the AN to not send a Gate-Open message to the remote AN (or other entity).
- Accounting and Billing information.
- Address: Port of the Primary Record-Keeping-Server that should receive event records.
- Address: Port of the Secondary Record-Keeping-Server, for use if the primary is unavailable.
- Flag indicating whether the Event Messages are to be sent to the Record Keeping Server in real-time, or whether they are to be batched and sent at periodic intervals.
- Billing-Correlation-ID, which will be passed to the Record-Keeping-Server with each QoS-Start/QoS-Stop event record.
- Additional billing information, if supplied, which will be used to generate Call-Answer and Call-Disconnect event messages.

The Gate-ID MUST be unique among all current gates allocated by the AN. The value of the 32-bit quantity SHOULD NOT be chosen from a set of small integers, since possession of the GateID value is a key element in the authentication of the COMMIT messages from the MTA. An algorithm that MAY be used to assign values of GateIDs is as follows: partition the 32-bit word into two parts, an index part, and a random part. The index part identifies the gate by indexing into a small table, while the random part provides some level of obscurity to the value.

The No-Gate-Open flag, and the No-Gate-Co-ordination flag, combine to offer the Gate Controller flexibility for connections to non-ANs, to non-compliant ANs, or to non-IPCablecom systems. The NCS Call Agent will typically provide its own address as the remote AN address, and set the No-Gate-Open flag. Upon call completion, the Call Agent will generate a Gate-Open message and send it to the AN; this starts Timer T2 (see clause 7.1.4) and forces the MTA to Commit the resources. On call termination due to various errors (where the MTA is unable to indicate this event), the Call Agent receives hangup notification via the Gate-Close message. Use of the No-Gate-Open flag reduces the processing load on the NCS Call Agent without loss of functionality.

The No-Gate-Co-ordination flag is typically used when the remote endpoint is not a IPCablecom-compliant system, and is not able to perform the gate co-ordination procedures. When combined with the No-Gate-Open flag, it causes the gate to function independently of the other endpoint. See clause 7.1.4 for further details on the effect of these two flag bits on the state transition diagram.

7.1.4 Gate Transition Diagram

Gates are considered to have the following states:

- Allocated - the initial state of a gate created at the request of the GC.
- Authorized - GC has authorized the flow with resource limits defined.
- Reserved - resources have been reserved for the flow.
- Committed - resources are being used.
- Remote-Committed and Local-Committed - transient states that exist as a gate proceeds through the gate co-ordination protocol with the remote gate.

The AN MUST support gate states and transitions as shown in and described in this clause. All gates assigned the same Gate-ID by the AN MUST transition together through the states shown in figure 10.

A gate is created in the AN by either a Gate-Alloc command or a Gate-Set command from the GC. In both cases, the AN allocates a locally unique identifier called a Gate-ID, which is returned to the GC. If the gate was created by a Gate-Set message, then the AN MUST mark the gate in state "Authorized" and MUST start Timer T1. If the gate was created by a Gate-Alloc message, then the AN MUST mark the gate in state "Allocated", start Timer T0, and MUST wait for a Gate-Set command, at which point the gate MUST be marked in state "Authorized". If the Timer T0 expires with the gate in state "Allocated" or Timer T1 expires with the gate in state "Authorized", then the AN MUST delete the gate. Timer T0 limits the amount of time the Gate-ID will remain valid without any specified gate parameters. Timer T1 limits the amount of time the authorization will remain valid.

A gate in the "Authorized" state is expecting the MTA to attempt to reserve resources. The MTA does this with either a RSVP-PATH message or via the MAC-layer Interface. On receipt of this reserve request, the AN MUST verify the request is within the limits established for the gate, and perform admission control procedures.

The AN MUST implement at least two admission control policies, one for normal voice communications and one for emergency communications. These two policies MUST have provisionable parameters that specify, at a minimum, 1) a maximum amount of resources that may be allocated non-exclusively to sessions of this type (which may be 100 % of the capacity), 2) the amount of resources that may be allocated exclusively to sessions of this type (which may be 0 % of the capacity), and 3) the maximum amount of resources that may be allocated to sessions of the two types. The admission control policy MAY also specify whether a new session of that type may "borrow" from lower priority classes or should pre-empt an existing session of some other type to satisfy the admission control policy settings.

If the admission control procedures are successful, the gate MUST be marked in the "Reserved" state. Otherwise, the gate stays in the "Authorized" state. Note that the actual reservation made by the MTA may be for less than that authorized, e.g. reservation for upstream only when a pair of gates were established authorizing upstream and downstream flows. If the individual gate was marked with the "Auto-Commit" flag, then the resources reserved are immediately committed, but the state of the gate is unchanged.

In the "Reserved" state the gate is expecting the MTA to Commit to the resources. The Commit command from the MTA is either a unicast UDP message, or an equivalent request via the MAC-layer Interface. The Commit is normally synchronized with the remote gate, via gate co-ordination messages; unless both endpoint clients issue the Commit commands nearly simultaneously, the authorization will be withdrawn. If the gate is still in the "Reserved" state and Timer T1 expires (i.e. the MTA does not issue the Commit command), the AN MUST release any resources reserved,

and delete the gate. If the Commit-Not-Allowed flag is set when the Commit command is received, the AN MUST respond with Commit-Err and MUST NOT change the state of the gate.

If, in the "Reserved" state, the AN receives a Commit command from the MTA, and the No-Gate-Co-ordination flag is set, then the AN MUST mark the gate in the "Committed" state and stop Timer T1. Unless the No-Gate-Open flag is set, the AN MUST initiate a Gate-Open message to the gate-co-ordination entity.

If, in the "Reserved" state, the AN receives a Commit command from the MTA, and the No-Gate-Co-ordination flag is not set, then the AN MUST mark the gate in the "Local-Committed" state and start Timer T2. Unless the No-Gate-Open flag is set, the AN MUST initiate a Gate-Open message to the gate-co-ordination entity. Timer T2 limits the amount of time a gate may have committed resources on one end and not on the other end.

In the "Local-Committed" state the gate has committed the local resources but is waiting for the remote endpoint client to commit resources at that end. If either Timer T1 or T2 expires in this state, the AN MUST deactivate all resources committed with this gate, release all resources reserved with this gate, initiate a Gate-Close message (only if the gate has been opened) with the gate-co-ordination entity, and delete the gate.

If, in the "Local Committed" state, the AN receives a Gate-Open message from the gate-co-ordination entity, the AN MUST stop timers T1 and T2, and MUST mark the gate in the "Committed" state.

If, in the "Reserved" state, the AN receives a Gate-Open message from the gate-co-ordination entity, the AN MUST mark the gate in the "Remote-Committed" state, and start Timer T2.

In the "Remote-Committed" state the gate has been notified that the far end MTA has activated resources, but the local MTA has not. If either Timer T1 or T2 expires in this state, the AN MUST release all resources reserved with this gate, initiate a Gate-Close message with the gate-co-ordination entity, and delete the gate. If the Commit-Not-Allowed flag is set when the Commit command is received, the AN MUST respond with Commit-Err and MUST NOT change the state of the gate.

If, in the "Remote-Committed" state, the AN receives a Commit command from the MTA, then the AN MUST stop timers T1 and T2, and MUST mark the gate in the "Committed" state. Unless the No-Gate-Open flag is set, the AN MUST initiate a Gate-Open message to the gate-co-ordination entity.

Once in the "Committed" state, the gate has reached a stable configuration and has no timers pending nor timeout actions to perform. Resources have been committed at both this gate and the corresponding gate at the remote entity. Resources will continue to be committed until either the local MTA issues a Release command, or the remote gate signals a desire to terminate the resources.

If, in the "Committed" state, the AN receives a Gate-Close message from the gate-co-ordination entity, the AN MUST deactivate all resources committed for the local MTA, release all resources reserved, and delete the gate.

If, in the "Committed" state, the AN receives a Release command from the MTA, either in the form of a RSVP-PATH-TEAR message or via the MAC-layer interface, or from a failure of the client to refresh a reservation, or from internal J.112 mechanisms that detect a client failure, the AN MUST deactivate all resources committed for the MTA, release all resources reserved, initiate a Gate-Close message to the gate-co-ordination entity, and delete the gate.

While in the "Committed" state, the AN MUST allow the MTA to initiate changes in the resource reservation or commitment, within the limits of the authorization and local admission control.

7.1.5 Gate Co-ordination

In addition to controlling the local J.112 Flow classification function, Gates MUST communicate with their remote counterparts for the same flow in order to confirm that the far side has also committed to billing for the session. This is required to avoid several theft-of-service scenarios, as described in annex L. The protocol for this communication is given in clause 8.

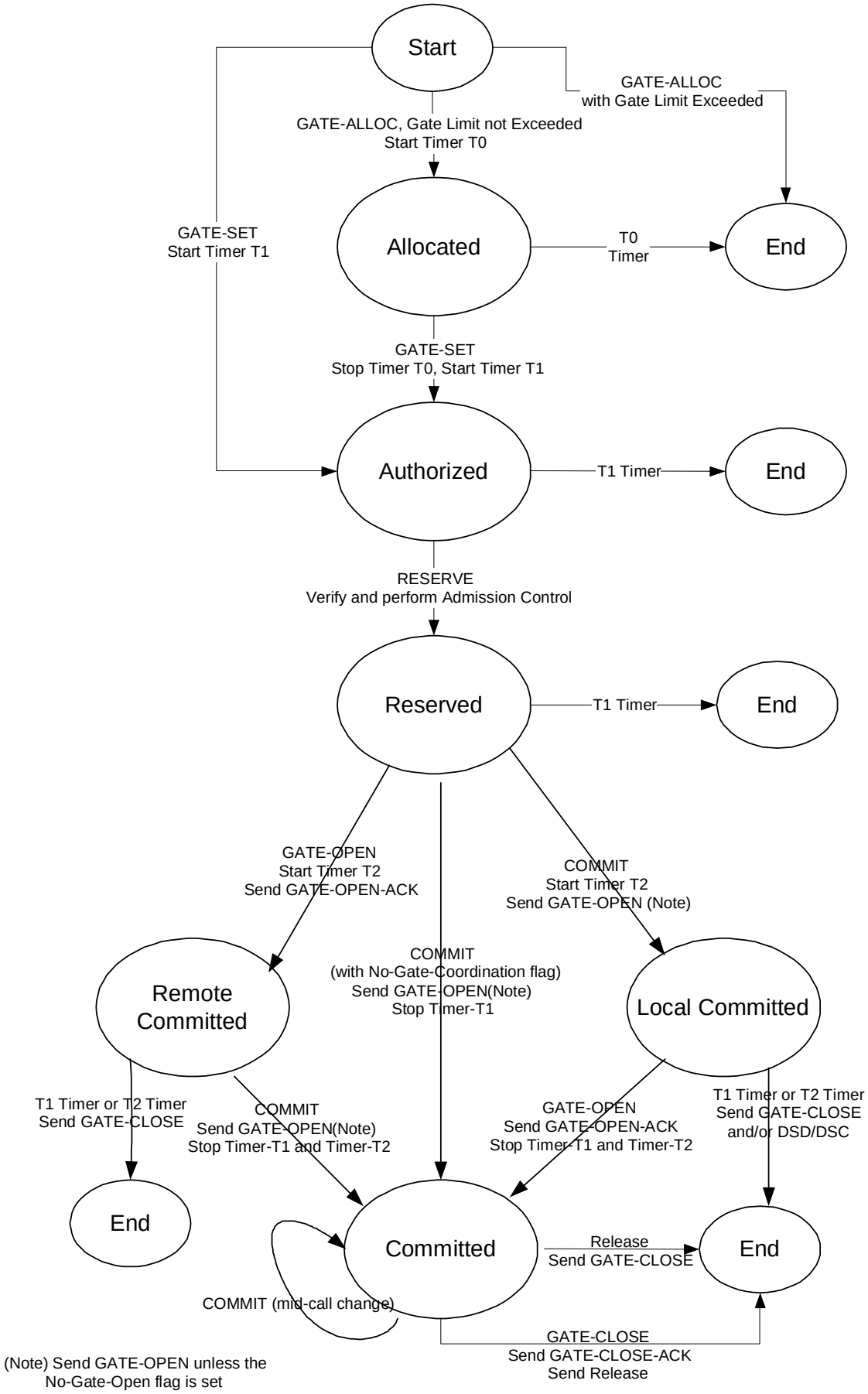


Figure 10: Gate State Transition Diagram

7.2 COPS Profile for IPCablecom

IPCablecom QoS Admission Control is the act of managing QoS resource allocation based on administrative policies and available resource. IPCablecom QoS Admission Control Service uses a client/server architecture. The high level operational modules are depicted in figure 11. The administrative policies are stored as policy database and controlled by the COPS Server. While a typical Intserv implementation of COPS has the server determine available resources, a Diffserv implementation pushes the policy into the client so that the client can make admission control decisions.

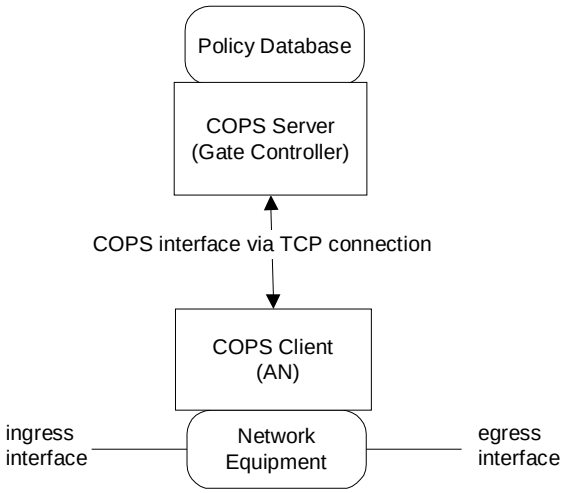


Figure 11: QoS Admission Control Layout

The QoS Admission Control decisions made by the COPS Server MUST be passed to the COPS Client using COPS. The COPS Client MAY make QoS Admission Control requests to the COPS Server based on network events triggered by either the QoS signalling protocol, or via data flow detection mechanisms. The network event can also be the need of QoS bandwidth management, e.g. a new QoS capable interface becomes operational.

QoS policy decisions made by the COPS Server MAY be pushed to the COPS Client based on an external, out-of-band, QoS service request, e.g. request from the terminating AN or a Gate Controller. These policy decisions MAY be stored by the COPS client in a local policy decision point and the AN may access that decision information to make admission control decisions on incoming session requests received at the AN.

The COPS Client-COPS Server interaction support for QoS Admission Control is provided by IETF's COPS protocol. The COPS protocol includes the following operations:

- Client-Open (OPN)/Client-Accept (CAT)/Client-Close (CC). The COPS Client sends an OPN message to initiated a connection with the COPS Server, and the Server responds with a CAT message to accept the connection. The server sends a CC message to terminate the connection with the Client.
- Request (REQ). The COPS Client sends a REQ message to the server to request admission control decision information or device configuration information. The REQ message may contain client-specific information that the server uses, together with data in the session admission policy database, to make policy-based decisions.
- Decision (DEC). The server responds to REQs by sending a DEC back to the client that initiated the original request. DEC messages may be sent immediately in response to a REQ (i.e. a solicited DEC) or at any time after to change/update a previous decision (i.e. an unsolicited DEC).
- Report State (RPT). The COPS Client sends a RPT message to the COPS Server indicating changes to the request state in the COPS Client. The COPS Client sends this to inform the COPS Server the actual resource reserved after the COPS Server has granted admission. The COPS Client can also use Report to periodically inform the COPS Server the current state of the COPS Client.
- Delete Request State (DEL). The COPS Client sends a DEL message to the COPS Server for request state cleanup. This can be the result of QoS resource release by the COPS Client.
- Keep Alive (KA). Sent by both the COPS Client and COPS Server for communication failure detection.

- Synchronize State Request (SSR)/Synchronize State Complete (SSC). SSR is sent by the COPS Server requesting current COPS Client state information. The client re-issues request queries to the server to perform the synchronization, and then the client sends a SSC message to indicate synchronization is complete. Because the GC is stateless, the SSR/SSC operations are of no importance in IPCablecom and are not used by the AN or GC.

Within the IPCablecom architecture, the Gate Controller is a COPS Policy Decision Point (i.e. PDP) entity and the AN is the COPS Policy Enforcement Point (i.e. PEP) entity.

The details of the COPS protocol are provided in draft-ietf-rap-cops-07 (see Bibliography). This IETF RFC provides a description of the base COPS protocol, independent of client type. Additional drafts provide information for using COPS for Integrated Services with RSVP and for Differentiated Services (i.e. provisioning clients). A more detailed overview of the COPS protocol is provided as information in annex M.

7.3 Gate Control Protocol Message Formats

Protocol messages for Gate Control are transported within the COPS protocol messages. COPS utilizes a TCP connection established between the AN and the Gate Controller, and will use the mechanisms specified in standards under development to secure the communication path.

7.3.1 COPS Common Message Format

Each COPS message consists of the COPS header followed by a number of typed objects. The GC and AN MUST support COPS messaging as defined below.

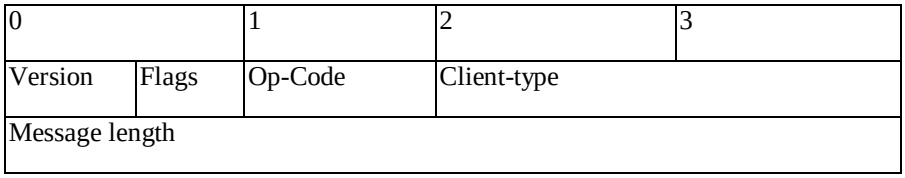


Figure 12: Common COPS Message Header

Version is a 4-bit field giving the current COPS version number. This MUST be set to 1.

Flags is a 4-bit field. 0x1 is the solicited message flag. When a COPS message is sent in response to another message (e.g. a solicited decision sent in response to a request) this flag MUST be set to 1. In other cases (e.g. an unsolicited decision) the flag MUST NOT be set (value = 0). All other flags MUST be set to zero.

Op-code is a 1-byte field that gives the COPS operation to be performed. COPS operations used in this IPCablecom specification are:

1 = Request (REQ)

2 = Decision (DEC)

3 = Report-State (RPT)

6 = Client-Open (OPN)

7 = Client-Accept (CAT)

9 = Keep-Alive (KA)

Client type is a 16-bit identifier. For IPCablecom use the Client type MUST be set to IPCablecom client (0x8005). For Keep-Alive messages (Op-code = 9) the client-type MUST be set to zero, as the KA is used for connection verification rather than per client session verification.

Message length is a 32-bit value giving the size of the message in octets. Messages MUST be aligned on 4-byte boundaries, so the length MUST be a multiple of four.

Following the COPS common header are a variable number of objects. All the objects follow the same object format; each object consists of one or more 32-bit words with a four-octet header, using the following format:

0	1	2	3
Length		C-Num	C-type
(Object contents)			

Figure 13: Common COPS Object Format

The length is a two-octet value that MUST give the number of octets (including the header) that compose the object. If the length in octets is not a multiple of four, padding MUST be added to the end of the object so that it is aligned to the next 32-bit boundary. On the receiving side, a subsequent object boundary MUST be found by rounding up the previous stated object length to the next 32-bit boundary.

C-Num identifies the class of information contained in the object, and the C-Type identifies the subtype or version of the information contained in the object. Standard COPS objects (as defined in draft-ietf-rap-cops-07, see Bibliography) used in the present document, and their values of C-num, are:

1 = Handle

6 = Decision

8 = Error

9 = Client Specific Info

10 = Keep-Alive-Timer

11 = PEP Identification

7.3.2 Additional COPS Objects for Gate Control

As with the COPS-PR and COPS-RSVP client types, the IPCablecom client type defines a number of object formats. These objects MUST be placed inside a Decision object, C-Num = 6, C-Type = 4 (Client specific Decision Data) when carried from GC to AN in a decision message. They MUST also be placed in a ClientSI object, C-Num = 9, C-Type = 1 (Signalled Client SI) when carried from AN to GC in a Report message. They are encoded similarly to the client-specific objects for COPS-PR; detailed encodings appear below. As in COPS-PR, these objects are numbered using a client-specific number space, which is independent of the top-level COPS object number space. For this reason, the object numbers and types are given as S-Num and S-Type respectively.

Additional COPS objects defined for use by IPCablecom are as follows.

7.3.2.1 Transaction-ID

The Transaction-ID contains a token that is used by the GC to match responses from the AN to the previous requests, and the command type that identifies the action to be taken or response.

Length = 8	S-Num = 1	S-Type = 1
Transaction Identifier	Gate Command Type	

Transaction Identifier is a 16-bit quantity that MAY be used by the GC to match responses to commands.

Gate Command Type MUST be one of the following:

GATE-ALLOC 1

GATE-ALLOC-ACK 2

GATE-ALLOC-ERR 3

GATE-SET 4

GATE-SET-ACK 5

GATE-SET-ERR 6

GATE-INFO 7

GATE-INFO-ACK 8

GATE-INFO-ERR 9

GATE-DELETE 10

GATE-DELETE-ACK 11

GATE-DELETE-ERR 12

7.3.2.2 Subscriber-ID

The Subscriber-ID identifies the subscriber for this service request. Its main use is to prevent various denial-of-service attacks.

Length = 8	S-Num = 2	S-Type = 1
IP v4 address (32-bits)		

or

Length = 20	S-Num = 2	S-Type = 2
IP v6 address (128-bits)		

7.3.2.3 Gate-ID

This object identifies the gate or set of gates referenced in the command message, or assigned by the AN for a response message.

Length = 8	S-Num = 3	S-Type = 1
Gate-ID (32-bits)		

7.3.2.4 Activity-Count

When used in a GATE-ALLOC message, this object specifies the maximum number of gates that can be simultaneously allocated to the indicated subscriber-ID. This object returns, in a GATE-SET-ACK or GATE-ALLOC-ACK message, the number of gates assigned to a single subscriber. It is useful in preventing denial-of-service attacks.

Length = 8	S-Num = 4	S-Type = 1
Count (32-bits)		

7.3.2.5 Gate-spec

Length = 60 or 88 or 116, etc.		S-Num = 5	S-Type = 1
Direction	Protocol ID	Flags, defined below	Session Class
Source IP Address (32-bits)			
Destination IP Address (32-bits)			
Source Port (16-bits)		Destination Port (16-bits)	
DS Field	Reserved	Reserved	Reserved
Timer T1 value			
Timer T2 value			

Token Bucket Rate [r] (32-bit IEEE floating point number)	Flow spec alt #1
Token Bucket Size [b] (32-bit IEEE floating point number)	
Peak Data Rate (p) (32-bit IEEE floating point number)	
Minimum Policed Unit [m] (32-bit integer)	
Maximum Packet Size [M] (32-bit integer)	
Rate [R] (32-bit IEEE floating point number)	
Slack Term [S] (32-bit integer)	
Additional sets of r, b, p, m, M, R, and S values, as needed, to describe the authorization	Flow spec alt #2, etc.

Direction is either 0 for a downstream gate, or 1 for an upstream gate.

Protocol-ID is the value to match in the IP header, or zero for no match.

Flags are defined as follows:

0x01 Auto-Commit, if set, causes resources to be committed immediately upon reservation.

0x02 Commit-Not-Allowed, if set, causes the AN to ignore any COMMIT messages for this gate.

The rest are reserved and MUST be zero.

Session class identifies the proper admission control policy or parameters to be applied for this gate. Permissible values are:

0x00 Unspecified.

0x01 Normal priority VoIP session.

0x02 High priority VoIP session (e.g. E911).

All other values are currently reserved.

Source IP Address and Destination IP Address are a pair of 32-bit IPV4 addresses, or zero for no match (i.e. a wildcard specification that will match any request from the MTA).

Source Port and Destination Port are a pair of 16-bit values, or zero for no match

The values of r, b, p, m, M, R, and S, are as described in clause 6.2. The Gate-Spec MAY contain multiple sets of these values to specify complex authorizations (as described in clause 6.2.1).

The DS field is defined by the following structure:

0	1	2	3	4	5	6	7
Differentiated Services Code Point (DSCP)						Not Used	Not Used

For backward compatibility with current system implementations and use of the IP Precedence as defined in RFC 2474 and RFC 791, the appropriate bits of the IPv4 TOS byte shown below MAY be inserted in the DS field. The IP TOS field (bits 3-6) is not supported in Diffserv networks.

0	1	2	3	4	5	6	7
IP Precedence			IPv4 IP TOS				Not Used

Timer T1 and Timer T2 are values in milliseconds, and used in the Gate Transition Diagram described in clause 8.1.4. If multiple Gate-Spec objects appear in a single COPS message, the values of T1 and T2 MUST be identical in all Gate-Spec occurrences.

7.3.2.6 Remote-Gate-Info

Length	S-Num = 6	S-Type = 1
AN IP Address (32-bits)		
AN Port (16-bits)	Flags, defined below	
Remote Gate-ID		
Algorithm	Security Key	

AN-IP-Address is the address of the remote AN with whom Gate Co-ordination is to be done.

AN-Port is the port number for the messages sent for gate co-ordination. If the port number is not available to the gate controller, it is set to zero. A value of zero causes the AN to ignore this field.

Flags are defined as follows:

0x0001 No-Gate-Co-ordination, if set, causes gate co-ordination to be skipped. AN will not require receipt of a Gate-Open from remote entity.

0x0002 No-Gate-Open, if set, causes AN to skip sending of the Gate-Open message when a Commit is processed.

The rest are reserved and MUST be zero

Remote-Gate-ID is the Gate-ID assigned by the remote AN for the gate or set of gates.

Algorithm is a 1-byte field that currently can be set to the following decimal values:

100 = MD5-based MAC, as specified by Radius in RFC 2138.

Additional choices for an authentication algorithm may be added in future versions of the present document.

Security key is a variable length key used in producing the authentication check in the gate co-ordination messages. The length of the key is 17 less than the length of the object.

7.3.2.7 Event-Generation-Info

The object contains all the information necessary to support the QoS-Start and QoS-Stop event messages as specified and required in [J.em].

Length = 36	S-Num = 7	S-Type = 1
Primary-Record-Keeping-Server-IP-Address (32-bits)		
Primary-Record-Keeping-Server-Port	Flags, see below	Reserved
Secondary-Record-Keeping-Server-IP-Address (32-bits)		
Secondary-Record-Keeping-Server-Port	Reserved	
Billing-Correlation-ID (16 bytes)		

Primary-Record-Keeping-Server-IP-Address is the address of the record keeper to whom event records are sent.

Primary-Record-Keeping-Server-Port is the port number for event records sent.

Flag values are as follows:

0x01 Batch processing indicator. If set, the AN MUST accumulate event records as part of a batch file and send to Record Keeping Server at periodic intervals. If clear, the AN MUST send the event records to the Record Keeping Server in real-time.

The rest are reserved and MUST be zero.

Secondary-Record-Keeping-Server-IP-Address is the address of the secondary record keeper to whom records are sent if the primary record keeping server is unavailable.

Secondary-Record-Keeping-Server-Port is the port number for event records sent.

Billing-Correlation-ID is the identifier assigned by the CMS for all records related to this session.

7.3.2.8 Media-Connection-Event-Info

The object contains all the information necessary to support the Call-Answer and Call-Disconnect event messages. If this object is present in the GATE-SET command, then the AN MUST generate the Call-Answer and Call-Disconnect event messages.

Length = 84	S-Num = 8	S-Type = 1
Called-Party-Number		
		Reserved
Routing-Number		
		Reserved
Charged-Number		
		Reserved
Location-Routing-Number		
		Reserved

7.3.2.9 IPCablecom-Error

A client-specific error object is defined as follows:

Length = 8	S-Num = 9	S-Type = 1
Error-code	Error Sub-code	

The Error-code values defined in the present document are:

1 = No gates currently available.

2 = Illegal Gate-ID.

3 = Illegal Session Class value.

4 = Subscriber exceeded gate limit.

127 = Other, unspecified error.

The Error Sub-code is reserved for future use.

7.3.2.10 Electronic-Surveillance-Parameters

Length = 20	S-Num = 10	S-Type = 1
DF-IP-Address-for-CDC (32 bits)		
DF-Port-for-CDC (16 bits)	Flags, defined below	
DF-IP-Address-for-CCC (32 bits)		
DF-Port-for-CCC (16 bits)	Reserved	

DF-IP-Address-for-CDC is the address of the Electronic Surveillance Delivery Function to whom the duplicated event messages are to be sent.

DF-Port-for-CDC is the port number for the duplicated event messages.

Flags are defined as follows:

0x0001 DUP-EVENT. If set, AN MUST send a duplicate copy of all event messages related to this gate (e.g. QoS-Start, QoS-Stop, and possibly Call-Answer and Call-Disconnect) to the DF-IP-Address-for-CDC.

0x0002 DUP-CONTENT. If set, AN MUST send a duplicate copy of all packets matching the classifier(s) for this gate to the DF-IP-Address-for-CCC.

The rest are reserved and MUST be zero.

DF-IP-Address-for-CCC is the address of the Electronic Surveillance Delivery Function to whom the duplicated call content packets are to be sent.

DF-Port-for-CCC is the port number for the duplicated call content.

7.3.2.11 Session-Description-Parameters

Length =	S-Num = 11	S-Type = 1
SDP-strings		

SDP-strings is the Session Description (SDP) of the upstream packet stream, followed by a NULL octet, followed by the Session Description (SDP) of the downstream packet stream. Sufficient padding of NULL octets is appended to make the total length a multiple of four octets.

If this object is present in the Gate-Set message, then the AN MUST include this information in the QoS-Start event message.

7.3.2.12 Gate-Co-ordination-Port

This object contains the UDP port number, which is used by an AN to listen for incoming gate co-ordination messages.

Length = 8	S-Num = 12	S-Type = 1
AN port (16 bits)	Reserved	

This object would normally be included in the GATE-ALLOC-ACK message, sent by an AN in response to a GATE-ALLOC. However, if a GATE-SET message is used to allocate a gate instead of GATE-ALLOC, this object must be present in the GATE-SET-ACK message.

7.3.3 Definition of Gate Control Messages

Messages that perform gate control between the GC and AN MUST be defined and formatted as follows. Note that messages from GC to AN are COPS Decision messages, and messages from AN to GC are COPS Report messages.

```
<Gate-Control-Cmd>      := <COPS-Common-Header> <Handle>
                             <Context> <Decision Flags>
                             <ClientSI-Data>

<ClientSI-Data          := <Gate-Alloc> | <Gate-Set> | <Gate-Info>> |
                             <Gate-Delete>

<Gate-Control-Response> := <COPS-Common-Header> <Handle>
                             <Report-Type> <ClientSI-Object>

<ClientSI-Object>       := <Gate-Alloc-Ack> | <Gate-Alloc-Err> |
                             <Gate-Set-Ack> | <Gate-Set-Err> |
                             <Gate-Info-Ack> | <Gate-Info-Err> |
                             <Gate-Delete-Ack> | <Gate-Delete-Err>

<Gate-Alloc>            := <Decision-Header> <Transaction-ID> <Subscriber-ID>>
                                              [<Activity-Count>]

<Gate-Alloc-Ack>        := <ClientSI-Header> <Transaction-ID> <Subscriber-ID>
                             <Gate-ID> <Activity-Count>>
                             <Gate-Co-ordination-Port>

<Gate-Alloc-Err>        := <ClientSI-Header> <Transaction-ID> <Subscriber-ID>
                             <IPCablecom-Error>

<Gate-Set>              := <Decision-Header> <Transaction-ID> <Subscriber-ID>
                           [<Activity-Count>] [<Gate-ID>]
                           [<Remote-Gate-Info>]
                           [<Event-Generation-Info>]
                           [<Media-Connection-Event-Info>]
                           [<Electronic-Surveillance-Parameters>]
                           [<Session-Description-Parameters>]
                           <Gate-Spec> [<Gate-Spec>]

<Gate-Set-Ack>          := <ClientSI-Header> <Transaction-ID> <Subscriber-ID>
                             <Gate-ID> <Activity-Count>
                             [<Gate-Co-ordination-Port>]

<Gate-Set-Err>          := <ClientSI-Header> <Transaction-ID> <Subscriber-ID>
                             <IPCablecom-Error>

<Gate-Info>             := <Decision-Header> <Transaction-ID> <Gate-ID>

<Gate-Info-Ack>         := <ClientSI-Header> <Transaction-ID> <Subscriber-ID>
                             <Gate-ID> [<Remote-Gate-Info>]
                             [<Event-Generation-Info>]
                             [<Media-Connection-Event-Info>]
                             <Gate-Spec> [<Gate-Spec>]

<Gate-Info-Err>         := <ClientSI-Header> <Transaction-ID> <Gate-ID>
                             <IPCablecom-Err>

<Gate-Delete>           := <Decision-Header> <Transaction-ID> <Gate-ID>

<Gate-Delete-Ack>       := <ClientSI-Header> <Transaction-ID> <Gate-ID>
```

```
<Gate-Delete-Err>       := <ClientSI-Header> <Transaction-ID> <Gate-ID>
                             <IPCablecom-Err>
```

The Context object (C-NUM = 2, C-TYPE = 1) in the COPS Decision message has the R-Type (Request Type Flag) value set to 0x08 (Configuration Request) and the M-Type set to zero. The Command-Code field in the mandatory Decision-Flags object (C-NUM = 6, C-TYPE = 1) is set to 1 (Install Configuration). Other values should cause the AN to generate a Report message indicating failure. The Report-Type object (C-NUM = 12, C-TYPE = 1) included in the COPS Report message has the Report-Type field set to 1 (Success) or 2 (Failure) depending on the outcome of the gate control command. All Report messages carrying the gate control response should have the solicited message flag bit set in the COPS header.

7.4 Gate Control Protocol Operation

7.4.1 Initialization Sequence

When the AN (i.e. COPS PEP) boots, it listens for TCP connections on port 2 126 (assigned by IANA). Any Gate Controller with a need to contact the AN MUST establish a TCP connection to the AN on that port. It is expected that multiple Gate Controllers will establish COPS connections with a single AN. When the TCP connection between the AN and GC is established, the AN sends information about itself to the GC in the form of a CLIENT-OPEN message. This information includes the provisioned AN-ID in the PEP Identification (PEPID) object. The AN SHOULD omit the Last PDP Address (LastPDPAddr) object from the CLIENT-OPEN message.

In response, the Gate Controller sends a CLIENT-ACCEPT message. This message includes the Keep-Alive-Timer object, which tells the AN the maximum interval between Keep-Alive messages.

The AN then sends a REQUEST message, including the Handle and Context objects. The Context object (C-NUM = 2, C-TYPE = 1) MAY have the R-Type (Request Type Flag) value set to 0x08 (Configuration Request) and M-Type set to zero. The Handle object contains a number, that is chosen by the AN. The only requirement imposed on this number is that an AN MUST NOT use the same number for two different REQUESTs on a single COPS connection; in the IPCablecom environment the handle has no other protocol significance. This completes the initialization sequence, which is shown in figure 14.

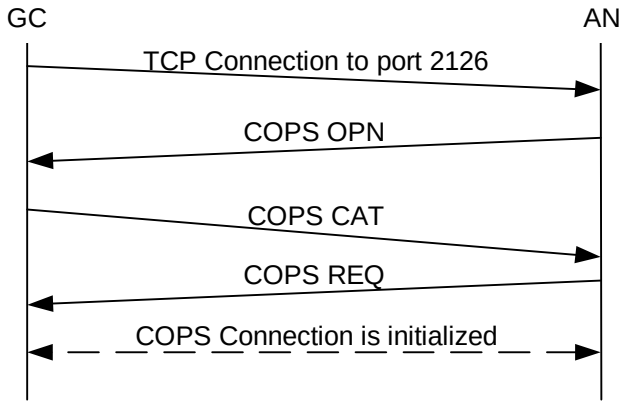


Figure 14: COPS Connection establishment

Periodically the AN MUST send a COPS KEEP-ALIVE (KA) message to the GC. Upon receipt of the COPS KA message, the GC MUST echo a COPS KA message back to the AN. This transaction is shown in figure 15 and is fully documented in RFC 2748. This MUST be done at least as often as specified in the Keep-Alive-Timer object returned in the CLIENT-ACCEPT message. The KEEP-ALIVE message is sent with Client-Type set to zero.

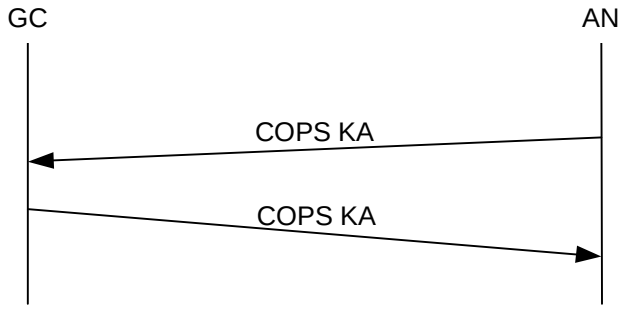


Figure 15: COPS Keep-alive exchange

7.4.2 Operation Sequence

The protocol between the Gate Controller and AN is for purposes of resource control and resource allocation policy. The Gate Controller implements all the allocation policies, and uses that information to manage the set of gates implemented in the AN. The Gate Controller initializes the gates with specific source, destination, and bandwidth restrictions; and once initialized, the MTA is able to request resource allocations within the limits imposed by the Gate Controller.

Messages initiated by the Gate Controller include GATE-ALLOC, GATE-SET, GATE-INFO, and GATE-DELETE. The procedures for these messages are described in the following clauses. All are sent using client specific objects within the decision object of COPS DECISION messages. The responses from the AN are sent as a REPORT-STATE message with client specific objects in the ClientSI object.

The DECISION messages and REPORT-STATE messages MUST contain the same handle as was used in the initial REQUEST sent by the AN when the COPS connection was initiated.

GATE-ALLOC validates the number of simultaneous sessions allowed to be set-up from the originating MTA, and allocates a Gate-ID to be used for all future messages regarding this gate or set of gates.

GATE-SET initializes and modifies all the policy and traffic parameters for the gate or set of gates, and sets the billing and gate co-ordination information.

GATE-INFO is a mechanism by which the Gate Controller can find out all the current state and parameter settings of an existing gate or set of gates.

The AN MUST periodically send a Keep Alive (KA) message to the GC to facilitate the detection of TCP connection failures. The Gate Controller keeps track of when KAs are received. If the Gate Controller has not received a KA from the AN in the time specified by RFC 2748 or the Gate Controller has not received an error indication from the TCP connection, then the Gate Controller MUST tear down the TCP connection and attempt to re-establish the TCP connection before the next time it is requested to allocate a gate from that AN.

GATE-DELETE allows a Gate Controller to delete a recently allocated gate under certain (see below) circumstances.

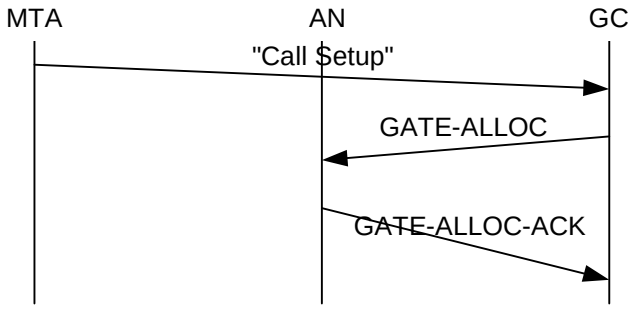
7.4.3 Procedures for Allocating a new Gate

A GATE-ALLOC message is sent by the Gate Controller to the AN at the time the "Call_Set-up" message is sent from the originating MTA (e.g. "Invite(stage1)" message when using DCS), as shown in figure 16.

The use of GATE-ALLOC ensures that not too many sessions are being simultaneously requested from a given MTA. This mechanism may be used to control a denial of service attack from the MTA. The AN, in its response to the GATE-ALLOC message, compares the number of currently allocated gates for the indicated subscriber-ID against the Count field of the Activity-Count object in the GATE-ALLOC message. If the current number of gates is greater than or equal to the Count field in the GATE-ALLOC, then the AN MUST return a GATE-ALLOC-ERR message. If the current number of gates is greater than the Count field in the GATE-ALLOC, then it is likely that the subscriber has been re-provisioned to have a lower gate limit than before. In this case, the subscriber's current sessions are not affected but any new sessions by that subscriber will be rejected by the AN until the subscriber's session count goes below the value specified in the Count field.

If the Activity-Count object is not present, the AN does not perform the gate limit check. A GC seeking to reduce call set-up time MAY decide to perform the gate limit check upon receipt of the GATE-ALLOC-ACK instead of having the AN perform the check so that the GC can do the GATE-ALLOC and subscriber policy lookup operations in parallel. When the results of both operations are available, the GC can do the gate limit check. If the check fails, the GC MUST send a GATE-DELETE message to the AN to delete the gate that was incorrectly allocated (see clause 7.4.6). The GC MAY include the Activity-Count object in subsequent GATE-ALLOCs for that subscriber once the policy has been cached.

The following diagram is an example of the GATE-ALLOC signalling:



NOTE: As an example, the "Call Setup" message in this context refers to the "Invite w/o ring" when using DCS.

Figure 16: Sample Signalling of GATE-ALLOC

The AN MUST respond to a GATE-ALLOC message with either a GATE-ALLOC-ACK (indicating success) or a GATE-ALLOC-ERR (indicating failure). The Transaction-ID in the response MUST match the transaction ID in the request.

Errors in allocating gates are reported by a GATE-ALLOC-ERR response. The IPCablecomError object contains one of the following Error-Codes:

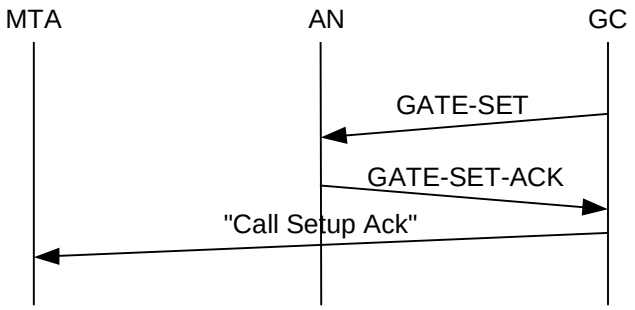
1 = No gates currently available.

4 = Subscriber exceeded gate limit.

127 = Other, unspecified error.

7.4.4 Procedures for Authorizing Resources Through a Gate

The GATE-SET message is sent by the Gate Controller to the AN to initialize or modify the operational parameters of the gate(s). Figure 17 is an example of the GATE-SET signalling.



NOTE: As an example, the "Call Setup Ack" message in this context refers to the "200 OK" message returned from the "Invite w/o ring" when using DCS.

Figure 17: Sample Signalling of GATE-SET

If a Gate-ID Object is present in the GATE-SET message, then the request is to modify an existing gate. If the Gate-ID Object is missing from the GATE-SET message, then it is a request to allocate a new gate, and the Activity-Count Object MAY be present so that the AN can determine if the subscriber has exceeded the maximum number of simultaneous gates.

The GATE-SET message MUST contain exactly one or two Gate-Spec objects, describing zero or one upstream gates, and zero or one downstream gates.

The AN MUST respond to a GATE-SET message with either a GATE-SET-ACK (indicating success) or a GATE-SET-ERR (indicating failure). The Transaction-ID in the response MUST match the transaction ID in the request.

Errors in allocating or authorizing gates are reported by a GATE-SET-ERR response. The IPCablecom-Error object contains one of the following Error-Codes:

1 = No gates currently available.

2 = Illegal Gate-ID.

3 = Illegal Session Class value.

4 = Subscriber exceeded gate limit.

127 = Other, unspecified error.

In handling a reservation request from an MTA, the AN MUST determine the proper gate by use of the RSVP Gate-ID object, or by the use of the Authorization Block TLV. The AN MUST verify the reservation request is within the authorized limits specified for the gate.

The AN then updates the reservation request based on gate parameters. If the auto-commit flag is set, then the AN MUST take appropriate action on the J.112 MAC-layer to commit the resources immediately. The AN MUST set the IP-Type-Of-Service-Overwrite (TOS) by the Diffserv Code Point (DSCP) parameter.

The AN MUST perform an admission control function, based on provisioned policy parameters and the Session Class value of the gate.

Note that a GATE-SET message can be used to allocate (and set) a gate instead of the GATE-ALLOC message. In such situations, it is possible that the port number being used by the remote gate for receiving gate co-ordination messages is not available to the gate controller. If that is the case, the AN-port in the Remote-Gate-Info object (carried in the GATE-SET message) is set to zero. This causes the AN to ignore the gate co-ordination port number. However, when the gate controller (later) learns about the port number being used by the remote gate, it must send another GATE-SET message (with the port number in the Remote-Gate-Info object) to inform the AN about this port.

7.4.5 Procedures for Querying a Gate

When a Gate Controller wishes to find out the current parameter settings of a gate, it sends to the AN a GATE-INFO message. The AN MUST respond to a GATE-INFO message with either a GATE-INFO-ACK (indicating success) or a GATE-INFO-ERR (indicating failure). The Transaction-ID in the response MUST match the transaction ID in the request.

Errors in querying gates are reported by a GATE-INFO-ERR response. The Error object contains one of the following Error-Codes:

2 = Illegal Gate-ID.

127 = Other, unspecified error.

7.4.6 Procedures for Deleting a Gate

In a normal call flow, a gate is deleted by the AN when it receives an RSVP-PATH-TEAR message or the request to release the J.112 Flow via the J.112 MAC-layer Interface (from an embedded MTA that does not support RSVP). The AN also deletes a gate at the receipt of a GATE-CLOSE message from a remote AN (DCS model) or a CMS (NCS model).

A gate controller, typically, does not initiate a gate delete operation. However, there could be certain abnormal situations where a gate controller might want to delete a gate on the AN. For instance, if the gate controller learns (at the receipt of a GATE-ALLOC-ACK response) that a subscriber has exceeded its gate limit, it might want to delete the recently allocated gate at the AN. In such scenarios, it MAY send a GATE-DELETE message to the AN (instead of allowing the gate to time out). There could be other situations in which the delete functionality might be useful.

The AN MUST respond to a GATE-DELETE message with a GATE-DELETE-ACK (indicating success) or a GATE-DELETE-ERR (indicating failure). The Transaction-ID in the response MUST match the Transaction-ID in the request. Errors in deleting gates are reported by a GATE-DELETE-ERR response. The Error object contains one of the following Error-Codes:

2 = Illegal Gate-ID.

127 = Other, unspecified error.

7.4.7 Termination Sequence

When the AN is shutting down its TCP connection to the GC, it MAY first send a DELETE-REQUEST-STATE message (including the handle object used in the REQUEST message). The AN MAY follow that with a CLIENT-CLOSE message. These messages are optional because the GC is stateless and because the COPS protocol requires a COPS server to automatically delete any state associated with the AN when the TCP connection is terminated.

When the Gate Controller is going to shutdown, it SHOULD send a COPS Client-Close (CC) message to the AN. In the COPS CC message, the Gate Controller SHOULD NOT send the PDP redirect address object <PDPRedirAddr>. If the AN receives a COPS CC message from the Gate Controller with a <PDPRedirAddr> object, the AN MUST ignore the <PDPRedirAddr> while processing the COPS CC message.

8 Gate-to-Gate Co-ordination Interface (PKT-Q8)

Messages are exchanged between the gates to synchronize their use. These are messages that include GATE-OPEN, GATE-CLOSE and their corresponding Acknowledgments. GATE-OPEN messages are exchanged when the gate has committed resources activated or changed as the result of a command from the MTA (see figure 18). GATE-CLOSE messages are exchanged when those resources are released. Timers within the gate implementation impose strict controls on the length of time these exchanges may occupy.

Gate synchronization messages may be exchanged directly between the ANs, or may be exchanged through proxies (typically the IPCablecom Call Management System (CMS), who desires notification of various error cases that cause gates to be prematurely closed). Figure 18 shows the direct gate-gate co-ordination, and figure 19 shows the gate co-ordination through CMS-proxies at both ends. Also possible, but not shown, are configurations with a proxy at only one end.

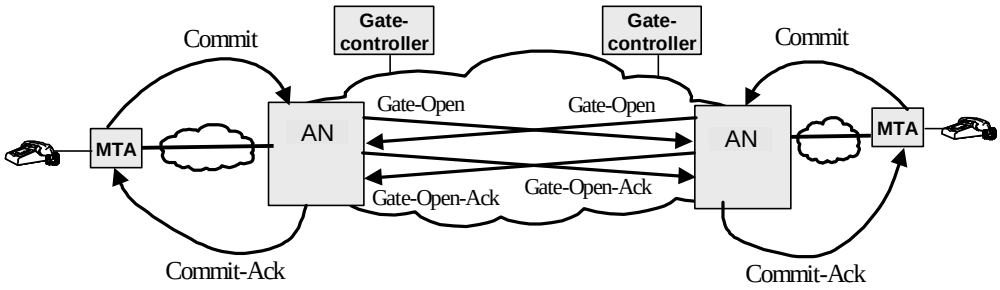


Figure 18: End-to-end Gate Co-ordination

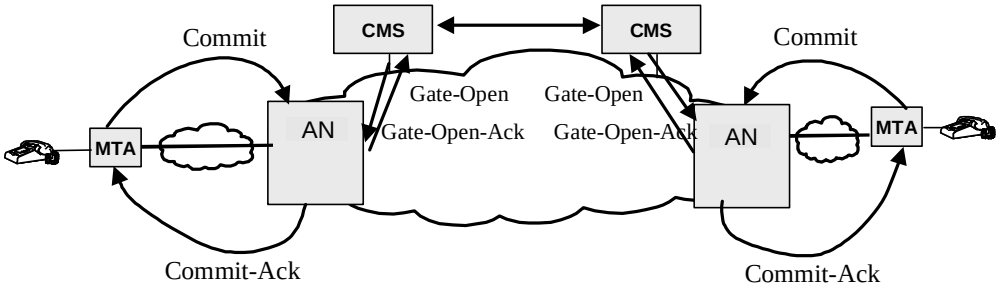


Figure 19: CMS-Proxy Gate Co-ordination

A gate is initially created by a GATE-SET command from the Gate Controller. The GATE-SET command will contain such information as the prototype classifiers (i.e. 6-tuple) and Flowspecs for both the local and remote gates. It also contains the IP address and port number of the remote AN so they can implement Gate-to-Gate co-ordination.

8.1 Gate-to-Gate Protocol Messages

Gate-to-Gate Protocol messages are sent as UDP/IP packets, where the UDP Destination Port is given by the GATE-SET command. The UDP Source Port MUST be the port at which the sender is listening for the acknowledgement. Exactly one message MUST be encapsulated in the UDP Data field. The format of the common header to all messages is shown below, and is identical to and copied from the specification of RADIUS.

Message Type	Transaction ID	Message Length
Message Authenticator (16 bytes)		
Parameters ….		

The Message Type is one octet, and identifies the type of packet. Type codes are assigned as follows:

GATE-OPEN	48
GATE-OPEN-ACK	49
GATE-OPEN-ERR	50
GATE-CLOSE	51
GATE-CLOSE-ACK	52
GATE-CLOSE-ERR	53

Transaction ID is one octet, and aids in matching requests and responses.

Message Length is two octets, and indicates the length of the message, including the header and all parameters.

The Message Authenticator is a 16-byte MD5 checksum. This value is used to authenticate the request and the response, and is based on a shared secret between the two ANs. The Message Authenticator in GATE-OPEN and GATE-CLOSE messages contains a one-way MD5 RFC 1321 hash calculated over a stream of octets consisting of the Message-Type + Transaction-ID + Message-Length + 16 zero octets + Parameters + shared secret. The Message Authenticator in GATE-OPEN-ACK, GATE-OPEN-ERR, and GATE-CLOSE-ACK messages contains a one-way MD5 hash calculated over a stream of octets consisting of the Message-Type, Transaction-ID + Message-Length + Message Authenticator from request message + Response parameters (if any) + shared secret. The resulting 16-byte MD5 hash value is stored in the Message Authenticator field of the packet. This algorithm for calculation of the Message Authenticator is identical to that described in RFC 2138.

Parameters are all encoded in the Type-Length-Value style of RADIUS RFC 2138. Parameters carry the specific request and indication information needed to achieve gate co-ordination. The parameter format MUST be as shown below:

Type	Length	Reserved, MUST be zero
Value ….		

The Type field is one octet, and contains the following values:

Gate-ID 224

Tspec 225

Reverse-Tspec 226

Error-code 227

The Length field is one octet and contains the length in bytes of the parameter. All length values in the present document are multiples of 4.

The Gate-ID parameter, when present in a message, has the following format:

224	8	0
Gate-ID value (32-bit integer)		

The Tspec parameter, when present in a message, has the following format (see clause 6.3.1 for explanation of fields):

225	36	0
0 (a) Reserved		7 (b)
1 (c)	0 Reserved	6 (d)
127 (e)	0 (f)	5 (g)
Token Bucket Rate [r] (32-bit IEEE floating point number)		
Token Bucket Size [b] (32-bit IEEE floating point number)		
Peak Data Rate [p] (32-bit IEEE floating point number)		
Minimum Policed Unit [m] (32-bit integer)		
Maximum Packet Size [M] (32-bit integer)		

The Reverse-Tspec parameter, when present in a message, has the following format (see clause 6.3.5 for explanation of fields):

226	36	0
0 (a) Reserved		7 (b)
1 (c)	0 Reserved	6 (d)
127 (e)	0 (f)	5 (g)
Token Bucket Rate [r] (32-bit IEEE floating point number)		
Token Bucket Size [b] (32-bit IEEE floating point number)		
Peak Data Rate [p] (32-bit IEEE floating point number)		
Minimum Policed Unit [m] (32-bit integer)		
Maximum Packet Size [M] (32-bit integer)		

The Error-code parameter, when present in a message, has the following format:

227	4	Error Code	Reserved

The Error Code values are as follows:

0 Normal release, initiated by MTA.

1 Close initiated by AN due to lack of Reservation Maintenance (e.g. RSVP refreshes).

2 Close initiated by AN due to lack of J.112 MAC-layer responses (e.g. station maintenance).

3 Timer T1 expired; No COMMIT received from MTA.

4 Timer T2 expired; Gate Co-ordination failure.

5 Close initiated by AN due to reservation reassignment (e.g. for priority session).

6 Close initiated by AN due to reservation mismatch.

129 Illegal Gate-ID.

130 Message Authenticator incorrect.

255 Other, unspecified error.

8.1.1 GATE-OPEN

The format of a GATE-OPEN message MUST be as follows:

<GATE-OPEN> ::= <RADIUS-Common-Header> <Gate-ID>
[<Tspec> <Reverse-Tspec>]

The value of Gate-ID is copied from the Remote-Gate-ID value contained in the Remote-Gate-Info object of the Gate-Set message.

When a GATE-OPEN message is generated an AN, the Tspec and Reverse-Tspec objects MUST be present.

The values in the Tspec parameter are copied from the Flowspec object of the COMMIT message, if it exists, and if not, from the Sender-Tspec object of the RSVP-PATH message that initiated the reservation, or generated from the J.112 MAC-layer messages that initiated the commit operation. In all cases, it indicates the resources committed in the upstream (forward) direction.

The values in the Reverse-Tspec parameter are copied from the Reverse-Sender-Tspec object of the COMMIT message, if it exists, and if not, from the Reverse-Sender-Tspec object of the RSVP-PATH message that initiated the reservation, or generated from the J.112 MAC-layer messages that initiated the commit operation. In all cases, it indicates the resources committed in the downstream (reverse) direction.

8.1.2 GATE-OPEN-ACK

The format of a GATE-OPEN-ACK message MUST be as follows:

<GATE-OPEN-ACK> ::= <RADIUS-Common-Header>

There are no parameters in this acknowledgement message. The Transaction-ID in the common header serves to identify to the recipient which GATE-OPEN message is being acknowledged.

8.1.3 GATE-OPEN-ERR

The format of a GATE-OPEN-ERR message MUST be as follows:

<GATE-OPEN-ERR> ::= <RADIUS-Common-Header> <Error-code>

The Transaction-ID in the common header serves to identify to the recipient which GATE-OPEN message is being acknowledged.

The Error-code parameter contains a reason code indicating the cause of the error.

If the error is such that the Gate-ID is not recognized, and therefore the proper authentication key is not known, or if the Message Authenticator of the GATE-OPEN message is incorrect, the Message Authenticator of the GATE-OPEN-ERR message MUST be an exact copy of the Message Authenticator of the GATE-OPEN message.

8.1.4 GATE-CLOSE

The format of a GATE-CLOSE message MUST be as follows:

<GATE-CLOSE> ::= <RADIUS-Common-Header> <Gate-ID> [<Error-Code>]

If the GATE-CLOSE message is being generated due to other than a normal release request from the MTA, then the Error-Code MUST be present giving the reason.

GATE-CLOSE MUST NOT be used in cases where no gate is open. In cases where no gate is open or the CMS (when not serving as a proxy for the remote AN) requires to close a gate, the GATE-DELETE message is used.

8.1.5 GATE-CLOSE-ACK

The format of a GATE-CLOSE-ACK message MUST be as follows:

<GATE-CLOSE-ACK> ::= <RADIUS-Common-Header>

The Transaction-ID in the common header serves to identify to the recipient which GATE-CLOSE message is being acknowledged.

8.1.6 GATE-CLOSE-ERR

The format of a GATE-CLOSE-ERR message MUST be as follows:

<GATE-CLOSE-ERR> ::= <RADIUS-Common-Header> <Error-String>

The Transaction-ID in the common header serves to identify to the recipient which GATE-CLOSE message is being acknowledged. The Message Authenticator is an exact copy of the Message Authenticator of the GATE-CLOSE message.

8.2 Gate Co-ordination Procedures

When the MTA performs a Commit operation (as described in clause 6.7 for any MTA, or in annex A or annex B for embedded MTAs), the AN MUST send a GATE-OPEN message. The GATE-OPEN message MUST contain both Flowspecs (i.e. bidirectional flows). The AN MUST retransmit the GATE-OPEN message, based on Timer T5, until receipt of a GATE-OPEN-ACK response. After a fixed number of retransmission attempts, the AN declares unacceptable packet loss and closes the gate.

On receipt of a GATE-OPEN message, the AN MUST acknowledge it with a GATE-OPEN-ACK message.

If the AN receives a GATE-OPEN message, but has no record of the Gate-ID, and therefore does not know the proper security key, it MUST send the GATE-OPEN-ERR with a Message Authenticator matching the Message Authenticator of the GATE-OPEN message.

The AN MUST ignore an incorrect Message Authenticator when the Message Type is GATE-OPEN-ERR, the Transaction-ID matches an outstanding GATE-OPEN message sent, and the Message Authenticator matches the Message Authenticator of the GATE-OPEN message.

On a Commit request or on receipt of the GATE-OPEN message, whichever occurs first, the AN MUST start Timer T2.

On a Commit request or on receipt of the GATE-OPEN message, whichever occurs second, the AN MUST cancel Timer T2. If the flowspecs do not match, the AN MUST close the gate, initiate the release of the J.112 Flow, and send a GATE-CLOSE message.

If Timer T2 expires after receipt of a Commit request, but without receipt of a GATE-OPEN message, the AN MUST close the gate, initiate the release of the J.112 Flow, and send a GATE-CLOSE message.

The AN MUST send a GATE-CLOSE message when it receives an explicit release message from the MTA client (as described in clause 6.5.3 for any MTA, or in annex A or annex B for embedded MTAs), or when it detects that the client is no longer actively generating packets and not generating proper refreshes for the flow associated with a gate. The AN MUST also close a gate when it receives a GATE-CLOSE message. This ensures that the gates associated with a session are closed almost simultaneously.

On receipt of a properly authenticated GATE-CLOSE message, the AN MUST always respond with a GATE-CLOSE-ACK, sent to the address given as the source address of the command. After sending the GATE-CLOSE-ACK, the AN MUST keep the Gate-ID and authentication key available for a period of at least 30 s to allow for possible retransmissions of the GATE-CLOSE message.

If the AN has no record of the Gate-ID, and therefore does not know the proper security key, it MUST send the GATE-CLOSE-ERR with a Message Authenticator matching the Message Authenticator of the GATE-CLOSE message.

The AN MUST ignore an incorrect Message Authenticator when the Message Type is GATE-CLOSE-ERR, the Transaction-ID matches an outstanding GATE-CLOSE message sent, and the Message Authenticator matches the Message Authenticator of the GATE-CLOSE message.

8.2.1 Example Procedures for end-to-end Gate Co-ordination

To perform end-to-end gate co-ordination, the Gate Controller establishes each gate with the address and Gate-ID of the other remote AN; each AN sends and receives the GATE-OPEN/GATE-CLOSE messages from the other.

Once the MTAs have completed their session signalling, they will start the session by performing a Commit operation (as described in clause 6.7 for any MTA, or annex A or annex B for embedded MTAs) to the AN. This causes the AN to open the gate. The AN now informs the remote AN that the gate is opened. The local AN sends a GATE-OPEN message to the remote AN and starts Timer T2, described in annex C. The GATE-OPEN message contains both Flowspecs (i.e. bidirectional flows). The remote AN acknowledges the GATE-OPEN message with a GATE-OPEN-ACK message.

In addition, the AN expects to receive a GATE-OPEN message from the remote AN after the remote MTA sends its COMMIT message. This remote GATE-OPEN message from the remote AN similarly contains both Flowspecs. These flowspec parameters are compared to those of the local AN. If the Flowspecs match, the Gate is allowed to remain open.

To disable the Timer T2, both a GATE-OPEN-ACK and a GATE-OPEN message are received from the remote AN. If the GATE-OPEN-ACK is not received from the remote AN within the expiration of Timer T5 (described in annex C, value is of the order of a round trip delay), the AN retransmits the local GATE-OPEN message to recover from the loss. This method of application-level recovery of the message is attempted up to a fixed number of retransmission attempts, after which the AN declares unacceptable packet loss and closes the Gate. The value of the Timer T2 should be large enough to allow for recovery of lost messages.

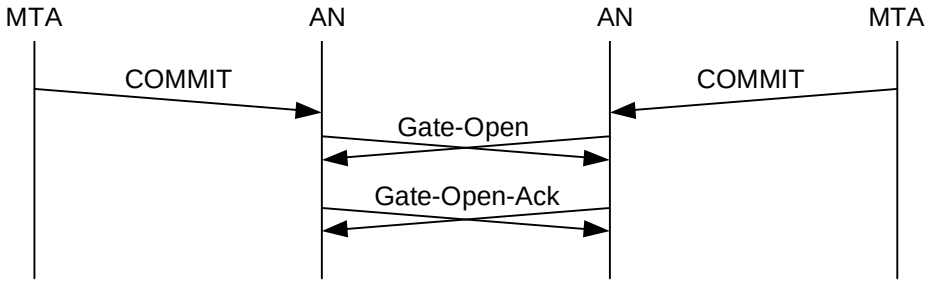


Figure 20: Gate Co-ordination at time of COMMIT

Gate co-ordination is also done at the time a gate is closed. Each AN sends a GATE-CLOSE message to its peer AN when it receives an explicit release message from the MTA (as described in clause 6.5.3 for any MTA, or in annex A or annex B for embedded MTAs), or when it detects that the client is no longer actively generating packets and not generating proper refreshes for the flow associated with a gate. An AN also closes a gate when it receives a GATE-CLOSE message from the remote AN. This ensures that the gates associated with a session are closed almost simultaneously.

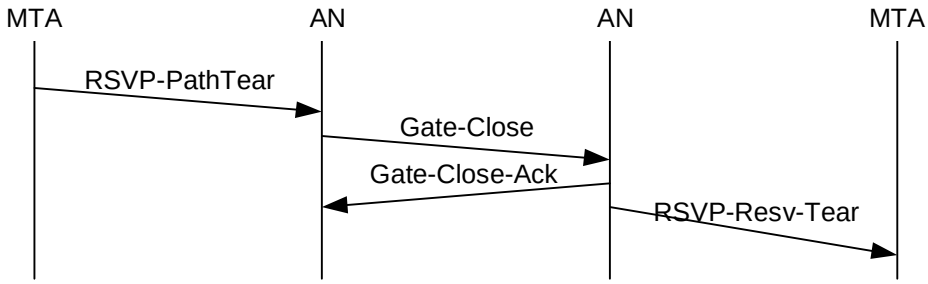


Figure 21: Gate Co-ordination on Release

On receipt of a properly authenticated GATE-CLOSE message, the AN responds with a GATE-CLOSE-ACK, sent to the address given as the source address of the command. After sending the GATE-CLOSE-ACK, the AN keeps the Gate-ID and authentication key available for a period of at least 30 s to allow for possible retransmissions of the GATE-CLOSE message.

8.2.2 Example Procedures for Proxied Gate Co-ordination

This example shows how a Call Management System (CMS) can use proxied gate co-ordination. The Gate Controller initializes each gate with the address of the CMS as the remote co-ordination entity, and a CMS-chosen identifier as a Gate-ID. The AN performs the gate co-ordination procedures by sending the GATE-OPEN/GATE-CLOSE messages to the CMS, who passes them on to the remote gate.

When the CMS determines that resources are available at the terminating (remote) end it will instruct the MTA to commit resources. It will also send a GATE-OPEN message to the AN and start Timer T5. The AN acknowledges the GATE-OPEN message with a GATE_OPEN_ACK message, which disables Timer T5 in the CMS. If the GATE_OPEN_ACK is not received from the AN within the expiration of Timer T5, the CMS retransmits the GATE-OPEN message to recover from the loss. This method of application-level recovery of the message is attempted up to a fixed number of retransmission attempts, after which the CMS declares unacceptable packet loss and closes the Gate. Upon the reception of GATE-OPEN from the CMS or COMMIT message from the MTA, the AN starts Timer T2.

To disable the Timer T2, the AN must successfully receive a COMMIT message from the MTA and GATE-OPEN message from the CMS. If Timer T2 expires, the AN initiates a GATE-CLOSE message or J.112 MAC layer message (as appropriate) in order to close the gate and releases all resources associated with the gate.

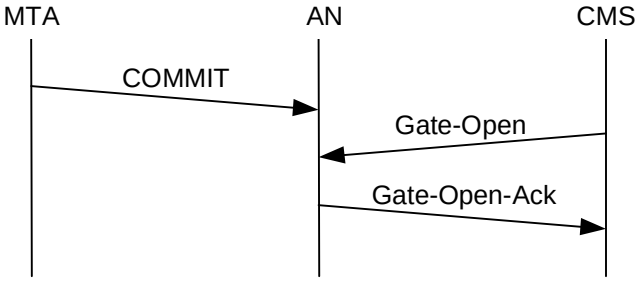


Figure 22: Gate Co-ordination at time of COMMIT

Gate co-ordination is also done at the time a gate is closed. The AN sends a GATE-CLOSE message to its CMS when it receives an explicit release message from the MTA (as described in clause 6.5.3 for any MTA, or in annex A or annex B for embedded MTAs), or when it detects that the client is no longer actively generating packets and not generating proper refreshes for the flow associated with a gate. an AN also closes a gate when it receives a GATE-CLOSE or GATE-DELETE message from the CMS. This ensures that the gates associated with non-responsive MTAs are closed.

On receipt of a properly authenticated GATE-CLOSE message, the CMS responds with a GATE-CLOSE-ACK, sent to the address given as the source address of the command. After sending the GATE-CLOSE-ACK, the CMS keeps the Gate-ID and authentication key available for a period of at least 30 s to allow for possible retransmissions of the GATE-CLOSE message.

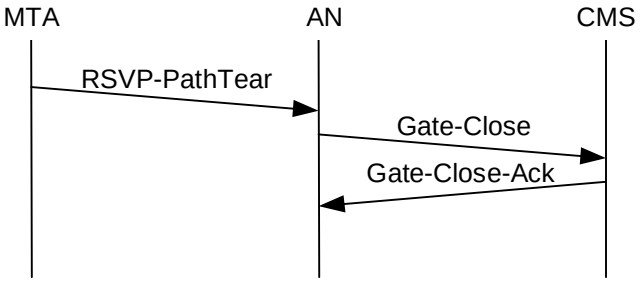


Figure 23: Gate Co-ordination on Release

Annex A (normative): Additional requirements for J.112 annex A implementations

Instead of using the PKT-Q3 (RSVP+) interface to request QoS in the J.112 network as described in clause 6, an embedded MTA MAY dynamically reserve local QoS resources using mechanisms defined in ITU-T Recommendation J.112. With this alternative approach, an embedded MTA directly signals its need for QoS in the local J.112 access network using the MAC Primitives defined in annex A of ITU-T Recommendation J.112. As opposed to clause 6, the QoS signalling using the J.112 MAC protocol (PKT-Q2) is initiated by the CM instead of the AN upon request of the MTA. In the mechanism described in clause 6 the request for QoS is received by the AN via a layer 4 interface (PKT-Q3) while the mechanism described in this annex uses a MAC-layer Interface (MAC Primitives) interfacing the MTA with the CM (PKT-Q1). All other interfaces and signals remain unchanged. Illustrative examples of this approach are given in annex J and VIII.

An embedded MTA receives at its application layer interface session based QoS requirements in signalling protocols (RFC 2543 and TS 101 909-4). Once the embedded MTA determines that QoS resources need to be reserved or committed, the MTA MUST initiate J.112 signalling to cause the translation of the session based QoS requirements of the application to a resource allocation based on J.112 Flows in the J.112 network and the resulting creation, change, and/or deletion of appropriate flows. Whether the session is originated by the embedded MTA or by a peer or by a network node of the CPE, the MTA passes the QoS requirements to the J.112 MAC protocol via the MAC Primitives. This triggers appropriate actions on the MAC layer to create or modify J.112 using Connection Establishment and/or Link Management messaging mechanisms of the J.112 MAC Protocol.

The following clauses discuss the MTA's mapping of the session based QoS requirements of the application into the resources required in the J.112 network, the use of the MAC Primitives and the support of two phase reserve/commit resource allocation in the J.112 network.

A.1 Terminology

In a network complying to J.112 annex A the customer side terminal can be formed either by a CM or a Set-top Box (STB). Both devices incorporate a Network Interface Unit (NIU) that provides the physical and logical interface between the J.112 network and the CPE. The Access Node (AN) is, in that case, implemented as an Interactive Network Adapter (INA) providing the interface to the backbone network and to elements of the IPCablecom architecture that are established outside of the J.112 network, like CMS and RKS. J.112 Flows are considered to be bidirectional connections.

Since this annex is only referring to J.112 networks complying to J.112 annex A the terms Access Node (AN) and Interactive Network Adapter (INA) are used interchangeably.

A.2 Mapping of Flowspecs into J.112 QoS parameters

An embedded MTA receives QoS requirements of an application on a per session basis and has to pass it to the J.112 MAC protocol using MAC Primitives. The QoS requirements are received in the format of higher-layer service descriptions (e.g. SDP as used in VoIP applications) if the session is initiated by the MTA itself or in the format of RSVP flowspecs if the session is initiated by a peer or network node. Other specifications (e.g. the IPCablecom CODEC specification J.acr) define the mapping of higher-layer service descriptions into flowspecs. This clause specifies how the MTA MUST map QoS requirements to J.112 MAC layer parameters. In this clause it is assumed that the transport protocol being used is UDP. If a different transport protocol is used, appropriate changes would be applicable to the MAC parameters specified herein and for header suppression.

In the J.112 network, resources are allocated on a connection basis. A connection is a single bidirectional data flow between the CM and the INA. As such, the connection comprises a downstream flow and an upstream flow. Resources are reserved for up- and downstream directions. They are described with a set of parameters, which in general might differ for up- and downstream directions. The J.112 MAC protocol defines several QoS parameters that are applicable for different J.112 annex A access modes. Thus, the MTA specifies in its request what QoS parameters to associate with the corresponding connection in up- and downstream direction.

To request a specific access mode the MTA MAY use policy information given by the J.112 network operator and the characteristics of the source as described in the QoS requirements for the session. However, the final decision on what resources are allocated to a particular connection is entrusted to the INA and MUST also be based on the total amount of available resources.

To give an example on the mapping of a session description to the J.112 QoS parameters consider a VoIP application that uses the audio codec G.729 annex E and the following SDP description:

- c = IN IP4 192.168.73.10
- m = audio 3456 RTP/AVP 96
- a = rtpmap: 96 G729E/8000
- a = ptime: 10

where "c" contains the connection information, "m" is the description of the media to be carried in this session and "a" describes attributes of the session. In this particular session, a "rtpmap" is included specifying the codec parameters. The attribute "ptime" defines that one packet represents 10 ms of audio. The session description can be mapped to J.112 MAC parameters in upstream direction as:

- Fixed-rate access.
- Requested bandwidth of 240 ATM cells per 1 200 ms (equivalent to 75 kbit/s).
- Cyclic assignment of two slots every 60 slots.

In the above example it is assumed that DirectIP is used as the upstream encapsulation method and that the upstream data rate is 3,088 Mbit/s. In calculating the requested bandwidth, the overhead of the encapsulation method and any J.112 MAC protocol overhead MUST be taken into account. By using header suppression the PDU size in upstream direction can potentially be reduced significantly, depending on what fields of the headers can be suppressed.

A classifier is used to assign packets arriving at either the CM or the INA to the appropriate connection to ensure that they receive the QoS they deserve. In order to be able to set up a classifier in both termination points of the J.112 network the embedded MTA may include Session Binding parameters in its request. However, the INA may also receive these parameters from the Gate via the J.112 MAC-layer interface. Upstream Session Binding parameters are:

- Source Address: the IP address of the MTA.
- Source Port: the port number on which the MTA will send the media stream.
- Destination Address: the IP address of the far end of the connection as given in the "c" parameter of the SDP description.
- Destination Port: the port number on which the far end will receive the media stream as given in the "m" parameter of the SDP description.
- Protocol: the transport protocol to be used (UDP in the example above).

Downstream Session Binding parameters include:

- Source Address: the IP address of the far end of the connection as given in the "c" parameter of the SDP description.
- Source Port: the port number on which the far end will send the media stream, this parameter is not available to the MTA and SHOULD NOT be specified as part of the classifier.
- Destination Address: the IP address of the MTA.
- Destination Port: the port number on which the MTA will receive the media stream.
- Protocol: the transport protocol to be used (UDP in the example above).

A.3 Use of J.112 MAC Primitives

Once the embedded MTA has determined that QoS resources need to be reserved or committed, it initiates the appropriate J.112 signalling using MAC Primitives. MAC Primitives are defined in annex A of ITU-T Recommendation J.112. This clause describes the usage of MAC Primitives.

The MAC_RESOURCE_REQ MAC Primitive MUST be used by the embedded MTA to signal a request to create, change, and/or delete a connection. The type of the resource that is requested (including the request to release reserved resources) is indicated by the Resource_Type parameter.

A.3.1 Reserving Resources

The MTA initiates the reservation of QoS resources by use of the MAC_RESOURCE_REQ Primitive with the Resource_Type parameter set to 1, 2 or 4. The MTA must include the Gate ID as the Connection ID. For a more detailed description of the parameters of the MAC_RESOURCE_REQ Primitive refer to J.112 annex A. If the CM receives this message it invokes MAC signalling leading to the establishment of a new connection. It confirms the reception of the Primitive by answering with a MAC_RESOURCE_CNF Primitive. Authorization of the MTA to request the resources and availability of the resources is checked by the INA. If the INA detects a Connection ID that is already in use as a Gate ID with a corresponding connection not existing it is an indication that resources are reserved but not yet committed. The final decision is made according to the Admit_Bit in the <MAC> Resource Request message. If the Admit_Bit is set, the INA MUST NOT commit the resources yet. If it is cleared, the INA MUST commit the resources to the connection if admission control was successful. If the requested resources are not available the request is denied. The CM notifies the MTA of the result of the resource request with the MAC_CONNECT_IND or a MAC_RESOURCE_DENIED_IND Primitive. The process of reserving resources is illustrated in the following figure.

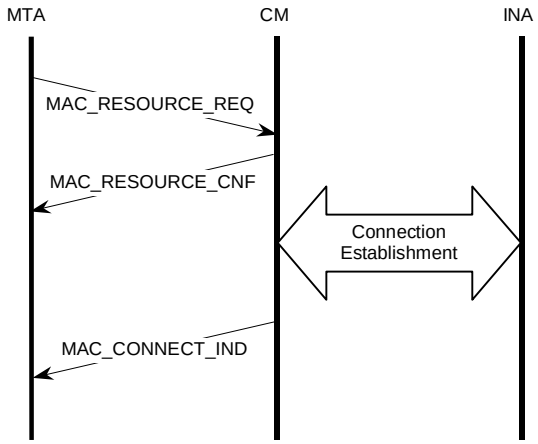


Figure 24: Resource Reservation using MAC Primitives

A.3.2 Committing Resources

The MTA initiates the commitment of QoS resources by use of the MAC_RESOURCE_REQ Primitive with the Resource_Type parameter set to 1 or 8. The MTA must include the Gate ID as the Connection ID. For a more detailed description of the parameters of the MAC_RESOURCE_REQ Primitive refer to J.112 annex A. The resources requested in this message MUST NOT be larger than the resources reserved with a previous request. If the CM receives this message it invokes MAC signalling leading to the reprovisioning of the existing connection. It confirms the reception of the Primitive by answering with a MAC_RESOURCE_CNF Primitive. If the INA detects a Connection ID that is already in use as a Gate ID with a corresponding connection existing and the Admit_Bit in the <MAC> Resource Request message received via the CM is cleared, resources are committed. The INA SHOULD NOT deny the requested if the resources are within the reserved envelope. The CM notifies the MTA of the result of the resource request with the MAC_CONNECT_IND or a MAC_RSV_ID_IND. The process of committing resources is illustrated in the following figure.

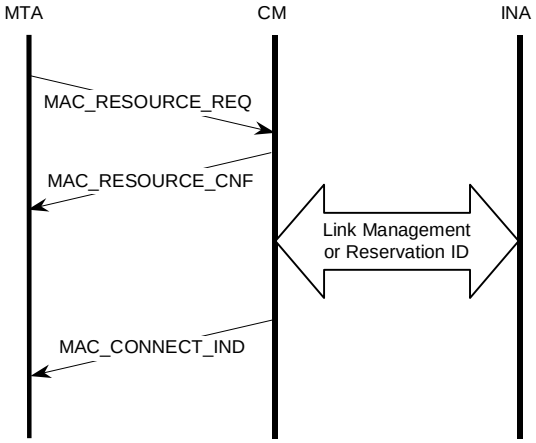


Figure 25: Resource Commitment using MAC Primitives

A.3.3 Releasing Resources

The MTA initiates the release of QoS resources by use of the MAC_RESOURCE_REQ Primitive with the Resource_Type parameter set to 16. The MTA must include the Gate ID as the Connection ID. For a more detailed description of the parameters of the MAC_RESOURCE_REQ Primitive refer to J.112 annex A. If the CM receives this message it invokes MAC signalling leading to the deletion of the connection and, thus, the release of the resources allocated to that connection. It confirms the reception of the Primitive by answering with a MAC_RESOURCE_CNF Primitive. The CM notifies the MTA of the result of the resource request with the MAC_RELEASE_IND. The process of releasing resources is illustrated in the following figure.

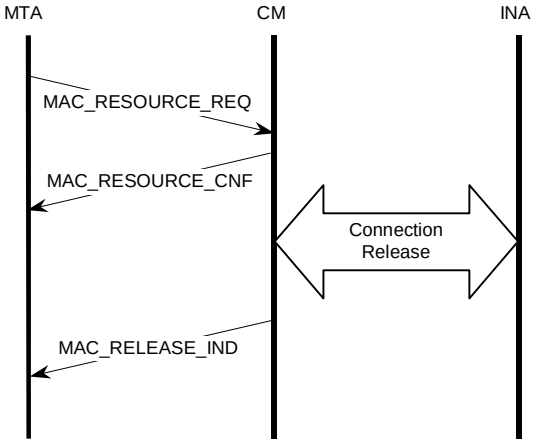


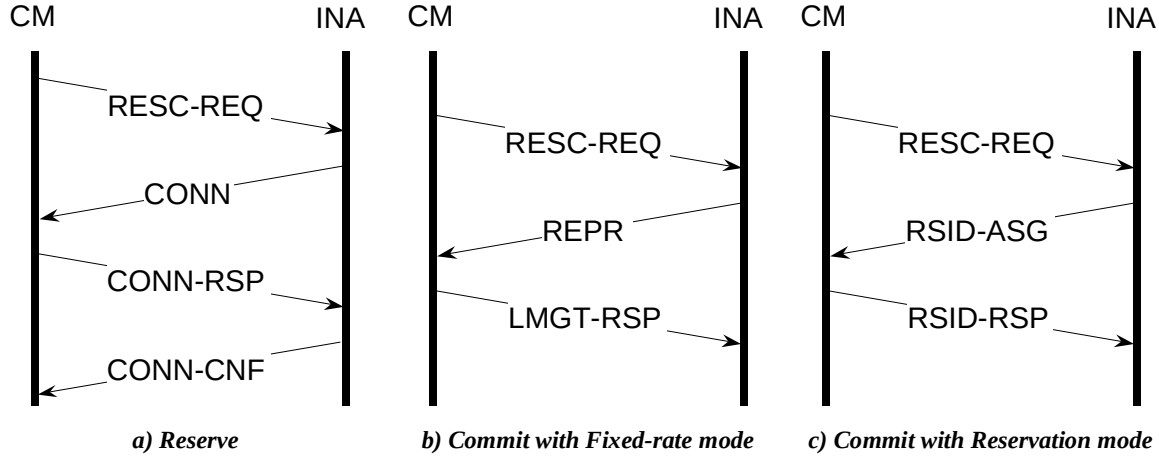
Figure 26: Resource Release using MAC Primitives

A.4 Support of Two-phase Resource Allocation

For a commercially deployable voice communications service it is essential to be able to distinguish between resources that are reserved for a session and resources that are committed to this session. The reason is, on the one hand side, to ensure that all resources are available before both communicating parties are notified that the may begin their conversation. On the other hand side, a two phase resource allocation ensures that recording and billing is not started until the media (i.e. the voice) is cut through. This clause describes the support of the J.112 network for this resource allocation mechanism.

A J.112 Flow has three associated sets of QoS parameters. The authorized parameter set is defined by the policy of the network operator and/or service provider and gives the maximum amount of resources a particular session may be awarded. Upon request resources are reserved. To commit these resources a second explicit request has to be submitted by both parties.

Both types of request, the Reserve and the Commit operation, are performed by the use of J.112 MAC messages initiated by the CM. The Reserve operation is carried out by setting up a new connection. Allocation and reservation of resources take place in the INA. The Commit operation makes use of the resource request mechanism for an existing connection established in the J.112 MAC protocol. The message exchanges comprising the Reserve and Commit operation are illustrated in the following figure.



a) Reserve *b) Commit with Fixed-rate mode* *c) Commit with Reservation mode*

Figure 27: Reserve and Commit operation using J.112 annex A MAC message exchanges

As an example, the following RESC-REQ message causes the INA to establish a connection and to reserve resources in up- and downstream of the J.112 network. Its reply is the CONN message shown below.

RESC-REQ (Resource Request Message)	
Resource Request ID	0x01
Connection ID	Gate ID
Field	
Aux_control_field_included	1
Admit_flag	1
Priority_included	0
Max_packet_size_included	1
Session_binding_US_included	0
Release_requested	0
Reservation_ID_requested	0
Cyclic_assignment_needed	1
Requested_bandwidth	240
Maximum_distance_between_slots	60
Encapsulation	DirectIP (1)
Aux_control_field	
IPv6_add	0
Flowspec_DS_included	1
Session_binding_DS_included	0

RESC-REQ (Resource Request Message)	
Frame_length	2
Flowspec_DS	
Max_packet_size	55
Average_bitrate	5 632
Jitter	0

CONN (Connect Message)	
Connection ID	Gate ID
Session_number	don't care
Connection_Control_Field_Aux	
Connection_control_field2_included	1
IPv6_add	0
Priority_included	0
Flowspec_DS_included	0
Session_binding_US_included	0
Session_binding_DS_included	0
Encapsulation_included	1
DS_multiprotocol_CBD_included	0
Resource_number	0x01
Connection_Control_Field	
DS_ATM_CBD_included	0
DS_MPEG_CBD_included	1
US_ATM_CBD_included	1
Upstream_Channel_Number	0x1
Slot_list_included	0
Cyclic_assignment	0
Frame_Length	0
Maximum_Contention_Access_Message_Length	1
Maximum_Reservation_Access_Message_Length	50
Downstream_MPEG_CBD	
Downstream_Frequency	472 000 000
Program_Number	0xA437

CONN (Connect Message)	
Upstream_ATM_CBD	
Upstream_Frequency	20 000 000
Upstream_VPI	0x01
Upstream_VCI	0x54AC
MAC_Flag_Set	0x01
Upstream_Rate	Upstream_3.088M
Encapsulation	DirectIP (1)
Connection_control_field2	
Upstream_modulation_included	1
Upstream_Modulation	QPSK (1)

Assuming that the media source exhibits a CBR like behaviour the MTA will most likely request a connection in Fixed-rate access mode with the INA previously having reserved appropriate resources. In this case, the following exchange of a RESC-REQ and REPR message will occur between the CM and the INA to commit the resources.

RESC-REQ (Resource Request Message)	
Resource Request ID	0x02
Connection ID	Gate ID
Field	
Aux_control_field_included	1
Admit_flag	0
Priority_included	0
Max_packet_size_included	1
Session_binding_US_included	0
Release_requested	0
Reservation_ID_requested	0
Cyclic_assignment_needed	1
Requested_bandwidth	240
Maximum_distance_between_slots	60
Encapsulation	DirectIP (1)
Aux_control_field	
IPv6_add	0
Flowspec_DS_included	1
Session_binding_DS_included	0

RESC-REQ (Resource Request Message)	
Frame_length	2
Flowspec_DS	
Max_packet_size	55
Average_bitrate	5 632
Jitter	0

REPR (Reprovision Message)	
Reprovision_Control_Field	
Reprovision_Control_Aux_Field_included	0
Delete_Reservation_Ids	0
New_Downstream_IB_Frequency_included	0
New_Downstream_OOB_Frequency_included	0
New_Upstream_Frequency_included	0
New_Frame_Length_inlcuded	1
New_Cyclical_Assignment_included	1
New_Slot_List_included	0
New_Frame_Length	2
Number_of_Connections	1
Connection_ID	Gate ID
Cyclic_Assignment	
Fixedrate_Start	0x0000
Fixedrate_Dist	60
Fixedrate_End	0xFFFF

A.5 Reservation Maintenance

For further study.

Annex B (normative): Additional requirements for J.112 annex B and annex C implementations

Rather than using the PKT-Q3 interface as described in clause 6, an embedded MTA MAY dynamically reserve local QoS resources using only mechanisms defined in J.112. Using this alternate approach, an embedded MTA directly signals for the local access QoS using the MAC Control Service interface defined in Appendix B.IV of J.112. As opposed to clause 6, the QoS signalling across the J.112 interface (PKT-Q2 interface) is initiated by the CM instead of the AN. All other interfaces remain unchanged. An illustrative example of this approach is given in annex J and annex K.

An embedded MTA signals its session level QoS requirements in signalling protocols (SIP RFC 2543 and TS 101 909-4). Once the embedded MTA determines that QoS resources need to be reserved or committed, the MTA MUST initiate J.112 Dynamic Service Flow signalling to cause the creation, change, and/or deletion of Service Flow(s) and the allocation of J.112 resources. Whether the session is originated by the embedded MTA or by a peer or network node, the MTA passes the QoS requirements to the J.112 MAC via the MAC Control Service Interface. This results in the creation or modification of the necessary Service Flow(s) for the session using the Dynamic Service Flow messaging mechanisms of J.112. The clauses that follow discuss the MTA's mapping of session level QoS requirements into those of J.112, the J.112 support for two phase reserve/commit, and the use of the J.112 MAC Control Service Interface.

B.1 Mapping Flowspecs into J.112 QoS Parameters

Other specifications (e.g. the IPCablecom CODEC specification TS 101 909-3) contain the mapping requirements of higher-layer service descriptions (e.g. SDP as used in VoIP applications) into Flowspecs. This clause specifies how the MTA MUST map Flowspecs to J.112 layer 2 parameters. The present document assumes that the transport protocol being used is UDP. If a different transport protocol is used, appropriate changes would be applicable in the classifiers and for payload header suppression.

J.112 defines a rich set of QoS parameters, which in general may be applied to either upstream or downstream service flows. A Service Flow Encoding defines the contents of the Provisioned, Admitted, or Active QoS Parameter Set for a service flow. Each set consists of multiple QoS parameters that define individual attributes of the Service Flow.

The MTA MUST specify:

- which J.112 service to use (e.g. unsolicited grant, real-time polled, etc.);
- what QoS parameters to associate with the corresponding Service Flow.

The choice of service class will affect both latency and efficiency. An unsolicited grant service will introduce additional latency no greater than the amount of time between grants. A polled service has the potential to introduce greater latency since the CM waits for a polling cycle and then for a grant to be made.

To decide whether to use the unsolicited grant mechanism or the real time polling mechanism, the MTA MAY use both policy information and the characteristics of the source as described in the QoS requirements for the session. In general, it makes sense to use unsolicited grants only if the source exhibits CBR like characteristics with a fixed packet size once every fixed time interval.

For UGS, the grant interval can be set to the packet formation time, although different values can be used depending on the latency and jitter requirement.

For example, consider a VoIP application that uses G.729E and the following SDP:

```
c = IN IP4 10.1.1.10
m = audio 3456 RTP/AVP 96
a = rtpmap:96 G729E/8000
```

a = ptime:10

where the rtpmap specifies the codec parameters, and ptime specifies the packet formation time of 10 ms. This can be mapped to Upstream Service Flow QoS parameters as:

- Unsolicited grant service.
- Grant size of 86 bytes (55 bytes for the IP packet, as given by the Flowspec, and 31 bytes of J.112 MAC layer overhead).
- Grant interval of 10 ms.

The Upstream PDU size MUST take into account the Ethernet overhead (18 bytes) as well as any J.112 overhead (typically 6-13 bytes). Payload Header Suppression has the potential to reduce PDU size by up to 42 bytes, depending on the use of the UDP checksum and the IP ident field, to which is added two bytes of J.112 extended header giving the value of PHS Index.

If UDP checksum not used and IP Ident field to be suppressed - 40 bytes subtracted from PDU size.

If UDP checksum is used and IP Ident field to be suppressed - ?? bytes subtracted from PDU size.

If UDP checksum not used and IP Ident field cannot be suppressed - ?? bytes subtracted from PDU size.

If UDP checksum is used and IP Ident field cannot be suppressed - ?? bytes subtracted from PDU size.

The upstream classifier MUST be set as follows. The Source Address is the MTA IP address. The Source Port is the port number on which the MTA will be sending the voice stream. The Destination Address is the destination IP address obtained from the c = line of the far-end SDP description. The Destination Port is the port number obtained from the m = line of the far-end SDP description. The protocol type is UDP.

The downstream classifier MUST be set as follows. The Source Address is the remote MTA IP address, obtained from the c = line of the far-end SDP description. The Source Port is not available in the SDP description, and SHOULD NOT be specified as part of the classifier. The Destination Address is the MTA IP address. The Destination Port is the local port on which the MTA has indicated it will receive the voice data packets. The protocol type is UDP.

The Upstream PHS Mask MUST be set to a bit string, one bit per byte of the packet, with the first bit corresponding to the first byte of the Ethernet header. All bits SHOULD be set to one, with the exception of the bits corresponding to the IP ident field, the IP checksum field, and the UDP checksum field, if those fields cannot be suppressed.

The Upstream PHS Field MUST be set to the byte string that the AN is to restore at the beginning of every packet, consisting of the value of the Ethernet Header, IP Header, and UDP Header. IP Ident, IP checksum, and UDP checksum bytes MUST be skipped in the PHS Field if they are not being suppressed.

The Downstream PHS Size SHOULD be set to 32 bytes. This amount includes the SA and Type of the Ethernet header (8 bytes), the full IP header (20 bytes), and UDP packet length and Destination Port (4 bytes). Not suppressed are the UDP Source port, the UDP checksum, and the Destination Address of the Ethernet header.

The Downstream PHS Mask SHOULD be set to 0xffffffff, indicating all the bytes listed above, starting after the Ethernet DA, are suppressed.

The Downstream PHS Field MUST be set to the byte string that the CM is to restore at the beginning of every packet, consisting of the value of the Ethernet Source Address (MAY be set to the address of the AN, or MAY be set to anything else convenient to the MTA), the IP header, UDP packet length, and Destination Port value.

B.2 J.112 Support for Resource Reservation

In J.112 there is no defined way of passing authorization information from the CM to the *Authorization Module* within the AN. The Authorization Module is a logical function of the AN defined in J.112. The present document utilizes a new J.112 TLV which passes an Authorization Block consisting of an arbitrary string of length *n* to the AN to be interpreted and processed only by the Authorization Module.

The DQoS model is one in which each session is authorized. The authorization of each session uses a handle given to both the AN and to the MTA, which is used to match requests with authorizations. This handle is the Gate-ID. Upon receiving call signalling information, the MTA passes the Gate-ID to the AN using the AuthBlock TLV contained in a DSA/DSC message.

An example of the use of the Authorization Block is found as part of the DSA-REQ messages in annex J.

B.2.1 Two-Phase QoS Reservation/Commit

A Service Flow has three associates sets of Quality of Service Parameters, referred to as the Provisioned, Admitted, or Active QoS Parameter Set. The relationship between these is identical to the description of Authorized, Reserved, and Committed resources given in clause 5.7.4. In addition, a vendor-specific option in J.112 is the ability to support multiple Admitted QoSParameterSets for a single Service Flow.

The Reserve and Commit operations are both performed by the use of J.112 Dynamic Service messages, by changing the values of the AdmittedQoSParameterSet and ActiveQoSParameterSet of the Service Flow. In a Dynamic Service Addition (DSA) or Dynamic Service Change (DSC) message, Reserve is accomplished by including, in the Upstream Service Flow Encodings or Downstream Service Flow Encodings, the QoSParameterSetType TLV with value set to Admitted (value 2). Similarly, Commit is accomplished by setting the QoSParameterSetType TLV to Active (value 4) or Admitted+Active (value 6).

DSA and DSC exchanges between the CM and AN are three-way handshakes, consisting of a request message followed by a response followed by an acknowledgement. This is illustrated in figure 28.

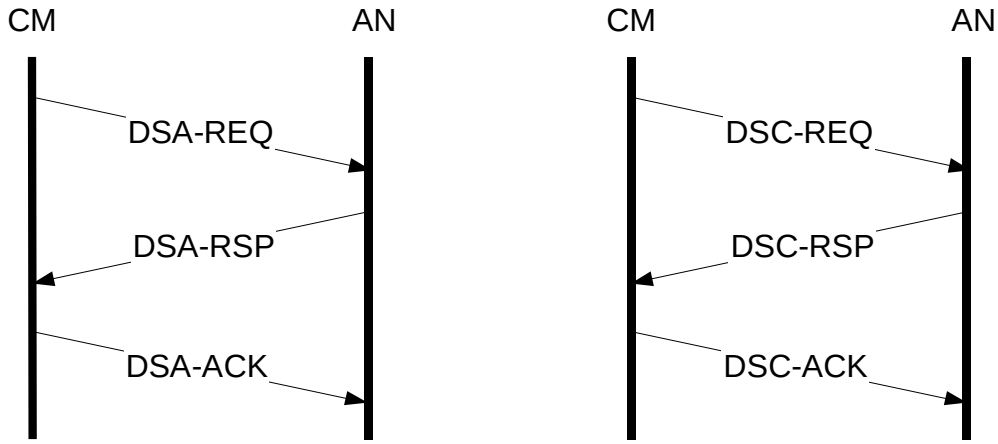


Figure 28: DSA and DSC exchanges between CM and AN

For example, the following DSA-REQ message causes the Upstream and Downstream Service Flows to be admitted, meaning the QoS resources to be used in the J.112 network are reserved.

DSA-REQ

TransactionID		1
UpstreamServiceFlow	ServiceFlowReference	1
	QoSParameterSetType	Admitted (2)
	ServiceFlowScheduling	UGS (6)
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	222
DownstreamServiceFlow	ServiceFlowReference	2
	QoSParameterSetType	Admitted (2)
	TrafficPriority	3
	MaximumSustainedRate	12 000

As a further example, the following DSC-REQ message causes the Service Flow to be activated, meaning the QoS resources used in the J.112 network are committed.

DSC-REQ

TransactionID		1
UpstreamServiceFlow	ServiceFlowID	10 288
	QoSParameterSetType	Admitted + Active (6)
	ServiceFlowScheduling	UGS (6)
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	222
DownstreamServiceFlow	ServiceFlowID	10 289
	QoSParameterSetType	Admitted + Active (6)
	TrafficPriority	3
	MaximumSustainedRate	12 000

Parameters such as ToleratedGrantJitter and TrafficPriority MAY be supplied by provisioning, or MAY be determined by the implementation of the MTA. It is anticipated that values proposed by the MTA may be overridden by policy in the AN.

Specification of Admitted and Activated QoS parameter sets by the MTA is via the MAC_CREATE_SERVICE_FLOW.request and MAC_CHANGE_SERVICE_FLOW.request. By the time a Service Flow is admitted, it typically has associated classifier(s). See annex J for further examples.

B.2.2 Reservation with Multiple Service Flow Specifications

There are various situations in which a reservation needs to cover a range of possible specifications. For example, some applications desire to create a reservation which can handle a switch from one flow specification to another mid-session without having to pass through admission control at each switch-over time. In order for the ActiveQoSParameterSet of a Service Flow to vary during a session, a suitable AuthorizedQoSParameterSet needs to be specified through policies at the Gate Controller.

Per J.112 it may be possible (vendor option) to have more than one Admitted set of QoSParameters. For example:

DSA-REQ

TransactionID		1
UpstreamServiceFlow	ServiceFlowReference	1
	QoSParameterSetType	Admitted (2)
	ServiceFlowScheduling	UGS (6)
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	UnsolicitedGrantSize	111
UpstreamServiceFlow	ServiceFlowReference	1
	QoSParameterSetType	Admitted (2)
	ServiceFlowScheduling	UGS (6)
	NominalGrantInterval	20 ms
	ToleratedGrantJitter	2 ms
	UnsolicitedGrantSize	444

This causes the AN to reserve resources such that either of the described flows may be later activated, and the AN cannot return an error due to "insufficient resources" on the activation attempt. However, the AN may reject such a reservation request with a 2-reject-unrecognized-configuration-setting. In that case, the MTA MUST use a least-upper-bound approach to resource reservation.

The least-upper-bound of two parameter sets is formed by taking, for each dimension of the resource reservation, the maximum resource required by any individual flow specification. This usually yields an over-estimate of the resources that will be required by the MTA, but is the best that can be done within the facilities available. Using the two service specifications from the example above, a DSC-REQ message that reserved resources for both flows but committed resources for only the first would be:

DSC-REQ

TransactionID		1
Upstream Service Flow	ServiceFlowID	10 288
	QoSParameterSetType	Admitted (2)
	ServiceFlowScheduling	UGS (6)
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	UnsolicitedGrantSize	444
UpstreamServiceFlow	ServiceFlowID	10 288
	QoSParameterSetType	Active (4)
	ServiceFlowScheduling	UGS (6)
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	UnsolicitedGrantSize	111

In the first UpstreamServiceFlow specification, the NominalGrantInterval was given as 10 ms, the greatest common divisor of the two separate resource specifications, and the UnsolicitedGrantSize was given as 444 bytes, the maximum of the two specifications.

B.2.3 Reservation Maintenance

Whereas RSVP has a soft-state model as described in clause 6.5.4, J.112 provides only a timeout mechanism across the J.112 interface. The Service Flow QoS parameters "Timeout for Active QoS Parameters" and "Timeout for Admitted QoS Parameters" allow a session to be terminated and its resources released due to inactivity.

The TimeoutForActiveQoSParameters is intended to recover resources allocated to CMs that die, crash, or otherwise lose their connectivity to the cable network. Normal transmission of data packets on the service flow is sufficient to prevent this recovery action.

If the MTA is performing Voice Activity Detection, using a service flow scheduling type of UGS/AD, then during extended silence periods the MTA MUST perform a DSC-REQ operation to reset the timer, or MUST send periodic data packets on the service flow. Alternatively, the MTA MAY set this timer to a value zero (i.e. no checking) if it employs VAD.

When a session is terminated, the AN sends the Gate-Close message, with appropriate error code, as described in clause 8.2.

The TimeoutForAdmittedQoSParameters is intended to recover resources that are reserved by a CM but not committed. In typical cases, the committed parameters will be identical to the reserved parameters, and this will not be a problem. When the reservation includes multiple service flow specifications, such as those described in clause B.2.2, or when the commitment is for less than the reservation, it is necessary to periodically reset the AN timer. This is accomplished by performing a DSC-REQ operation that reserves the same resources as previous.

B.2.4 Support for Dynamic Binding of Resources

Dynamic binding of resources, as required in clause 5.7.7 and described in clause 6.1.4, is accomplished in J.112 through the use of Dynamic-Service-Change messages on an established Service Flow, changing the classifiers associated with the Service Flow.

B.2.5 QoS Parameter Mapping for Authorization

The Gate identified by the GateID is parameterized by RSVP objects (FlowSpec). The Authorization Module in the AN MUST convert the Gate parameter into J.112 QoS parameters using the rules defined in clauses B.3.4 and 7.1. The resulting converted J.112 QoS objects MUST then be verified against the corresponding Service Flow QoS envelopes.

For example, if the Upstream Authorization is given as:

- bucket depth (b) = 120 bytes
- bucket rate (r) = 12 000 bytes/s
- peak rate (p) = 12 000 bytes/s
- min policed unit (m) = 120 bytes
- maximum datagram size (M) = 120 bytes

The authorization will be converted into J.112 QoS parameters:

- Scheduling: UGS
- Nominal Grant Interval: 10 ms
- Tolerated Grant Jitter: 5 ms
- Unsolicited Grant Size: 151 Bytes

This converted J.112 objects will be checked against the resource envelope of the corresponding Service Flow.

B.2.6 Automatically-Committed Resources

If the individual gate was marked with the "auto-commit" flag (see clause 7.3.2.5) then the resources reserved are immediately activated, but the state of the gate is unchanged.

In the case of a non-RSVP embedded MTA, where resource reservation is initiated by the MTA with a J.112 DSA-REQ, the AN MUST initiate a J.112 DSC-REQ exchange with the MTA upon completion of the reservation establishment, with a QoSParameterSetType of Admitted+Active (value 6) for the service flow to be committed. See annex K for an example.

B.3 Use of J.112 MAC Control Service Interface

The J.112 QoS parameters for the Service Flow derived from the SDP description are signalled to establish the Service Flow(s). In this clause, we describe how this can be done using the J.112 MAC control service interfaces (Appendix B.IV of J.112).

At the level of J.112 MAC Control Service Interface primitives, the Embedded MTA signals for QoS resources as follows:

1) MAC_CREATE_SERVICE_FLOW.request

As described in B.IV.3.2, the Embedded MTA can request that a Service Flow be added via this primitive. This primitive may also be used to define classifiers for the new Service Flow, as well as supply the Admitted and Active QoS Parameter Sets of the Service Flow. The success or failure of the primitive is indicated via the MAC_CREATE_SERVICE_FLOW.response primitive.

2) MAC_CHANGE_SERVICE_FLOW.request

The Embedded MTA can initiate a change in the Admitted and Active QoS Parameter Sets via this primitive. One possible scenario is the case of putting a caller on hold. The success or failure of the primitive is indicated via the MAC_CHANGE_SERVICE_FLOW.response primitive.

3) MAC_DELETE_SERVICE_FLOW.request

When the Embedded MTA no longer needs the Service Flow, it issues a MAC_DELETE_SERVICE_FLOW.request to the Embedded CM to zero the Active and Admitted QoS Parameter Sets of the Service Flow.

The parameters of these primitives match the parameters associated with the DSA, DSC, and DSD messages as given in J.112 Appendix B.II.

B.3.1 Reservation Establishment

The MTA initiates the reservation of QoS resources by use of the MAC_CREATE_SERVICE_FLOW.request primitive. The MTA MUST include the Gate-ID in the Authorization Block TLV. Upon reception of this message, the MAC layer of the CM invokes DSA signalling by sending a DSA_REQ to the AN. The AN MUST check the authorization based on the Gate-ID (contained in the Authorization Block TLV), and reject the request if the gate is invalid or the authorized resources are insufficient for the request. Upon receiving the DSA_RSP from the AN, the MAC service notifies the upper layer using the MAC_CREATE_SERVICE_FLOW.response message. This is illustrated in the following figure.

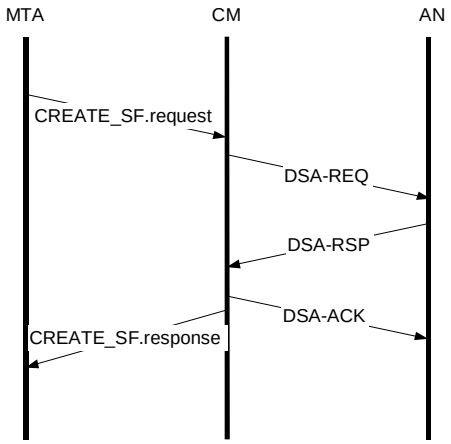


Figure 29: Reservation Establishment

B.3.2 Reservation Change

The MTA initiates changes in QoS resources by use of the MAC_CHANGE_SERVICE_FLOW.request primitive. This is illustrated in the following figure.

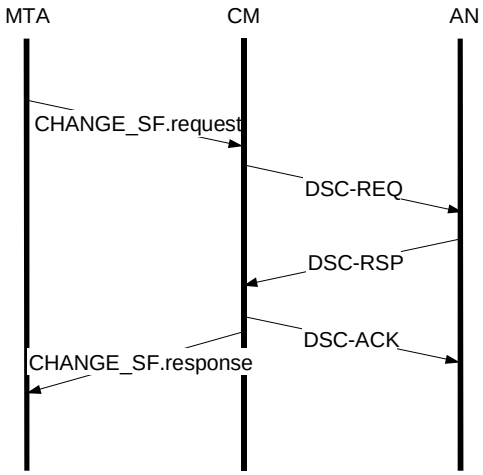


Figure 30: Reservation Change

Upon reception of this message, the MAC layer of the CM invokes DSC signalling. Upon receiving the DSC_RSP from the AN, the MAC service notifies the upper layer using the MAC_CHANGE_SERVICE_FLOW.response message.

B.3.3 Reservation Deletion

The MTA initiates the de-allocation of QoS reservation by use of the MAC_DELETE_SERVICE_FLOW.request primitive. Upon reception of this message, the MAC layer invokes DSD signalling. Upon receiving the DSD_RSP from the AN, the MAC service notifies the upper layer using the MAC_DELETE_SERVICE_FLOW.response message. This is illustrated in the following figure.

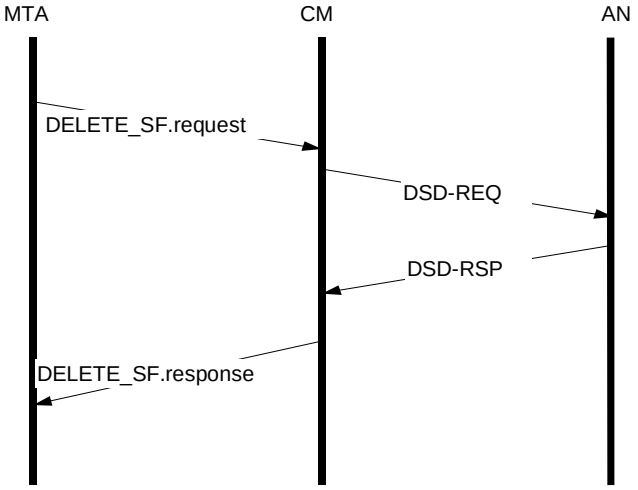


Figure 31: Reservation Deletion

B.3.4 Mapping RSVP Flowspecs into J.112 QoS Parameters

The AN, on receiving a reservation request, decides:

- what type of J.112 service to use (e.g. unsolicited grant, real-time polled, etc.);
- what QoS parameters to associate with the corresponding Service Flow.

The choice of service class will affect both latency and efficiency. An unsolicited grant service will introduce additional latency no greater than the amount of time between grants. A polled service has the potential to introduce greater latency since the CM waits for a polling cycle and then for a grant to be made.

To decide whether to use the unsolicited grant mechanism or the real time polling mechanism, the AN MAY use both policy information and the characteristics of the source as described in the TSpec. In general, it makes sense to use unsolicited grants only if the source exhibits CBR like characteristics with a fixed packet size once every fixed time interval. Such a CBR source could be identified by having a peak rate (p) nearly equal to the average rate (r) in the Sender-Tspec, and a burst size (b) equal to the maximum packet size (M). Policy information could be used to determine how close p would be to r, and b to M, before an unsolicited grant mode would be used.

For bursty VBR-like sources, the source burstiness would result in a peak rate (p) "average rate (r) and b" M in the TSpec and real time polling mode SHOULD be used.

For VoIP sources described in the present document, with p = r and M = b, Unsolicited Grant Service SHOULD be used.

Once the AN has picked a scheduling mechanism, it MAY provide information to its RSVP neighbour in the form of an AdSpec. The AdSpec allows the AN to advertise the extent to which its behaviour deviates from "ideal", i.e. the amount of additional delay that it may introduce. This delay has two parts:

- A fixed component, e.g. delay that might be introduced while processing a routing update, propagation delays, etc. (represented as D in the above delay formula).
- A rate-dependent component, e.g. due to the interval between grants, which becomes less as the rate of the reservation increases (represented as C in the above formula).

The AN MAY determine both delay components based on whether it has chosen a polled or unsolicited grant service given the Sender-Tspec. In the case of the rate-dependent component, the AN uses the maximum datagram size (M) and reserved rate (r) to determine C. For example, if an AN installs a reservation of rate R bytes/second, it could make an unsolicited grant of size M bytes every M/R seconds. Thus, the advertised value of C would be M. If using a real-time polled service, the AN MUST determine how long it could take for a packet queued at the CM to receive a grant given the polling interval that will be used, link propagation delays etc. Those factors may have fixed and rate-dependent components, which the AN SHOULD advertise accordingly.

To set the nominal grant interval the AN MUST use the rate parameter from the RSpec (R) and the maximum datagram size M. As noted above, a grant interval of M/R will provide the appropriate reservation rate. However, if the slack term permits additional delay to be introduced, the AN MAY offer larger grants less frequently, e.g. a grant of 2M bytes every 2M/R seconds.

For Unsolicited Grant Service, the AN MUST use the "Maximum Datagram Size (M)" of the TSpec in bytes to compute the Unsolicited Grant Size in minislots (after computing link level overhead) for the upstream channel on which the calling client lies.

The other key parameter that is necessary for a UGS Service Flow is the Tolerated Grant Jitter. A client needing less stringent jitter than the best case MAY pick a non-zero value for the slack term S, which gives the AN additional latitude to increase jitter if necessary. An example jitter calculation is given in clause B.3.4.1.

For real-time polled service, the polling interval MAY be a function of rate, or it MAY be fixed. For example, a polling interval of M/R would enable the CM to send one maximum sized packet each polling interval to sustain its average rate. Longer or shorter polling intervals MAY be used but will affect the total delay.

The AdSpec MAY be used to convey information about the coding delay introduced by the sender's Codec. This would be included in the D term, and the AN MUST add its own delay components to the AdSpec in calculating the tolerance for increased jitter.

The AN uses the Session Object and Sender Template to generate the upstream classifier, and uses the Reverse Session Object and Reverse Sender Template to generate the downstream classifier.

B.3.4.1 Example of Mapping

Consider the following example. A voice codec produces a CBR output data stream of 64 kbps which is packetized at 10 ms intervals, thus producing an 80 byte payload each 10 ms. The payload is encapsulated using RTP/UDP/IP, an extra 40 bytes, yielding a 120 byte packet each 10 ms. The TSpec in this case is:

- bucket depth (b) = 120 bytes
- bucket rate (r) = 12 000 bytes/s

- peak rate (p) = 12 000 bytes/s
- min policed unit (m) = 120 bytes
- maximum datagram size (M) = 120 bytes

Suppose a client requests a reservation using this TSpec and an RSpec with R = r. An AN receiving this request will establish a Service Flow that uses Unsolicited Grant Service because p = r and M = b, indicating a CBR flow. It may use a grant size of M bytes at an interval of M/R = 10 ms.

For the calculation of jitter, the MTA does not know how much the AN deviates from ideal in its scheduling behaviour. The client should assume that the AN is ideal, which means that the delay it will experience with the above TSpec and its reserved rate R = r is simply:

b/r + propagation delays

Ignoring the propagation delay, this results in a delay of 10 ms. Suppose that the client is willing to tolerate a 15 ms delay for this session (on the client-AN path only). It would then set its slack term (S) to 15–10 = 5 ms. On receiving the reservation, the AN interprets this as an indication that a 5 ms grant jitter is acceptable to the client.

Suppose that the client is willing to tolerate a 25 ms delay, and sets its slack term to 25–10 = 15 ms. The AN may use this information to determine that it can use a longer grant interval, e.g. 20 ms, since this potentially increases delay up to 20 ms for a packet that arrives at the CM right after a grant. There is still 5 ms of slack left, which the AN may use to set the grant jitter.

NOTE: This approach leaves considerable flexibility in the AN to meet the requirements of the client with regard to delay in whatever way best matches the capabilities of the AN.

B.3.4.2 Payload Header Suppression and VAD

If the AN and CM perform header suppression, then the bandwidth that is needed on a Service Flow can be reduced. The client MUST convey to the AN the fact that suppression may be applied prior to the installation of a reservation to ensure that appropriate bandwidth is reserved. The general solution to this problem is described in draft-davie-intserv-compress-00. The sender (client) adds a parameter (Compression_Hint), described in *Integrated Services in the Presence of Compressible Flows*, to the Sender-Tspec that identifies the type of compression or header suppression that might be applied to the data. The Compression_Hint parameter contains a Hint field that advertises the type(s) of compression that is possible.

An MTA that desires the CM to perform header suppression MUST include the Compression_Hint parameter, *Integrated Services in the Presence of Compressible Flows*, in the Tspec. The Compression factor field, a percentage in the range 1 to 100 inclusive, MUST be set to an amount that yields the bandwidth savings when PHS (42 bytes) is used. The value for Compression factor varies relative to the traffic profile of the CODEC. The Hint MUST be set to one of the following values depending on the type(s) of compression/suppression the MTA desires:

0x????0001 Do not suppress UDP checksum AND Do not suppress IP-Ident field nor IP-Checksum field

0x????0002 Do not suppress UDP checksum AND suppress IP-Ident field and IP-Checksum field

0x????0003 Suppress UDP checksum AND Do not suppress IP-Ident field nor IP-Checksum field

0x????0004 Suppress UDP checksum AND suppress IP-Ident field and IP-Checksum field

NOTE: ???? = TBD IANA number for IPCablecom.

Note that suppression of the IP Ident field will create problems if the packet is subsequently fragmented within the IP network. For packets less than 576 bytes in length (Internet default value of MAX-MTU), it is reasonable to assume no fragmentation will occur. The MTA SHOULD NOT request the IP-Ident field be suppressed if it will be sending packets longer than 576 bytes.

An AN connected to a CM that is capable of performing header suppression uses the Compression_Hint parameter [*Integrated Services in the Presence of Compressible Flows*] to reduce the effective rate and depth of the token bucket supplied by the sender. If header suppression is not supported on a link, the Compression_Hint parameter is ignored and the full TSpec is used.

When performing header suppression on a J.112 link, it is also necessary to communicate the *contents* of the header that will be suppressed to the AN in advance of the first data packet's transmission so that the suppression context can be established at the CM and the AN. All this information is present in the RSVP message that is used to establish the reservation, including source and destination IP addresses and ports. Since PATH messages are processed by any intermediate hops between the client and the AN, an arriving PATH message will contain the same TTL value as data packets, provided PATH messages and data packets have the same initial TTL when sent by the client. The AN MUST use the contents of the PATH to learn the values of the fields that will be suppressed. The AN MUST use J.112 MAC messaging to convey to the CM the fact that suppression should be used for a particular flow, and instructs it to suppress appropriate fields given the presence or absence of UDP checksums and IP Sequence numbers.

If the MTA initiates a PATH message specifying a wildcard sender, then no contents of the PHS field can be accurately determined. The AN MUST specify the PHS Size so the CM can accurately assess the resource needs of the service flow.

The same basic approach enables support of Voice Activity Detection (VAD). An AN may use different scheduling algorithms for flows that are using VAD, and thus needs to know which flows may be treated with VAD. The Compression_Hint parameter carried in the Tspec MUST contain the flag bit to indicate that the data flow for which this reservation is being requested may be treated with VAD.

Annex C (normative): Timer Definitions and Values

Several timers are referenced in the present document. This annex contains the list of those timers, and their recommended values.

Timer-T0

This timer is implemented in the AN in the Gate state machine, and limits the period of time that a gate may be allocated without the gate parameters being set. This enables the AN to recover the gate-ID resources when the Call Signalling System fails to complete the signalling sequence for a new session.

This timer is started when a gate is allocated.

This timer is reset when the gate parameters are set.

On expiration of this timer, the AN MUST consider the assigned gate-ID to be invalid.

The RECOMMENDED value of this timer is 30 s.

Timer-T1

This timer is implemented in the AN in the Gate state machine, and limits the period of time that may elapse between the authorization and a commit is performed.

This timer is started whenever a Gate is established.

This timer is reset whenever a Commit operation is performed on the resources authorized by the gate.

On expiration of this timer, the AN MUST revoke any reservations made by the MTA that were authorized by this gate, release all resources reserved in the AN, initiate a GATE-CLOSE message for any gate opened, and signal the CM via J.112 MAC messages to release resources it had reserved.

Timer-T1 MUST be set to the value given in the GATE-SET message. If the value given in the GATE-SET message is zero, then Timer-T1 MUST be set to a provisionable default value. The RECOMMENDED value of this default is in the range 200-300 s.

Timer-T2

This timer is implemented in the AN in the Gate state machine, and limits the time in the transient states of gate co-ordination. This timer is long enough to accommodate loss and retransmission of Gate co-ordination messages, but is short enough to not allow significant theft of service.

This timer is started when the AN receives a COMMIT message, or when the AN receives a GATE-OPEN message.

This timer is reset when the AN has received both a COMMIT message and a GATE-OPEN message for the gate.

On expiration of this timer, the AN MUST revoke any reservations made by the MTA that were authorized by this gate, release all resources reserved in the AN, release all resources activated by the AN, and signal the CM via J.112 MAC-layer specific signalling mechanisms to release resources it had reserved or activated, and use GATE-CLOSE to close any open gate.

Timer-T2 MUST be set to the value given in the GATE-SET message. If the value given in the GATE-SET message is zero, then Timer-T2 MUST be set to a provisionable default value. The RECOMMENDED value of this default is 2 s.

Timer-T3

This timer is implemented in the MTA or AN in the handling of RSVP reservations. It controls the total time that can elapse before the RSVP retransmit process gives up without receiving an acknowledgement in the presence of network loss. It is short enough to recover quickly from lost messages and not significantly impact the post-dial delay, but is long enough to allow the AN to acknowledge the request and all intermediate routers in the customer network.

This timer is started when the MTA or AN sends an RSVP message that requires an acknowledgement (such as RSVP-PATH). This timer is reset when the sender of the message to be acknowledged receives a response to that message. In the case of an RSVP-PATH message, such a response MAY be RSVP-RESV, RSVP-PATH-ERROR, or RSVP-MESSAGE-ACK, or RSVP-MESSAGE-NACK.

On expiration of this timer, the RSVP retransmit procedure ends.

The RECOMMENDED value of this timer is 4 s (4 000 ms).

Timer-T4

This timer is implemented in the MTA in the handling of COMMIT messages. It controls the retransmission of COMMIT messages that may have been lost by the network. It is short enough to recover quickly from lost commit requests and not significantly impact the post-pickup delay, but is long enough to allow processing of the COMMIT request at the AN.

This timer is started when the MTA sends a COMMIT message.

This timer is reset when the MTA receives a COMMIT-ACK or COMMIT-ERR message that is recognized as a response to the COMMIT.

On expiration of this timer, the MTA re-sends the COMMIT message.

The RECOMMENDED value of this timer is 500 ms.

Timer-T5

This timer is implemented in the AN (and AN-proxy) in the gate co-ordination processing. It controls the retransmission of GATE-OPEN and GATE-CLOSE messages that may have been lost by the network. It is short enough to recover quickly from lost gate co-ordination messages, but long enough to allow processing of the gate co-ordination message at the AN or AN-proxy. This timer interacts in the case of GATE-OPEN with Timer-T2, and SHOULD be significantly smaller than Timer-T2.

This timer is started when the AN (or AN-proxy) sends a GATE-OPEN/GATE-CLOSE message.

This timer is reset when the AN (or AN-proxy) receives a GATE-OPEN-ACK/GATE-CLOSE-ACK message that is recognized as a response to the GATE-OPEN/GATE-CLOSE.

On expiration of this timer, the AN (or AN-proxy) re-sends the GATE-OPEN/GATE-CLOSE message.

Retransmissions of the GATE-OPEN/GATE-CLOSE message is repeated for a fixed number of repetitions.

The RECOMMENDED value of this timer is 500 ms.

Timer-T6

This timer is implemented in the MTA or AN in the handling of RSVP reservations. It controls the initial delay used by the RSVP retransmit procedure.

The RECOMMENDED value of this timer is 500 ms.

Annex D (informative): Sample mapping of SDP descriptions into RSVP flowspecs

Session descriptor protocol messages are used to describe multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation per RFC 2327. This annex describes a mechanism for mapping the SDP description into RSVP flowspecs.

A typical SDP description contains many fields that contain information regarding the session description (protocol version, session name, session attribute lines, etc.), the time description (time the session is active, etc.), and media description (media name and transport, media title, connection information, media attribute lines, etc.). The two critical components for mapping an SDP description into an RSVP flowspec message are the media name and transport address (m) and the media attribute lines (a).

The media name and transport address (m) are of the form:

m=<media> <port> <transport> <fmt list>

The media attribute line(s) (a) are of the form:

a=<token>:<value>

A typical IP voice communication would be of the form:

m=audio 3456 RTP/AVP 0

a=ptime:10

On the transport address line (m), the first term defines the media type, which in the case of an IP voice session is audio. The second term defines the UDP port to which the media is sent (port 3 456). The third term indicates that this stream is an RTP Audio/Video profile. Finally, the last term is the media payload type as defined in the RTP Audio/Video Profile (reference RFC 1890). In this case, the 0 represents a static payload type of u-law PCM coded single channel audio sampled at 8 kHz. On the media attribute line (a), the first term defines the packet formation time (10 ms).

Payload types other than those defined in RFC 1890 are dynamically bound by using a dynamic payload type from the range 96-127, as defined in RFC 2327, and a media attribute line. For example, a typical SDP message for G.726 would be composed as follows:

m=audio 3456 RTP/AVP 96

a = rtpmap:96 G726-32/8000

The payload type 96 indicates that the payload type is locally defined for the duration of this session, and the following line indicates that payload type 96 is bound to the encoding "G726-32" with a clock rate of 8 000 samples/s. For every defined CODEC (whether it is represented in SDP as a static or dynamic payload type) there needs to be a table mapping from either the payload type or ASCII string representation to the bandwidth requirements for that CODEC.

The mapping of RTP/AVP code to RSVP Flowspec is according to the following table, as required by the IPCablecom CODEC specification J.acr:

Table 2: Mapping of Session Description Parameters to RSVP Flowspec

Parameters from Session Description			Flowspec parameters		Comments
RTP/AVP code	Rtpmap	Ptime	Values b,m,M	Values r,p	
0	<none>	10	120 bytes	12 000 bytes/s	G.711 using the Payload Type defined by IETF
0	<none>	20	200 bytes	10 000 bytes/s	
0	<none>	30	280 bytes	9 333 bytes/s	
96-127	PCMU/8000	10	120 bytes	12 000 bytes/s	G.711 PCM, 64 kb/sec, default CODEC
96-127	PCMU/8000	20	200 bytes	10 000 bytes/s	
96-127	PCMU/8000	30	280 bytes	9 333 bytes/s	
96-127	G726-16/8000	10	60 bytes	6 000 bytes/s	
96-127	G726-16/8000	20	80 bytes	4 000 bytes/s	
96-127	G726-16/8000	30	100 bytes	3 333 bytes/s	
96-127	G726-24/8000	10	70 bytes	7 000 bytes/s	
96-127	G726-24/8000	20	100 bytes	5 000 bytes/s	
96-127	G726-24/8000	30	130 bytes	4 333 bytes/s	
2	<none>	10	80 bytes	8 000 bytes/s	G.726-32, identical to G.721, which is assigned Payload Type 2 by IETF
2	<none>	20	120 bytes	6 000 bytes/s	
2	<none>	30	160 bytes	5 333 bytes/s	
96-127	G726-32/8000	10	80 bytes	8 000 bytes/s	
96-127	G726-32/8000	20	120 bytes	6 000 bytes/s	
96-127	G726-32/8000	30	160 bytes	5 333 bytes/s	
96-127	G726-40/8000	10	90 bytes	9 000 bytes/s	
96-127	G726-40/8000	20	140 bytes	7 000 bytes/s	
96-127	G726-40/8000	30	190 bytes	6 333 bytes/s	
15	<none>	10	60 bytes	6 000 bytes/s	G.728, assigned Payload Type 15 by IETF
15	<none>	20	80 bytes	4 000 bytes/s	
15	<none>	30	100 bytes	3 333 bytes/s	
96-127	G728/8000	10	60 bytes	6 000 bytes/s	G.728, LD-CELP, 16 kb/s
96-127	G728/8000	20	80 bytes	4 000 bytes/s	
96-127	G728/8000	30	100 bytes	3 333 bytes/s	
18	<none>	10	50 bytes	5 000 bytes/s	G.729A, identical to G.729, assigned Payload Type 18 by IETF
18	<none>	20	60 bytes	3 000 bytes/s	
18	<none>	30	70 bytes	2 333 bytes/s	
96-127	G729A/8000	10	50 bytes	5 000 bytes/s	G.729A, CS-ACEL, 8 kb/s, 10 ms frame size with 5 ms lookahead
96-127	G729A/8000	20	60 bytes	3 000 bytes/s	
96-127	G729A/8000	30	70 bytes	2 333 bytes/s	
96-127	G729E/8000	10	55 bytes	5 500 bytes/s	G.729E, CS-ACELP, 11,8 kb/s, 10 ms frame size with 5 ms lookahead
96-127	G729E/8000	20	70 bytes	3 500 bytes/s	
96-127	G729E/8000	30	85 bytes	2 833 bytes/s	

Annex E (informative): Sample Protocol Message Exchanges for Basic DCS On-Net to On-Net Call for Standalone MTA

This is an informational, informal description of the relationship between the Distributed Call Signalling protocol and the Dynamic QoS methods that may be invoked at different points in the call flow. This description is not meant to be complete. While we attempt to be accurate here in this example, the DCS call signalling specification overrides this description for the specification of the call signalling flows.

When an INVITE message is issued from the originating MTA_O and arrives at the GC_O, the GC_O issues a GATE-ALLOC request to the AN_O closest to the originating MTA_O. This is a request for the allocation of a 32-bit GateID that is unique within that AN_O. This GateID is communicated to the remote AN_T in the INVITE message that is forwarded by the GC_O. In addition, the originating AN_O communicates the number of active connections (gates) that are used by MTA_O to allow the GC_O or DP to report the current activity level for the subscriber.

The terminating GC_T knows all the possible codecs that may be used for the call, as proposed by MTA_O, and can calculate an "Authorized Envelope" based on this and issue a GATE-SET command to AN_T. Alternately, GC_T can issue only a GATE-ALLOC command at this time, wait for the results of codec negotiation procedures done by MTA_T, calculate a more accurate "Authorization Envelope" after receiving the 200-OK from MTA_T, and then issue the GATE-SET command. The latter is shown in the following call flow diagrams. In either case, the GateID is allocated and given to MTA_T in the INVITE message, and MTA_T waits for the ACK signalling message to determine the final negotiated codec values.

Included in the 200-OK message from GC_T to GC_O is the GateID at the terminating end. This is provided to AN_O in the corresponding GATE-SET exchange along with the "Authorized Envelope" of Flowspec parameters.

After the 200-OK is returned to MTA_O, it knows the address of the destination MTA_T and the parameters associated with the call (codecs used), and translates these to Flowspec parameters for both directions. The originating MTA_O sends out an ACK for the 200-OK and now performs a resource reservation. When the ACK arrives at the terminating MTA_T, it has all the information necessary, and performs a resource reservation.

Reservation involves issuing a RSVP-PATH message with Flowspec parameters for both directions. The AN performs admission control, after checking the parameters against both the Authorized Envelope as well as resource availability, and acknowledges successful reservation with a RSVP-RESV message. In between, the J.112 MAC message exchange for the layer 2 resource allocation is performed by the AN and the CM. The resources required for the call are now ready to be committed. However, they await one more phase of the call signalling protocol, and the users on both ends of the call picking up the "phone" to communicate.

The second 200-OK message from MTA_T to the originating MTA_O is an indication that the two users (in this simple 2-party basic call) are ready to communicate. The terminating MTA_T sends a COMMIT message immediately after sending the 200-OK. The originating MTA_O on receiving the 200-OK acknowledges this message and issues a COMMIT message also. The COMMIT message goes from each MTA to its local AN, and causes a J.112 MAC message exchange to commit the resources to the flow. When the COMMIT is acknowledged by the ANs, the two ends may begin to communicate while receiving enhanced QoS. When the COMMIT message is received by either of the two ANs, it starts Timer-T2 that awaits reception of the Gate-Open message from the remote AN with its GateID. On receipt of the COMMIT message the ANs also record the QoS-Start event, and the Call-Answer event.

Also indicated are the Gate Co-ordination messages between the two ANs indicating to each other that the Gate has been opened, and the description (FlowSpec) of the flow expected from the other end has been exchanged. Reception of the Gate-Open message indicates that the timer at the ANs would be disabled.

On completion of the call, the MTAs send a RSVP-PATH-TEAR message to tear down the reservations. At this time, the ANs also send a Gate-Close co-ordination message to the remote AN, and a QoS-Stop event message and a Call-Disconnect event message to the Record Keeping Server.

E.1 Example Call Flow with J.112 annex A messages

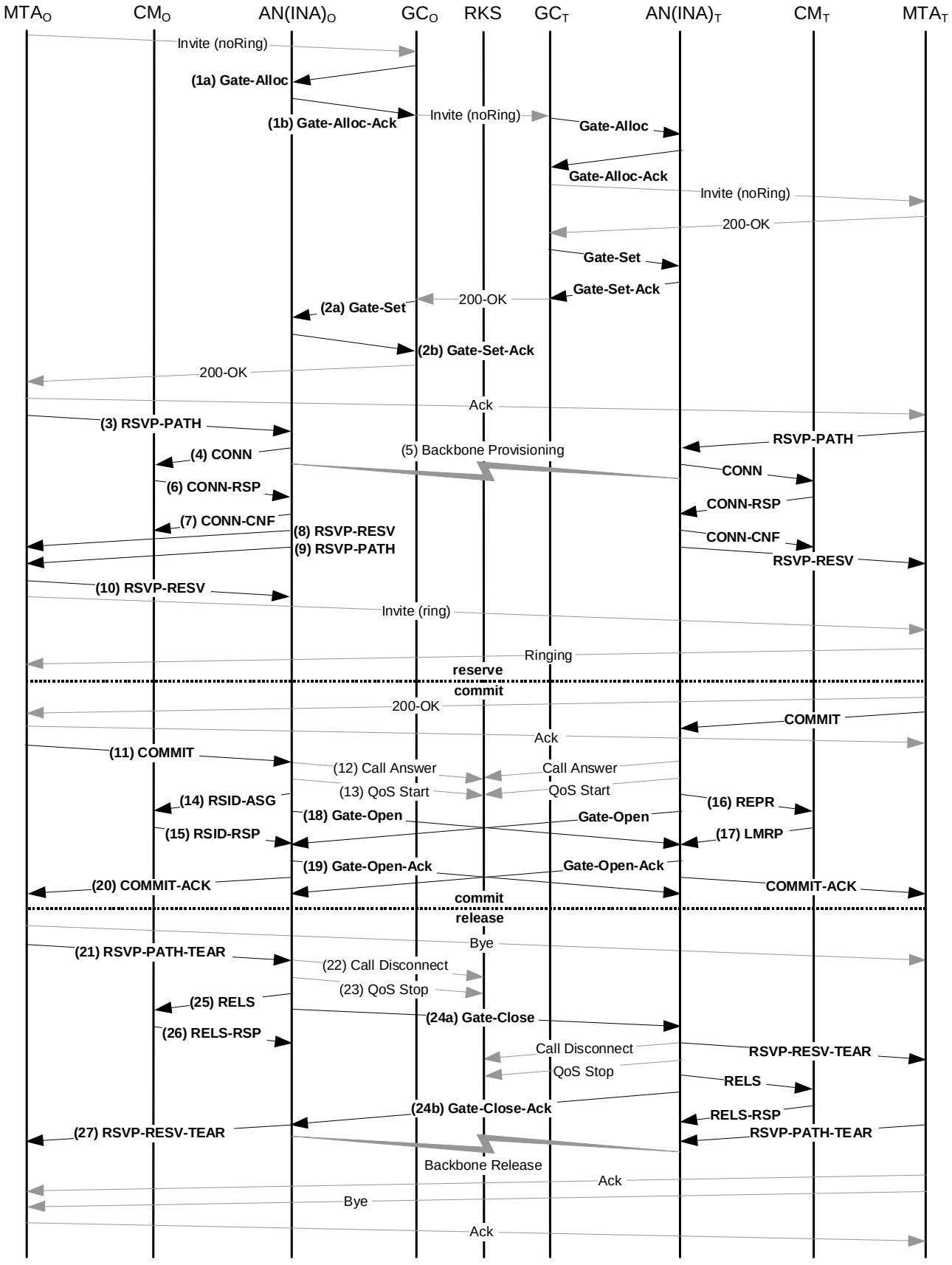


Figure 32: Basic Call Flow with J.112 annex A messages - DCS

1) GCo, upon receipt of signalling information from MTAo, checks the current resource consumption of MTAo by consulting ANo (1a).

GATE-ALLOC

TransactionID		3 176	
Subscriber		MTAo	Request for total resources in use by this endpoint.
Activity-Count		4	Maximum number of gates allowed for this subscriber.

ANo checks current resource usage by MTAo, and responds telling the number of allocated gates (1b).

GATE-ALLOC-ACK

TransactionID		3 176	
Subscriber		MTAo	Answer to request for total resources in use by this endpoint.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		3	Total number of gates established for this subscriber.
Gate Co-ordination Port		4 104	UDP port at which AN will listen for gate co-ordination messages.

2) GCo, upon further signalling exchanges, gives ANo authorization to initiate the reserve phase of the resource allocation process for the new J.112 Flow (2a).

GATE-SET

Transaction ID		3 177	Unique Transaction ID for this message exchange.
Subscriber		MTAo	Request for the specification of the previously allocated gate.
Gate-ID		37 125	Identifier for allocated Gate.
Remote-Gate-Info	Address	Ant	Information needed to perform gate co-ordination.
	Port	2 052	
	Remote Gate-ID	1 273	
	Security Key	<key>	
Event-Generation-Info	RKS-Addr	RKS	Address of Record Keeping Server.
	RKS-Port	3 288	Port on Record Keeping Server.
	Billing Correlation ID	<id>	Opaque data that will be passed to RKS when resources are committed.

GATE-SET

Media-Connection-Info	Called Number	0531-3915-2478	Fields needed for generation of Call Answer event message.
	Routing Number	???	
	Charged Number	0531-3915-2480	
	Location Routing Number	???	
Gate-Spec	Direction	up	
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAo	
	Destination Address	MTAt	
	Source port	0	
	Destination port	7 000	
	DSCP	6	Packet Type value for upstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

GATE-SET

Gate-Spec	Direction	Down	
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAt	
	Destination Address	MTAo	
	Source port	0	
	Destination port	7 120	
	DSCP	9	Packet Type value for downstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

ANo responds to the Gate-Set command with an acknowledgement (2b).

GATE-SET-ACK

TransactionID		3 177	
Subscriber		MTAo	Answer to request for the specification of the previously allocated gate.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		4	Total number of gates established for this subscriber.

3) MTAo, upon receiving call signalling information, sends an RSVP-PATH message, addressed to MTAt, but with the Router-Alert bit set in the IP header. Intermediate routers in the CPE network intercept, process, and forward this message as a normal RSVP-PATH.

RSVP-PATH

Session-Object	Protocol	UDP	The parameters identify the RSVP session, match the authorization previously sent by the GateController, and are also used for QoS classifiers.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Sender-Tspec	b	120	These are the negotiated traffic parameters actually being requested for this call. The AN calculates the actual upstream QoS parameters using these Tspec and Rspec parameters. This is a standard RSVP object, which will be interpreted by all intermediate routers in the path between the MTA and AN. NOTE 1: The HdrSuppression parameter is only used to identify the flows on which header suppression will be performed. The header suppression context is established using MAC messages.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	No	
	VAD	Off	
Forward Rspec	R	12 000	
	S	0	
Reverse-Session.	Protocol	UDP	New RSVP objects that provides the AN with sufficient information to calculate downstream traffic parameters and to generate an RSVP-PATH message for the downstream flow.
	Destination Addr	MTAo	
	Destination port	7 120	
Reverse-Sender Templ	Source Address	MTAt	
	Source port	0	
Reverse-Sender-Tspec	b	120	Negotiated traffic parameters actually being requested for this call. The AN calculates the actual downstream QoS parameters using these Tspec and Rspec parameters. This is a new RSVP object, which will be ignored by intermediate routers. NOTE 2: The HdrSuppression parameter is only used to identify the flows on which header suppression will be performed. The header suppression context is established using MAC messages.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	no	
	VAD	off	
Reverse-Rspec	R	12 000	
	S	0	
Gate ID		37 125	

4) ANo uses the RSVP-PATH message and calculates the QoS parameters for the J.112 link. ANo sends the following Connect Message to CMo. This message is used to establish both upstream and downstream parameters. Assuming that an upstream rate of 3,088 Mbit/s is used and IP packets are encapsulated using DirectIP, the upstream resources are calculated as follows. An IP packet of size 120 Byte (from Tspec) including the 5 Byte AAL 5 Trailer fits in 3 ATM cells. Thus, using Reservation Access mode ANo has to grant 3 slots every 10 ms. In Fixed-rate Access mode, a Cyclic Assignment of 3 slots at a time is required with a Maximum Distance of 60 slots. The Requested Bandwidth is 360 slots per 1 200 ms. However, no resources are allocated in the Connect Message. That indicates to CMo that the resources for that J.112 Flow are reserved but not yet committed.

CONN

Connection_ID	37 125 <Gate ID>
Session_number	<not used>
Connection_Control_Field_Aux	
Connection_control_field2_included	1 <yes>
IPv6_add	0 <no>
Priority_included	0 <no>
Flowspec_DS_included	0 <no>
Session_binding_US_included	1 <yes>
Session_binding_DS_included	1 <yes>
Encapsulation_included	1 <yes>
DS_multiprotocol_CBD_included	0 <no>
Resource_number	0x00
Connection_Control_Field	
DS_ATM_CBD_included	0 <no>
DS_MPEG_CBD_included	1 <yes>
US_ATM_CBD_included	1 <yes>
Upstream_Channel_Number	0x1
Slot_list_included	0 <no>
Cyclic_assignment	0 <no>
Frame_Length	0 <no>
Maximum_Contention_Access_Message_Length	1 <slots>
Maximum_Reservation_Access_Message_Length	50 <slots>
Downstream_MPEG_CBD	
Downstream_Frequency	472 000 000 <Hz>
Program_Number	0xA437
Upstream_ATM_CBD	
Upstream_Frequency	20 000 000 <Hz>

CONN

Upstream_VPI	0x01
Upstream_VCI	0x54AC
MAC_Flag_Set	0x01
Upstream_Rate	Upstream_3.088M
Encapsulation	DirectIP (1)
Session_binding_US	
US_session_binding_control	0x1F
NIU_client_source_IP_add	MTAo
NIU_client_destination_IP_add	MTAt
NIU_client_source_port	0
NIU_client_destination_port	7 000
Upstream_transport_protocol	UDP
Session_binding_DS	
DS_session_binding_control	0x1F
INA_client_source_IP_add	MTAt
INA_client_destination_IP_add	MTAo
INA_client_source_port	0
INA_client_destination_port	7 120
Downstream_transport_protocol	UDP
Connection_control_field2	
Upstream_modulation__included	1 <yes>
Upstream_Modulation	QPSK (1)

5) Simultaneous with message No. 4, ANo initiates any required backbone reservations for the requested quality of service. The content of this message is dependent on the particular backbone algorithms in use, and is outside the scope of the present document. The backbone router sends to ANo any required notification that the reservation is successful.

6) CMo checks the resources it is required to allocate (e.g. header suppression context, Connection IDs, classifier context), and installs the classifiers. If the operation is successful it returns the Connect Response message stating the success.

CONN-RSP

Connection_ID	37 125 <Gate ID>

7) Upon receipt of the Connect Response message, ANo acknowledges receipt with a Connect Confirm message.

CONN-CNF

Connection_ID	37 125 <Gate ID>

8) Once the J.112 reservation is complete, and the backbone reservation is successful, ANo responds to the RSVP-PATH message by sending an RSVP-RESV message. The message includes the Resource ID that is assigned by ANo to this IP flow. The RSVP-RESV message is sent with the source address of MTAt and destination address of MTAo. All intermediate routers will intercept, process, and forward this as a standard RSVP-RESV message.

RSVP-RESV

Session-Object	Protocol	UDP	These fields identify the IP flow for which the reservation is being established.
	Destination Address	MTAt	
	Destination port	7 000	
Flowspec	b	120	These fields identify the resources being reserved for this flow.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	
ResourceID		1	New Resource ID created for this reservation.

9) If the address of the previous hop in the RSVP-PATH message differs from the Source Address, then the AN is required to generate a RSVP-PATH message to reserve downstream resources at all intermediate routers. This condition would only be met if the MTA was not immediately adjacent to the CM.

- For this example, assume an intermediate router exists between MTAo and CMo, but not between MTAt and CMt.
- ANo constructs a RSVP-PATH message using the Reverse Path info and sends the message to the originating MTAo. The message includes the ResourceID object.

RSVP-PATH

Session-Object	Protocol	UDP	The Session-Object and Sender-Template are faked as if the RSVP message had come from the far end.
	Destination Address	MTAo	
	Destination port	7 120	
Sender-Tspec	b	120	The Sender-Tspec came from the Reverse-Sender-Tspec in the RSVP-PATH message from MTAo. This identifies the resources that will be needed in the downstream direction (from MTAt to MTAo).
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	no	
	VAD	off	
Forward Rspec	R	12 000	
	S	0	
ResourceID		1	New Resource ID created for this reservation.

10) MTAo, in response to the RSVP-PATH, sends RSVP-RESV to MTAt. This message is sent with "router alert" set, and all intermediate routers intercept, process, and forward this message until it reaches ANo.

RSVP-RESV

Session-Object	Protocol	UDP	The Session-Object and Sender-Template are copied from the RSVP-PATH message received.
	Destination Address	MTAo	
	Destination port	7 120	
Flowspec	b	120	These values are also copied from the RSVP-PATH message, and specify the amount of resources being reserved for the flow.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	no	
	VAD	off	
	R	12 000	
	S	0	
ResourceID		1	Resource ID, copied from RSVP-PATH.

11) In response to signalling messages that indicate the call set-up has completed (i.e. the other side has gone off-hook), MTAo sends the COMMIT message to ANo. This message is directed to ANo at a UDP port determined by call signalling. The Session-Object and Sender Template give ANo enough information to identify the "gate" and to identify which reserved resources are being committed.

COMMIT

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple must match those for the Gate ID.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Gate-ID		37 125	

12) ANo sends the event record to the Record Keeping Server that the media connection has started. The format of this message is described in [J.em].

13) ANo sends the event record to the Record Keeping Server that enhanced Quality of Service has been granted to this call. The format of this message is described in [J.em].

14) The AN may commit the reserved resources either using Fixed-rate Access mode or Reservation Access mode. Upon reception of the COMMIT message, it needs to send the appropriate MAC-layer messages to complete the set-up of a J.112 Flow.

- For this example, it is assumed that ANo decides to use Reservation Access mode while ANt commits resources in Fixed-rate Access mode.
- Continuous Piggybacking is used to accommodate the CBR like characteristic of this traffic. To initiate the transmission ANo sends a Reservation ID Assignment message.

RSID-ASG

Connection_ID	37 125 <Gate ID>
Reservation_ID	0x1234
Grant_Protocol_Timeout	15 <ms>
Piggy_Back_Request_Values	
Continuous_Piggy_Back_Timeout	4 <36 ms>
GFC_11_Slots	9 <slots>
GFC_10_Slots	3 <slots>
GFC_01_Slots	1 <slots>

15) CMo sends a Reservation ID Response message showing the operation was successful.

RSID-RSP

Connection_ID	37 125 <Gate ID>
Reservation_ID	0x1234
Grant_Protocol_Timeout	15 <ms>

16) ANt at the terminating side of the call decided to provide the requested resources using Fixed-rate Access mode. To commit the resources and to initiate the transmission ANt sends a Reprovision message to CMt.

REPR

Reprovision_Control_Field	
Reprovision_Control_Aux_Field_included	0 <no>
Delete_Reservation_IDs	0 <no>
New_Downstream_IB_Frequency_included	0 <no>
New_Downstream_OOB_Frequency_included	0 <no>
New_Upstream_Frequency_included	0 <no>
New_Frame_Length_included	1 <yes>
New_Cyclical_Assignment_included	1 <yes>
New_Slot_List_included	0 <no>
New_Frame_Length	3
Number_of_Connections	1
Connection_ID	1 273 <Gate ID>
Cyclic_Assignment	
Fixedrate_Start	0x0000
Fixedrate_Dist	60
Fixedrate_Stop	0xFFFF

17) CMt sends a Link Management Response message showing the operation was successful.

LMRP

Link_Management_Msg_Number	<Reprovision Message Type Value>

18) ANo sends the gate co-ordination message to the remote ANt to inform it that the resources at the local end have been committed.

GATE-OPEN

Transaction ID		72	Identifier to match this message with its response.
Gate ID		1 273	Gate-ID at AN receiving this message.
Tspec	b	120	These are the traffic parameters actually being utilized for the resources committed to the flow in the MTAo to MTAt direction.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
Reverse Tspec	b	120	These are the expected traffic parameters being utilized for the flow in the MTAt to MTAo direction.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
HMAC			Security checksum for this message.

19) Upon reception of the GATE-OPEN message from the remote ANt, ANo responds with a GATE-OPEN-ACK.

GATE-OPEN-ACK

Transaction ID		8 096	Identifier to match this message with its request.
HMAC			Security checksum for this message.

20) ANo acknowledges the COMMIT with a COMMIT-ACK message.

COMMIT-ACK

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port may assist in matching the acknowledgement to the COMMIT message.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Gate-ID		37 125	

21) When the call is finished MTAo sends RSVP-PATH-TEAR message to the AN. For each RSVP reservation, MTAo sends a separate RSVP-PATH-TEAR message.

RSVP-PATH-TEAR

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port identify the RSVP flow.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	

22) ANo sends the notification to the Record Keeping Server that the Media Connection has terminated. The format of this event message is described in [J.em].

23) ANo sends the notification to the Record Keeping Server that the call has ended. The format of this event message is described in [J.em].

24) ANo, upon receiving RSVP-PATH-TEAR, sends the gate co-ordination message to the address given in the GATE-ET command earlier, which in the case of DCS is ANt serving MTAt (24b).

GATE-CLOSE

Transaction ID		73	Identifier to match this message with its response.
Gate-ID		1 273	GateID at the network element receiving this message.
HMAC			Security checksum for this message.

ANt responds with a GATE-CLOSE-ACK message (24b).

GATE-CLOSE-ACK

Transaction ID		73	Identifier to match this message with its request.
HMAC			Security checksum for this message.

25) ANo, upon receiving RSVP-PATH-TEAR, sends a Release message to CMo indicating the J.112 Flow that is to be deleted.

RELS

Number_of_Connections	1
Connection_ID	37 125 <Gate ID>

26) CMt releases the J.112 Flow and sends the Release Response to ANo.

RELS-RSP

Connection_ID	37 125 <Gate ID>

27) ANo sends the RSVP-RESV-TEAR to MTAo.

RSVP-RESV-TEAR

Session-Object	Protocol	UDP	These parameters identify the IP flow that is being terminated.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	

E.2 Example Call Flow with J.112 annex B/annex C messages

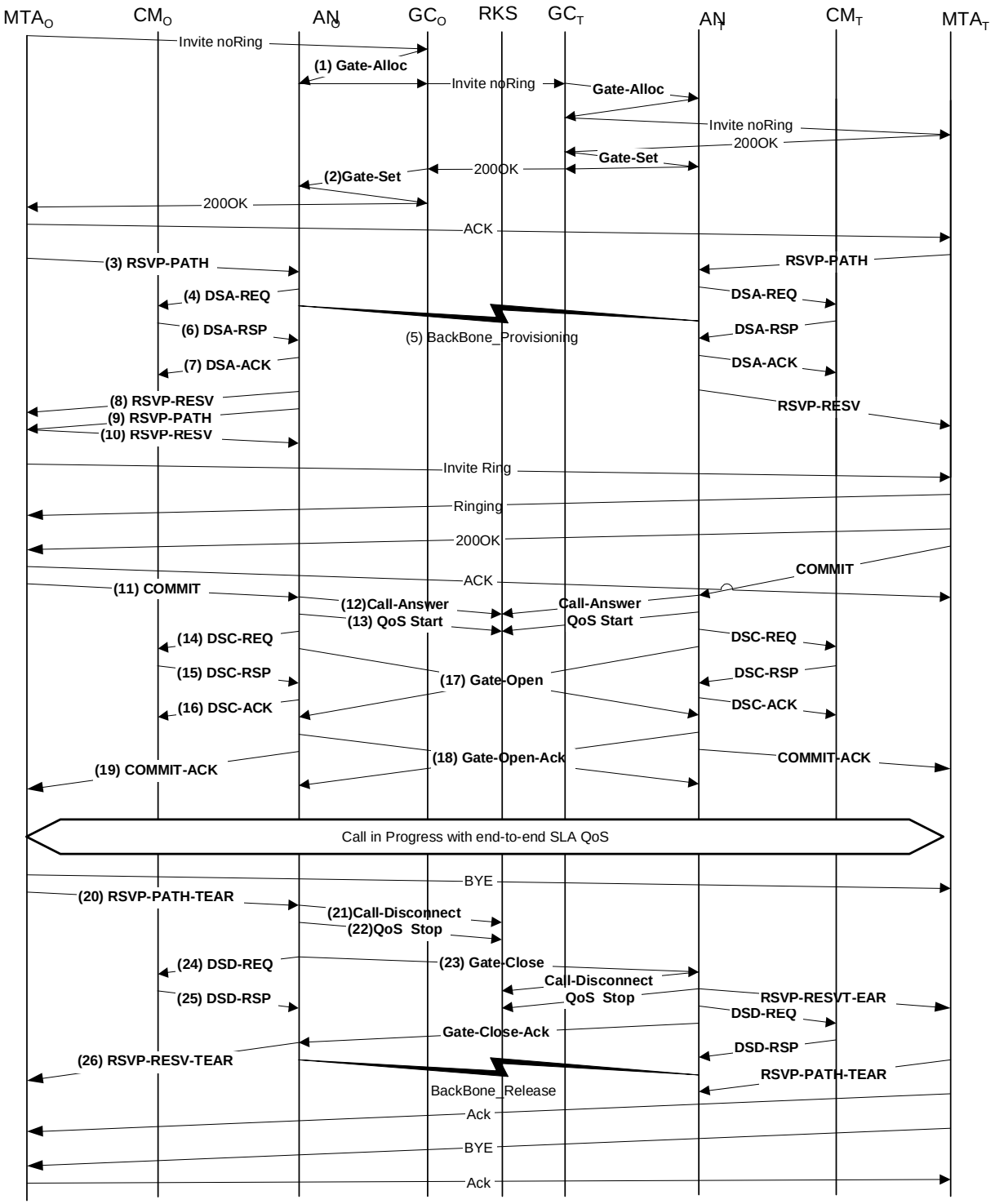


Figure 33: Basic Call Flow - DCS Signalling

1) GCo, upon receipt of signalling information from MTAo, checks the current resource consumption of MTAo by consulting ANo.

GATE-ALLOC

TransactionID		3 176	
Subscriber		MTAo	Request for total resources in use by this endpoint.
Activity-Count		4	Maximum connections allowed by client.

ANo checks current resource usage by MTAo, and responds telling the number of connections active.

GATE-ALLOC-ACK

TransactionID		3 176	
Subscriber		MTAo	Request for total resources in use by this endpoint.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		3	Total connections established by this client.
Gate Co-ordination Port		4 104	UDP port at which AN will listen for gate co-dination messages.

2) GCo, upon further signalling exchanges, gives ANo authorization to admit the new connection.

GATE-SET

Transaction ID		3 177	Unique Transaction ID for this message exchange.
Subscriber		MTAo	Request for total resources in use by this client.
Gate-ID		37 125	Identifier for allocated Gate.
Remote-Gate-Info	AN Address	ANt	Information needed to perform gate co-ordination.
	AN Port	2 052	
	Remote Gate-ID	1 273	
	Security Key	<key>	

GATE-SET

Event-Generation-Info	RKS-Addr	RKS	Address of Record Keeping Server.
	RKS-Port	3 288	Port on Record Keeping Server.
	Billing Correlation ID	<id>	Opaque data that will be passed to RKS when resources are committed.
Media-Connection-Info	Called-Number	212-555-2222	Fields needed for generation of Call-Answer message.
	Routing Number	???	
	Charged Number	212-555-1111	
	Location Routing Number	???	
Gate-Spec	Direction	up	
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAo	
	Destination Address	MTAt	
	Source port	0	
	Destination port	7 000	
	DSCP	6	Packet Type value for upstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

GATE-SET

Gate-Spec	Direction	down	
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAt	
	Destination Address	MTAo	
	Source port	0	
	Destination port	7 120	
	DSCP	9	Packet Type value for downstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

ANo responds to the Gate Setup command with an acknowledgement.

GATE-SET-ACK

TransactionID		3 177	
Subscriber		MTAo	Request for total resources in use by this client.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		4	Total connections established by this client.

3) MTAo, upon receiving call signalling information, sends an RSVP-PATH message, addressed to MTAt, but with the Router-Alert bit set in the IP header. Intermediate routers in the home LAN intercept, process, and forward this message as a normal RSVP-PATH.

RSVP-PATH

Session-Object	Protocol	UDP	The parameters identify the RSVP session, match the authorization previously sent by the GateController, and are also used for QoS classifiers.
	Destination Address	MTAt	
	Destination port	7 000	

RSVP-PATH

Sender Templ	Source Address	MTAo	
	Source port	7 120	
Sender-Tspec	b	120	These are the negotiated traffic parameters actually being requested for this call. The AN calculates the actual upstream QoS parameters using these Tspec and Rspec parameters. This is a standard RSVP object, which will be interpreted by all intermediate routers in the path between the MTA and AN.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	40	
	VAD	off	
Forward Rspec	R	12 000	
	S	0	
Reverse-Session.	Protocol	UDP	New RSVP objects that provides the AN with sufficient information to calculate downstream traffic parameters and to generate an RSVP-PATH message for the downstream flow.
	Destination Addr	MTAo	
	Destination port	7 120	
Reverse-Sender Templ	Source Address	MTAt	
	Source port	0	
Reverse-Sender-Tspec	b	120	Negotiated traffic parameters actually being requested for this call. The AN calculates the actual downstream QoS parameters using these Tspec and Rspec parameters. This is a new RSVP object, which will be ignored by intermediate routers.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	0	
	VAD	off	
Reverse-Rspec	R	12 000	
	S	0	
Gate ID		37 125	

4) The AN uses the RSVP-PATH message and calculates the QoS parameters for the J.112 link. The AN sends the following DSA-REQ to the CM. This message is used to establish both upstream and downstream parameters. The Upstream Unsolicited Grant Size was calculated as 120 (from Tspec) plus 18 (Ethernet overhead) minus 40 (Header Suppression amount) plus 13 (J.112 overhead). Header suppression, being specified as length 40 in the RSVP-PATH, indicates the 42 bytes of Ethernet/IP/UDP header. Contents of the suppressed header is taken from the RSVP packet.

DSA-REQ

TransactionID		1
UpstreamServiceFlow	ServiceFlowIdentifier	1 001
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	ServiceFlowScheduling	UGS (6)
	Request/Transmission Policy	0x00000017
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111
DownstreamServiceFlow	ServiceFlowIdentifier	2 001
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	TrafficPriority	5
	MaximumSustainedRate	12 000
UpstreamPacketClassification	ServiceFlowIdentifier	1 001
	PacketClassifierIdentifier	3 001
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAo
	IPSourcePort	7 120
	IPDestinationAddress	MTAt
	IPDestinationPort	7 000
	IPProtocol	UDP (17)
DownstreamPacketClassification	ServiceFlowIdentifier	2 001
	PacketClassifierIdentifier	3 002
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAt
	IPDestinationAddress	MTAo
	IPDestinationPort	7 120
	IPProtocol	UDP (17)

DSA-REQ

PayloadHeaderSuppression	ClassifierIdentifier	3 001
	ServiceFlowIdentifier	1 001
	HeaderSuppressionIndex	1
	HeaderSuppressionField	<42bytes>
	HeaderSuppressionMask	<42bits>
	HeaderSuppressionSize	42
	HeaderSuppressionVerify	Verify (0)
HMAC		

5) Simultaneous with message No. 2, the AN initiates any required backbone reservations for the requested quality of service. The content of this message is dependent on the particular backbone algorithms in use, and is outside the scope of the present document. The backbone router sends to the AN any required notification that the reservation is successful.

6) The CM checks the resources it is required to allocate (e.g. header suppression table space, Service Flow IDs, classifier table space, local network bandwidth), and installs the classifiers. If the operation is successful it returns the DSA-RSP message stating the success.

DSA-RSP

TransactionID		1
ConfirmationCode		Success (0)
HMAC		

7) Upon receipt of the DSA-RSP, the AN acknowledges receipt with a DSA-ACK message.

DSA-ACK

TransactionID		1
ConfirmationCode		Success (0)
HMAC		

8) Once the J.112 reservation is complete, and the backbone reservation is successful, the AN responds to the RSVP-ATH message by sending an RSVP-RESV message. The message includes the ResourceID that is assigned by the AN to this connection. The RSVP-RESV message is sent with the source address of MTA_T and destination address of MTA_o. All intermediate routers will intercept, process, and forward this as a standard RSVP-RESV message.

RSVP-RESV

Session-Object	Protocol	UDP	These fields identify the IP flow for which the reservation is being established.
	Destination Address	MTAt	
	Destination port	7 000	
Flowspec	b	120	These fields identify the resources being

	r	12 000	reserved for this flow.
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	
ResourceID		1	New Resource ID created for this reservation.

9) If the address of the previous hop differs from the Source Address, then the AN is required to generate a RSVP-ATH message to reserve downstream resources at all intermediate routers. This condition would only be met it the MTA was not immediately adjacent to the CM.

- For this example, assume an intermediate router exists between MTA-o and its CM, but not between MTA-t and its CM.
- The AN constructs RSVP-PATH message using the Reverse Path info it received from the RSVP-PATH message and sends the message to the originating MTA. The message includes the ResourceID object.

RSVP-PATH

Session-Object	Protocol	UDP	The Session-Object and Sender-Template are faked as if the RSVP message had come from the far end.
	Destination Address	MTAo	
	Destination port	7 120	
Sender-Tspec	b	120	The Sender-Tspec came from the Reverse-Sender-Tspec in the RSVP-PATH message from MTAo. This identifies the resources that will be needed in the downstream direction (from MTAt to MTAo).
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	40	
	VAD	off	
Forward Rspec	R	12 000	
	S	0	
ResourceID		1	New Resource ID created for this reservation.

10) MTAo, in response to the RSVP-PATH(7), sends RSVP-RESV to MTAt. This message is sent with "router alert" set, and all intermediate routers intercept, process, and forward this message until it reaches the AN.

RSVP-RESV

Session-Object	Protocol	UDP	The Session-Object and Sender-Template are copied from the RSVP-PATH message received.
	Destination Address	MTAo	
	Destination port	7 120	
Filter-Spec	Source Address	MTAt	
	Source port	7 000	
Flowspec	b	120	These also are copied from the RSVP-PATH message, and specify the amount of resources being reserved for the flow.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	40	
	VAD	off	
	R	12 000	
	S	0	
ResourceID		1	Resource ID, copied from RSP-PATH.

11) In response to signalling messages that indicate the call has completed (i.e. the other side has gone off-hook), MTAo sends the COMMIT message to the AN. This message is directed to the AN at a UDP port determined via call signalling.

The Session-Object and Sender Template give the AN enough information to identify the "gate" and to identify which reserved resources are being committed.

COMMIT

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple must match those for the Gate ID.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Gate-ID		37 125	

12) AN_O sends the event record to the Record Keeping Server that the Media Connection has started.

Call-Answer

Header	Time Stamp	<time>	The time of the event being recorded.
	Billing Correlation ID	<string>	Billing Correlation ID given in Gate-Set.
Called Party	Called Party Number	212-555-2222	Items provided by CMS in Gate-Set.
Routing Number	Routing Number	???	
Charged Number	Charged Number	212-555-1 111	
Location Routing Number	Location Routing Number	???	

13) ANo sends the event record to the Record Keeping Server that an enhanced Quality-of-Service connection has been granted to this call.

QoS-START

Header	Timestamp	<time>	Time of the event being recorded.
	Billing Correlation ID	<string>	Correlation ID given in Gate-Set.
QoS Descriptor	Type	UGS	Description of the QoS provided for this connection.
	Grant interval	10 ms	
	Grant Jitter	2 ms	
	Grant/Interval	1	
	Grant Size	111	
MTA Port	Port	7 120	

14) The AN resolves which reservation is to be activated, and sends a DSC-REQ to the CM to activate the flow.

DSC-REQ

TransactionID		2
UpstreamServiceFlow	ServiceFlowIdentifier	1 001
	QoSParameterSetType	Admitted + Activated (6)
	TimeOutActive	10
	ServiceFlowScheduling	UGS (6)
	Request/TransmissionPolicy	0x00000017
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111
DownstreamServiceFlow	ServiceFlowIdentifier	2 001
	QoSParameterSetType	Admitted + Activated (6)
	TimeOutActive	10
	TrafficPriority	5
	MaximumSustainedRate	12 000
UpstreamPacketClassification	ServiceFlowIdentifier	1 001
	PacketClassifierIdentifier	3 001
	ClassifierChangeAction	Replace (1)
	ClassifierPriority	150
	ClassifierActivationState	Active (1)
	IPSourceAddress	MTAo
	IPSourcePort	7 120
	IPDestinationAddress	MTAt
	IPDestinationPort	7 000
	IPProtocol	UDP (17)

DSC-REQ

DownstreamPacketClassification	ServiceFlowIdentifier	2 001
	PacketClassifierIdentifier	3 002
	ClassifierChangeAction	Replace (1)
	ClassifierPriority	150
	ClassifierActivationState	Active (1)
	IPSourceAddress	MTAt
	IPSourcePort	7 000
	IPDestinationAddress	MTAo
	IPDestinationPort	7 124
	IPProtocol	UDP (17)
HMAC		

15) The CM sends a DSC-RSP message showing the operation was successful.

DSC-RSP

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

16) The AN sends DSC-ACK message to indicate that the DSC-RSP has been received and agreed.

DSC-ACK

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

17) The AN sends the gate co-ordination message to the remote AN to inform it that the resources at this end have been committed.

GATE-OPEN

Transaction ID		72	Identifier to match this message with its response.
Gate ID		1 273	Gate-ID at remote AN.
Tspec	b	120	These are the committed traffic parameters actually being utilized in the MTAo to MTAt direction.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
Reverse-Tspec	b	120	These are the expected traffic parameters being utilized in the MTAt to MTAo direction.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
HMAC			Security checksum for this message.

18) The remote AN responds to the GATE-OPEN with:

GATE-OPEN-ACK

Transaction ID		72	Identifier to match this message with its response.
HMAC			Security checksum for this message.

19) The AN acknowledges the COMMIT with:

COMMIT-ACK

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port may assist in matching the acknowledgement to the COMMIT message.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Gate-ID		37 125	

20) When the call is finished the MTA sends RSVP-PATH-TEAR message to the AN. For each RSVP reservation, the MTA sends a separate RSVP-PATH-TEAR message.

RSVP-PATH-TEAR

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port identify the RSVP flow.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	

21) The AN sends the notification to the Record Keeping Server that the Media Connection has terminated.

Call-Disconnect

Header	Timestamp	<time>	Time of the event being recorded.
	Billing Correlation ID	<string>	Billing Correlation ID provided in Gate-Set.
Termination Cause	Cause	???	Cause code as defined by Event Messages.

22) The AN sends the notification to the Record Keeping Server that the call has ended. This message is only a sample of what might be included in a QoS-Stop message.

QoS-Stop

TimeStamp		<time>	The time of the event being recorded.
Header	Time Stamp	<time>	Time of the event being recorded.
	Billing Correlation ID	<string>	Correlation ID from Gate-Set message.
SF-ID	SF-ID	1 001	Service Flow Identifier.

23) The AN, upon receiving RSVP-PATH-TEAR, sends the gate co-ordination message to its corresponding AN serving MTAt.

GATE-CLOSE

Transaction ID		73	Identifier to match this message with its response.
Gate-ID		1 273	This identifies the GateID at the remote AN.
HMAC			Security checksum for this message.

The remote AN responds with:

GATE-CLOSE-ACK

Transaction ID		73	Identifier to match this message with its response.
HMAC			Security checksum for this message.

24) The AN, upon receiving RSVP-PATH-TEAR, sends a DSD-REQ to the CM indicating the Service Flow ID that is to be deleted.

DSD-REQ

TransactionID		3
ServiceFlowID		1 001
HMAC		

DSD-REQ

TransactionID		4
ServiceFlowID		2 001
HMAC		

25) The CM deletes the Service Flow ID and sends the response to the AN.

DSD-RSP

TransactionID		3
ServiceFlowID		1 001
ConfirmationCode		Success (0)
HMAC		

DSD-RSP

TransactionID		3
ServiceFlowID		2 001
ConfirmationCode		Success (0)
HMAC		

26) The AN sends the RSVP-RESV-TEAR to MTA.

RSVP-RESV-TEAR

Session-Object	Protocol	UDP	These parameters identify the IP flow that is

	Destination Address	MTAt	being terminated.
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	

Annex F (informative): Sample Protocol Message Exchanges for Basic NCS On-Net to On-Net Call for Standalone MTA

This is an informational description of a possible relationship between the call signalling protocol (TS 101 909-4) and the Dynamic QoS methods that may be invoked at various points in the call flow.

When the originating MTA_O completes dialling, i.e. the digit map indicates a complete phone number has been entered, the digits are sent to the CMS_O via a Notify message. The CMS_O, in its first step of initiating a new call, tells the MTA_O to create a new inactive connection. The MTA_O allocates a receive port for the media stream, and responds with an ACK message that includes the Session Description listing all the media streams the MTA_O is willing to receive. The CMS_O performs a GATE-ALLOC exchange with the AN_O to allocate a Gate-ID, and passes this information to the terminating CMS_T along with the originating SDP profile.

The terminating CMS_T sets up the gate at the terminating AN_T (using a GATE-SET command), allowing all of the media flows that are acceptable to the originator within the "Authorized Envelope" and allowing a wildcard destination port on MTA_T. The AN_T also assigns a Gate-ID and returns it to the CMS_T. The CMS_T passes the local Gate-ID to the terminating MTA_T in a Create Connection command, along with the proposed SDP profile. MTA_T, in its response, indicates the set of media streams it finds acceptable, and the allocated port for reception of those streams.

At this point, MTA_T knows the sending codec, the receiving codec, the destination address and port for voice packets it sends, and the local port for reception of voice packets. It therefore begins the reserve sequence by sending a RSVP-PATH to AN_T.

When CMS_O receives the SDP profile from MTA_T, it has sufficient information to establish the gate at AN_O. It therefore performs the GATE-SET operation, including the remote Gate-ID and the address of AN_T. CMS_O now issues a Modify Connection command to MTA_O, telling it the destination address, port, and codec to use. MTA_O now has sufficient information to make a resource reservation. When the reservation completes, it sends a successful acknowledgement to CMS_O. CMS_T now tells MTA_T to alert the user of an incoming call. MTA_T first checks that the resource reservation that it earlier initiated has completed successfully, and if so, proceeds to ring the phone.

When the called party answers, MTA_T informs CMS_T with a Notify message, indicating Offhook. CMS_T now sends a Modify Connection command to MTA_T making the connection mode send+receive; MTA_T does the COMMIT exchange with AN_T and then sends the acknowledgement. CMS_O also sends a Modify Connection command to MTA_O making its connection mode send+receive, causing MTA_O to also do the COMMIT exchange with AN_O. The call is now established.

Either party can initiate a call termination by sending a Notify message to their CMS indicating Onhook. In the diagram, MTA_O is shown doing this. CMS_O responds to the Onhook notification by sending a Delete Connection command, which triggers the RSVP-PATH-TEAR sequence to release the resources. MTA_T is informed of the hangup both by call signalling (a Delete Connection command, not shown) or by the RSVP-RESV-TEAR DQoS message. When MTA_T later goes onhook, it produces the same Notify message as was earlier sent by MTA_O, and ends the sequence.

F.1 Example Call Flow with J.112 annex A messages

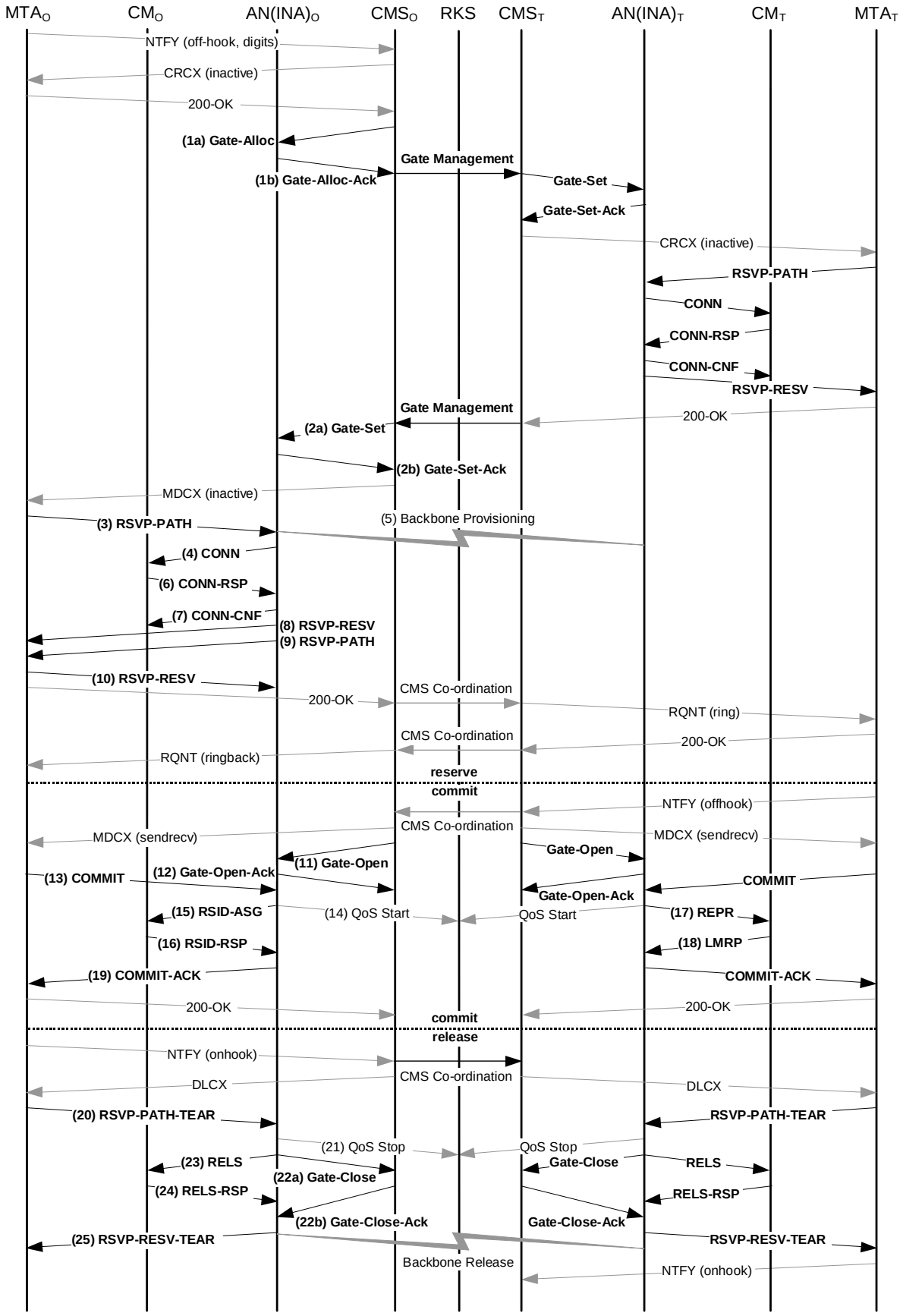


Figure 34: Basic Call Flow with J.112 annex A messages - NCS

1) GCo/CMSo, upon receipt of signalling information from MTAo, checks the current resource consumption of MTAo by consulting ANo (1a).

GATE-ALLOC

TransactionID		3 176	
Subscriber		MTAo	Request for total resources in use by this endpoint.
Activity-Count		4	Maximum number of gates allowed for this subscriber.

ANo checks current resource usage by MTAo, and responds telling the number of allocated gates (1b).

GATE-ALLOC-ACK

TransactionID		3 176	
Subscriber		MTAo	Request for total resources in use by this endpoint.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		3	Total number of gates established for this subscriber.

2) GCo/CMSo, upon further signalling exchanges, gives ANo authorization to initiate the reserve phase of the resource allocation process for the new J.112 Flow (2a).

GATE-SET

Transaction ID		3 177	Unique Transaction ID for this message exchange.
Subscriber		MTAo	Request for the specification of the previously allocated gate.
Gate-ID		37 125	Identifier for allocated Gate.
Remote-Gate-Info	Address	CMSo	Information needed to perform gate co-ordination. Note that CMSo has given itself as the entity for exchanging gate co-ordination messages. Flag value indicates that the AN should not send a Gate-Open message when it receives a COMMIT from the MTA, but still expect to receive a Gate-Open message from CMSo.
	Port	2 052	
	Remote Gate-ID	8 095	
	Security Key	<key>	
	Flag	No-gate-open	
Event-Generation-Info	RKS-Addr	RKS	Address of Record Keeping Server.
	RKS-Port	3 288	Port on Record Keeping Server.
	Billing Correlation ID	<id>	Opaque data that will be passed to RKS when resources are committed.
Gate-Spec	Direction	up	
	Protocol	UDP	The protocol, Destination Address, Source

GATE-SET

	Source Address	MTAo	Address, and Destination Port quadruple are used for QoS classifiers.
	Destination Address	MTAt	
	Source port	0	
	Destination port	7 000	
	DSCP	6	Packet Type value for upstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	
Gate-Spec	Direction	Down	
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAt	
	Destination Address	MTAo	
	Source port	0	
	Destination port	7 120	
	DSCP	9	Packet Type value for downstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

ANo responds to the Gate Setup command with an acknowledgement (2b).

GATE-SET-ACK

TransactionID		3 177	
Subscriber		MTAo	Request for the specification of the previously allocated gate.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		4	Total number of gates established for this subscriber.

3) MTAo, upon receiving a Modify-Connection command, sends an RSVP-PATH message, addressed to MTAt, but with the Router-Alert bit set in the IP header. Intermediate routers in the home LAN intercept, process, and forward this message as a normal RSVP-PATH.

RSVP-PATH

Session-Object	Protocol	UDP	The parameters identify the RSVP session, match the authorization previously sent by the GateController, and are also used for QoS classifiers.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Sender-Tspec	b	120	These are the negotiated traffic parameters actually being requested for this call. The AN calculates the actual upstream QoS parameters using these Tspec and Rspec parameters. This is a standard RSVP object, which will be interpreted by all intermediate routers in the path between the MTA and AN. NOTE: The HdrSuppression parameter is only used to identify the flows on which header suppression will be performed. The header suppression context is established using MAC messages.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	no	
	VAD	Off	
Forward Rspec	R	12 000	
	S	0	
Reverse-Session	Protocol	UDP	New RSVP objects that provides the AN with sufficient information to calculate downstream traffic parameters and to generate an RSVP-PATH message for the downstream flow.
	Destination Addr	MTAo	
	Destination port	7 120	
Reverse-Sender Templ	Source Address	MTAt	
	Source port	0	

RSVP-PATH

Reverse-Sender-Tspec	b	120	Negotiated traffic parameters actually being requested for this call. The AN calculates the actual downstream QoS parameters using these Tspec and Rspec parameters. This is a new RSVP object, which will be ignored by intermediate routers. NOTE: The HdrSuppression parameter is only used to identify the flows on which header suppression will be performed. The header suppression context is established using MAC messages.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	no	
	VAD	Off	
Reverse-Rspec	R	12 000	
	S	0	
Gate ID		37 125	

4) The AN uses the RSVP-PATH message and calculates the QoS parameters for the J.112 link. The AN sends the following Connect Message to the CM. This message is used to establish both upstream and downstream parameters. Assuming that an upstream rate of 3,088 Mbit/s is used and IP packets are encapsulated using DirectIP, the upstream resources are calculated as follows. An IP packet of size 120 Byte (from Tspec) including the 5 Byte AAL 5 Trailer fits in 3 ATM cells. Thus, using Reservation Access mode the AN has to grant 3 slots every 10 ms. In Fixed-rate Access mode, a Cyclic Assignment of 3 slots at a time is required with a Maximum Distance of 60 slots. The Requested Bandwidth is 360 slots per 1 200 ms. However, no resources are allocated in the Connect Message. That indicates to the CM that the resources for that J.112 Flow are reserved but not yet committed.

CONN

Connection_ID	37 125 <Gate ID>
Session_number	<not used>
Connection_Control_Field_Aux	
Connection_control_field2_included	1 <yes>
IPv6_add	0 <no>
Priority_included	0 <no>
Flowspec_DS_included	0 <no>
Session_binding_US_included	1 <yes>
Session_binding_DS_included	1 <yes>
Encapsulation_included	1 <yes>
DS_multiprotocol_CBD_included	0 <no>
Resource_number	0x00
Connection_Control_Field	
DS_ATM_CBD_included	0 <no>
DS_MPEG_CBD_included	1 <yes>

CONN

US_ATM_CBD_included	1 <yes>
Upstream_Channel_Number	0x1
Slot_list_included	0 <no>
Cyclic_assignment	0 <no>
Frame_Length	0 <no>
Maximum_Contention_Access_Message_Length	1 <slots>
Maximum_Reservation_Access_Message_Length	50 <slots>
Downstream_MPEG_CBD	
Downstream_Frequency	472000000 <Hz>
Program_Number	0xA437
Upstream_ATM_CBD	
Upstream_Frequency	20 000 000 <Hz>
Upstream_VPI	0x01
Upstream_VCI	0x54AC
MAC_Flag_Set	0x01
Upstream_Rate	Upstream_3.088M
Encapsulation	DirectIP (1)
Session_binding_US	
US_session_binding_control	0x1F
NIU_client_source_IP_add	MTAo
NIU_client_destination_IP_add	MTAt
NIU_client_source_port	0
NIU_client_destination_port	7 000
Upstream_transport_protocol	UDP
Session_binding_DS	
DS_session_binding_control	0x1F
INA_client_source_IP_add	MTAt
INA_client_destination_IP_add	MTAo
INA_client_source_port	0
INA_client_destination_port	7 120
Downstream_transport_protocol	UDP
Connection_control_field2	

CONN

Upstream_modulation_included	1 <yes>
Upstream_Modulation	QPSK (1)

5) Simultaneous with message No. 4, the AN initiates any required backbone reservations for the requested quality of service. The content of this message is dependent on the particular backbone algorithms in use, and is outside the scope of the present document. The backbone router sends to the AN any required notification that the reservation is successful.

6) The CM checks the resources it is required to allocate (e.g. header suppression context, Connection IDs, classifier context), and installs the classifiers. If the operation is successful it returns the Connect Response message stating the success.

CONN-RSP

Connection_ID	37 125 <Gate ID>

7) Upon receipt of the Connect Response message, the AN acknowledges receipt with a Connect Confirm message.

CONN-CNF

Connection_ID	37 125 <Gate ID>

8) Once the J.112 reservation is complete, and the backbone reservation is successful, the AN responds to the RSVP-PATH message by sending an RSVP-RESV message. The message includes the ResourceID that is assigned by the AN to this IP flow. The RSVP-RESV message is sent with the source address of MTAt and destination address of MTAo. All intermediate routers will intercept, process, and forward this as a standard RSVP-RESV message.

RSVP-RESV

Session-Object	Protocol	UDP	These fields identify the IP flow for which the reservation is being established.
	Destination Address	MTAt	
	Destination port	7 000	
Filter-Spec	Source Address	MTAo	
	Source port	7 120	
Flowspec	b	120	These fields identify the resources being reserved for this flow.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	
ResourceID		1	New Resource ID created for this reservation.

9) If the address of the previous hop in the RSVP-PATH message differs from the Source Address, then the AN is required to generate a RSVP-PATH message to reserve downstream resources at all intermediate routers. This condition would only be met if the MTAo was not immediately adjacent to the CM.

For this example, assume an intermediate router exists between MTAo and its CM, but not between MTAt and its CM.

The AN constructs a RSVP-PATH message using the Reverse Path info and sends the message to the originating MTAo. The message includes the ResourceID object.

RSVP-PATH

Session-Object	Protocol	UDP	The Session-Object and Sender-Template are faked as if the RSVP message had come from the far end.
	Destination Address	MTAo	
	Destination port	7 120	
Sender-Tspec	B	120	The Sender-Tspec came from the Reverse-Sender-Tspec in the RSVP-PATH message from MTAo. This identifies the resources that will be needed in the downstream direction (from MTAt to MTAo).
	R	12 000	
	P	12 000	
	m	120	
	M	120	
	Hdr Suppression	no	
	VAD	Off	
Forward Rspec	R	12 000	
	S	0	
ResourceID		1	New Resource ID created for this reservation.

10) MTAo, in response to the RSVP-PATH, sends RSVP-RESV to MTAt. This message is sent with "router alert" set, and all intermediate routers intercept, process, and forward this message until it reaches the AN.

RSVP-RESV

Session-Object	Protocol	UDP	The Session-Object and Sender-Template are copied from the RSVP-PATH message received.
	Destination Address	MTAo	
	Destination port	7 120	
Flowspec	b	120	These also are copied from the RSVP-PATH message, and specify the amount of resources being reserved for the flow.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	no	
	VAD	off	
	R	12 000	
	S	0	
ResourceID		1	Resource ID, copied from RSVP-PATH.

11) The CMS sends the gate co-ordination message to the AN to inform it that the resources should be committed. If the AN does not receive a COMMIT message from the MTA within Timer T2, it will abort the call.

GATE-OPEN

Transaction ID		8 096	Identifier to match this message with its response.
Gate ID		37 125	Gate-ID at AN receiving this message.
HMAC			Security checksum for this message.

12) The AN responds to the GATE-OPEN with a GATE-OPEN-ACK.

GATE-OPEN-ACK

Transaction ID		8 096	Identifier to match this message with its request.
HMAC			Security checksum for this message.

13) In response to a Modify-Connection command, which indicates the call set-up has completed (i.e. the other side has gone off-hook), MTAo sends the COMMIT message to the AN. This message is directed to the AN at a UDP port given in the RSVP-RESV Commit-Entity object. The Session-Object and Sender Template give the AN enough information to identify the "gate" and to identify which reserved resources are being committed.

COMMIT

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple must match those for the Gate ID.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Gate-ID		37 125	

14) ANo sends the event record to the Record Keeping Server that enhanced Quality of Service has been granted to this call. The format of this message is described in [J.em].

15) The AN may commit the reserved resources either using Fixed-rate Access mode or Reservation access mode. Upon reception of the COMMIT message, it needs to send the appropriate MAC-layer messages to complete the set-up of a J.112 Flow.

- For this example, it is assumed that the AN of MTAo decides to use Reservation Access mode while the AN of MTAt commits resources in Fixed-rate Access mode.
- Continuous Piggybacking is used to accommodate the CBR like characteristic of this traffic. To initiate the transmission the AN sends a Reservation ID Assignment message.

RSID-ASG

Connection_ID	37 125 <Gate ID>
Reservation_ID	0x1234
Grant_Protocol_Timeout	15 <ms>
Piggy_Back_Request_Values	
Continuous_Piggy_Back_Timeout	4 <36 ms>
GFC_11_Slots	9 <slots>
GFC_10_Slots	3 <slots>
GFC_01_Slots	1 <slots>

16) The CM sends a Reservation ID Response message showing the operation was successful.

RSID-RSP

Connection_ID	37 125 <Gate ID>
Reservation_ID	0x1234
Grant_Protocol_Timeout	15 <ms>

17) The AN at the terminating side of the call decided to provide the requested resources using Fixed-rate Access mode. To commit the resources and to initiate the transmission the AN sends a Reprovision message to the CM.

REPR

Reprovision_Control_Field	
Reprovision_Control_Aux_Field_included	0 <no>
Delete_Reservation_IDs	0 <no>
New_Downstream_IB_Frequency_included	0 <no>
New_Downstream_OOB_Frequency_included	0 <no>
New_Upstream_Frequency_included	0 <no>
New_Frame_Length_included	1 <yes>
New_Cyclical_Assignment_included	1 <yes>
New_Slot_List_included	0 <no>
New_Frame_Length	3
Number_of_Connections	1
Connection_ID	37 125 <Gate ID>
Cyclic_Assignment	
Fixedrate_Start	0x0000
Fixedrate_Dist	60
Fixedrate_Stop	0xFFFF

18) The CM sends a Link Management Response message showing the operation was successful.

LMRP

Link_Management_Msg_Number	<Reprovision Message Type Value>

19) The AN acknowledges the COMMIT with a COMMIT-ACK message.

COMMIT-ACK

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port may assist in matching the acknowledgement to the COMMIT message.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Gate-ID		37 125	

20) When the call is finished, in response to a Delete-Connection command, the MTA sends RSVP-PATH-TEAR message to the AN. For each RSVP reservation, the MTA sends a separate RSVP-PATH-TEAR message.

RSVP-PATH-TEAR

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port identify the RSVP flow.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	

21) The AN sends the notification to the Record Keeping Server that the call has ended. The format of this event message is described in [J.em].

22) The AN, upon receiving RSVP-PATH-TEAR, sends the gate co-ordination message to the address given in the GATE-SET command earlier, which in the case of NCS is the Call Agent (21b).

GATE-CLOSE

Transaction ID		73	Identifier to match this message with its response.
Gate-ID		8 095	GateID at the network element (here: CMS) receiving this message.
HMAC			Security checksum for this message.

The CMS responds with a GATE-CLOSE-ACK message (22b).

GATE-CLOSE-ACK

Transaction ID		73	Identifier to match this message with its request.
HMAC			Security checksum for this message.

23) The AN, upon receiving RSVP-PATH-TEAR, sends a Release message to the CM indicating the J.112 Flow that is to be deleted.

RELS

Number_of_Connections	1
Connection_ID	37 125 <Gate ID>

24) The CM releases the J.112 Flow and sends the Release Response to the AN.

RELS-RSP

Connection_ID	37 125 <Gate ID>

25)The AN sends the RSVP-RESV-TEAR to MTA.

RSVP-RESV-TEAR

Session-Object	Protocol	UDP	These parameters identify the IP flow that is being terminated.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	

F.2 Example Call Flow with J.112 annex B/annex C messages

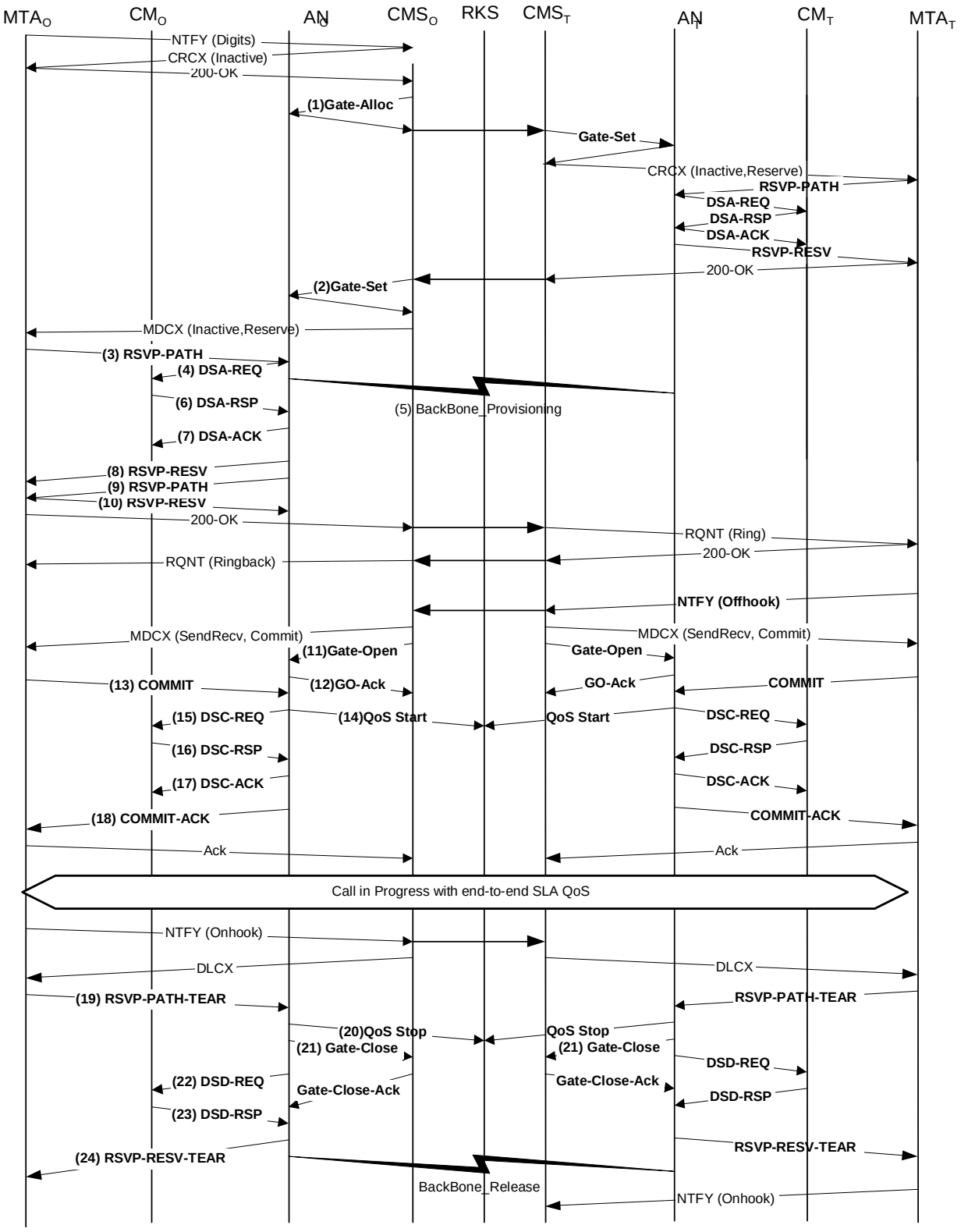


Figure 35: Basic Call Flow - NCS

1) GCo, upon receipt of signalling information from MTAo, checks the current resource consumption of MTAo by consulting ANo.

GATE-ALLOC

TransactionID		3 176	
Subscriber		MTAo	Request for total resources in use by this endpoint.
Activity-Count		4	Maximum connections allowed by client.

ANo checks current resource usage by MTAo, and responds telling the number of connections active.

GATE-ALLOC-ACK

TransactionID		3 176	
Subscriber		MTAo	Request for total resources in use by this endpoint.
Gate-ID		3 7125	Identifier for allocated Gate.
Activity Count		3	Total connections established by this client.

2) GCo, upon further signalling exchanges, gives ANo authorization to admit the new connection.

GATE-SET

Transaction ID		3 177	Unique Transaction ID for this message exchange.
Subscriber		MTAo	Request for total resources in use by this client.
Gate-ID		37 125	Identifier for allocated Gate.
Remote-Gate-Info	Address	CMSo	Information needed to perform gate co-ordination. Note that CMSo has given itself as the entity for exchanging gate co-ordination messages. Flag value indicates that the AN should not send a Gate-Open message when it receives a COMMIT from the MTA, but still expect to receive a Gate-Open message from CMS_O.
	Port	2 052	
	Remote Gate-ID	8 095	
	Security Key	<key>	
	Flag	No-gate-open	
Event-Generation-Info	RKS-Addr	RKS	Address of Record Keeping Server.
	RKS-Port	3 288	Port on Record Keeping Server.
	Billing Correlation ID	<id>	Opaque data that will be passed to RKS when resources are committed.

GATE-SET

Gate-Spec	Direction	up	
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAo	
	Destination Address	MTAt	
	Source port	0	
	Destination port	7 000	
	DSCP	6	Packet Type value for upstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

GATE-SET

Gate-Spec	Direction	down	
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAt	
	Destination Address	MTAo	
	Source port	0	
	Destination port	7 120	
	DSCP	9	Packet Type value for downstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

ANo responds to the Gate Setup command with an acknowledgement.

GATE-SET-ACK

TransactionID		3 177	
Subscriber		MTAo	Request for total resources in use by this client.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		4	Total connections established by this client.

3) MTAo, upon receiving a Modify-Connection command, sends an RSVP-PATH message, addressed to MTAt, but with the Router-Alert bit set in the IP header. Intermediate routers in the home LAN intercept, process, and forward this message as a normal RSVP-PATH.

RSVP-PATH

Session-Object	Protocol	UDP	The parameters identify the RSVP session, match the authorization previously sent by the GateController, and are also used for QoS classifiers.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Sender-Tspec	b	120	These are the negotiated traffic parameters actually being requested for this call. The AN calculates the actual upstream QoS parameters using these Tspec and Rspec parameters. This is a standard RSVP object, which will be interpreted by all intermediate routers in the path between the MTA and AN.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	40	
	VAD	off	
Forward Rspec	R	12 000	
	S	0	
Reverse-Session	Protocol	UDP	New RSVP objects that provides the AN with sufficient information to calculate downstream traffic parameters and to generate an RSVP-PATH message for the downstream flow.
	Destination Addr	MTAo	
	Destination port	7 120	
Reverse-Sender Templ	Source Address	MTAt	
	Source port	0	

RSVP-PATH

Reverse-Sender-Tspec	b	120	Negotiated traffic parameters actually being requested for this call. The AN calculates the actual downstream QoS parameters using these Tspec and Rspec parameters. This is a new RSVP object, which will be ignored by intermediate routers.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	0	
	VAD	off	
Reverse-Rspec	R	12 000	
	S	0	
Gate ID		37 125	

4) The AN uses the RSVP-PATH message and calculates the QoS parameters for the J.112 link. The AN sends the following DSA-REQ to the CM. This message is used to establish both upstream and downstream parameters. The Upstream Unsolicited Grant Size was calculated as 120 (from Tspec) plus 18 (Ethernet overhead) minus 40 (Header Suppression amount) plus 13 (J.112 overhead). Header suppression, being specified as length 40 in the RSVP-PATH, indicates the 42 bytes of Ethernet/IP/UDP header. Contents of the suppressed header is taken from the RSVP packet.

DSA-REQ

TransactionID		1
UpstreamServiceFlow	ServiceFlowIdentifier	1 001
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	ServiceFlowScheduling	UGS (6)
	Request/TransmissionPolicy	0x00000017
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111
DownstreamServiceFlow	ServiceFlowIdentifier	2 001
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	TrafficPriority	5
	MaximumSustainedRate	12 000

DSA-REQ

UpstreamPacketClassification	ServiceFlowIdentifier	1 001
	PacketClassifierIdentifier	3 001
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAo
	IPSourcePort	7 120
	IPDestinationAddress	MTAt
	IPDestinationPort	7 000
	IPProtocol	UDP (17)
DownstreamPacketClassification	ServiceFlowIdentifier	2 001
	PacketClassifierIdentifier	3 002
	ClassifierPriority	150
	ClassifierActivationState	Inactive(0)
	IPSourceAddress	MTAt
	IPSourcePort	7 000
	IPDestinationAddress	MTAo
	IPDestinationPort	7 120
	IPProtocol	UDP (17)
PayloadHeaderSuppression	ClassifierIdentifier	3 001
	ServiceFlowIdentifier	1 001
	HeaderSuppressionIndex	1
	HeaderSuppressionField	<42bytes>
	HeaderSuppressionMask	<42bits>
	HeaderSuppressionSize	42
	HeaderSuppressionVerify	Verify (0)
HMAC		

5) Simultaneous with message No. 2, the AN initiates any required backbone reservations for the requested quality of service. The content of this message is dependent on the particular backbone algorithms in use, and is outside the scope of the present document. The backbone router sends to the AN any required notification that the reservation is successful.

6) The CM checks the resources it is required to allocate (e.g. header suppression table space, Service Flow IDs, classifier table space, local network bandwidth), and installs the classifiers. If the operation is successful it returns the DSA-RSP message stating the success.

DSA-RSP

TransactionID		1
ConfirmationCode		Success (0)
HMAC		

7) Upon receipt of the DSA-RSP, the AN acknowledges receipt with a DSA-ACK message.

DSA-ACK

TransactionID		1
ConfirmationCode		Success (0)
HMAC		

8) Once the J.112 reservation is complete, and the backbone reservation is successful, the AN responds to the RSVP-PATH message by sending an RSVP-RESV message. The message includes the ResourceID that is assigned by the AN to this connection. The RSVP-RESV message is sent with the source address of MTAT and destination address of MTAo. All intermediate routers will intercept, process, and forward this as a standard RSVP-RESV message.

RSVP-RESV

Session-Object	Protocol	UDP	These fields identify the IP flow for which the reservation is being established.
	Destination Address	MTAt	
	Destination port	7 000	
Filter-Spec	Source Address	MTAo	
	Source port	7 120	
Flowspec	b	120	These fields identify the resources being reserved for this flow.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	
ResourceID		1	New Resource ID created for this reservation.

9) If the address of the previous hop differs from the Source Address, then the AN is required to generate a RSVP-PATH message to reserve downstream resources at all intermediate routers. This condition would only be met it the MTA was not immediately adjacent to the CM.

- For this example, assume an intermediate router exists between MTA-o and its CM, but not between MTA-t and its CM.
- The AN constructs RSVP-PATH message using the Reverse Path info it received from the RSVP-PATH message and sends the message to the originating MTA. The message includes the ResourceID object.

RSVP-PATH

Session-Object	Protocol	UDP	The Session-Object and Sender-Template are faked as if the RSVP message had come from the far end.
	Destination Address	MTAo	
	Destination port	7 120	
Sender-Tspec	b	120	The Sender-Tspec came from the Reverse-Sender-Tspec in the RSVP-PATH message from MTAo. This identifies the resources that will be needed in the downstream direction (from MTAt to MTAo).
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	40	
	VAD	Off	
Forward Rspec	R	12 000	
	S	0	
ResourceID		1	New Resource ID created for this reservation.

10) MTAo, in response to the RSVP-PATH (7), sends RSVP-RESV to MTAt. This message is sent with "router alert" set, and all intermediate routers intercept, process, and forward this message until it reaches the AN.

RSVP-RESV

Session-Object	Protocol	UDP	The Session-Object and Sender-Template are copied from the RSVP-PATH message received.
	Destination Address	MTAo	
	Destination port	7 120	
Flowspec	b	120	These also are copied from the RSVP-PATH message, and specify the amount of resources being reserved for the flow.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	40	
	VAD	Off	
	R	12 000	
	S	0	
ResourceID		1	Resource ID, copied from RSP-PATH.

11) The CMS sends the gate co-ordination message to the AN to inform it that the resources should be committed. If the AN does not receive a COMMIT message from the MTA within Timer T2, it will abort the connection.

GATE-OPEN

Transaction ID		8 096	Identifier to match this message with its response.
Gate ID		37 125	Gate-ID at remote AN.
HMAC			Security checksum for this message.

12) The AN responds to the GATE-OPEN with:

GATE-OPEN-ACK

Transaction ID		8 096	Identifier to match this message with its response.
HMAC			Security checksum for this message.

13) In response to Modify-Connection command, which indicate the call has completed (i.e. the other side has gone off-hook), MTAo sends the COMMIT message to the AN. This message is directed to the AN at a UDP port given in the RSVP-RESV Commit-Entity object. The Session-Object and Sender Template give the AN enough information to identify the "gate" and to identify which reserved resources are being committed.

COMMIT

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple must match those for the Gate ID.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Gate-ID		37 125	

14) ANo sends the event record to the Record Keeping Server that an enhanced Quality-of-Service connection has been granted to this call.

QoS-START

Header	Timestamp	<time>	Time of the event being recorded.
	Billing Correlation ID	<string>	Correlation ID given in Gate-Set.
QoS Descriptor	Type	UGS	Description of the QoS provided for this connection.
	Grant interval	10 ms	
	Grant Jitter	2 ms	
	Grant/Interval	1	
	Grant Size	111	

QoS-START

MTA Port	Port	7 120	

15) The AN resolves which reservation is to be activated, and sends a DSC-REQ to the CM to activate the flow.

DSC-REQ

TransactionID		2
UpstreamServiceFlow	ServiceFlowIdentifier	1 001
	QoSParameterSetType	Admitted + Activated (6)
	TimeOutActive	10
	ServiceFlowScheduling	UGS (6)
	Request/TransmissionPolicy	0x00000017
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111
DownstreamServiceFlow	ServiceFlowIdentifier	2 001
	QoSParameterSetType	Admitted + Activated (6)
	TimeOutActive	10
	TrafficPriority	5
	MaximumSustainedRate	12 000
UpstreamPacketClassification	ServiceFlowIdentifier	1 001
	PacketClassifierIdentifier	3 001
	ClassifierChangeAction	Replace (1)
	ClassifierPriority	150
	ClassifierActivationState	Active (1)
	IPSourceAddress	MTAo
	IPSourcePort	7 120
	IPDestinationAddress	MTAt
	IPDestinationPort	7 000
	IPProtocol	UDP (17)
DownstreamPacketClassification	ServiceFlowIdentifier	2 001
	PacketClassifierIdentifier	3 002
	ClassifierChangeAction	Replace (1)

DSC-REQ

	ClassifierPriority	150
	ClassifierActivationState	Active(1)
	IPSourceAddress	MTAt
	IPSourcePort	7 000
	IPDestinationAddress	MTAo
	IPDestinationPort	7 124
	IPProtocol	UDP (17)
HMAC		

16) The CM sends a DSC-RSP message showing the operation was successful.

DSC-RSP

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

17) The AN sends DSC-ACK message to indicate that the DSC-RSP has been received and agreed.

DSC-ACK

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

18) The AN acknowledges the COMMIT with:

COMMIT-ACK

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port may assist in matching the acknowledgement to the COMMIT message.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Gate-ID		37 125	

19) When the call is finished, in response to a Delete-Connection command, the MTA sends RSVP-PATH-TEAR message to the AN. For each RSVP reservation, the MTA sends a separate RSVP-PATH-TEAR message.

RSVP-PATH-TEAR

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port identify the RSVP flow.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	

20) The AN sends the notification to the Record Keeping Server that the call has ended.

QoS-Stop

TimeStamp		<time>	The time of the event being recorded.
Header	Time Stamp	<time>	Time of the event being recorded.
	Billing Correlation ID	<string>	Correlation ID from Gate-Set message.
SF-ID	SF-ID	1 001	Service Flow Identifier.

21) The AN, upon receiving RSVP-PATH-TEAR, sends the gate co-ordination message to the address given in the GATE-SET command earlier, which in the case of NCS is the Call Agent.

GATE-CLOSE

Transaction ID		73	Identifier to match this message with its response.
Gate-ID		8 095	This identifies the GateID at the remote AN.
HMAC			Security checksum for this message.

The CMS responds with:

GATE-CLOSE-ACK

Transaction ID		73	Identifier to match this message with its response.
HMAC			Security checksum for this message.

22) The AN, upon receiving RSVP-PATH-TEAR, sends a DSD-REQ to the CM indicating the Service Flow ID that is to be deleted.

DSD-REQ

TransactionID		3
ServiceFlowID		1 001
HMAC		

DSD-REQ

TransactionID		4
ServiceFlowID		2 001
HMAC		

23) The CM deletes the Service Flow ID and sends the response to the AN.

DSD-RSP

TransactionID		3
ServiceFlowID		1 001
ConfirmationCode		Success (0)
HMAC		

DSD-RSP

TransactionID		4
ServiceFlowID		2 001
ConfirmationCode		Success (0)
HMAC		

24) The AN sends the RSVP-RESV-TEAR to MTA.

RSVP-RESV-TEAR

Session-Object	Protocol	UDP	These parameters identify the IP flow that is being terminated.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	

Annex G (informative): Sample Protocol Message Exchanges for Mid-Call Codec Change

The codec change is achieved by the MTAs transmitting a new RSVP-PATH message subsequent to the call signalling exchange between them to determine what new codec is being used. The new FlowSpec for the call is described in the RSVP-PATH, and must fit within the Authorized Envelope specified in the Gate-Set message that was exchanged between the GCs and the ANs earlier for this Gate. The RSVP-PATH includes the same GateID that was previously used for this call. Observe that the initial INVITE to establish the call should have included the codecs in the SDP to ensure that the Authorized Envelope is large enough to accommodate the codec change. The RSVP-PATH message includes the FlowSpec for both the codecs as explained below.

G.1 Example Call Flow with J.112 annex A messages

For further study.

G.2 Example Call Flow with J.112 annex B/annex C messages

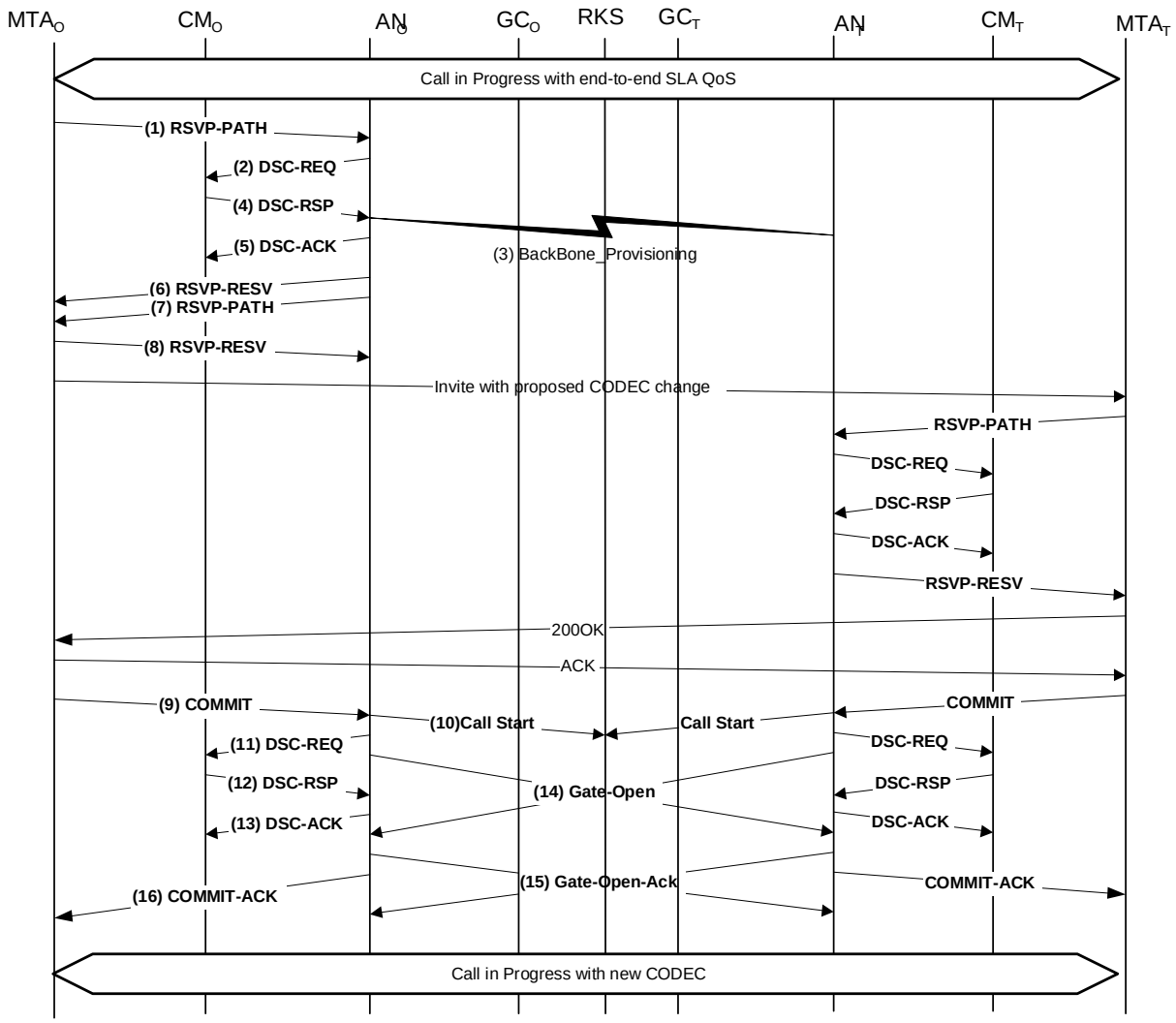


Figure 36: QoS Signalling for Codec Change

1) MTAo and MTAt are assumed to have a G.728 (20 ms packets, each 80 bytes) call active when MTAo decides, for whatever reason, that a CODEC change is needed to G.711 (10 ms packets, each 120 bytes). After an initial signalling exchange that determines MTAt is capable of handling the desired new CODEC, MTAo sends an RSVP-PATH message addressed to MTAt, but with the Router-Alert bit set in the IP header. Intermediate routers in the home LAN intercept, process, and forward this message as a normal RSVP-PATH, understanding only the least-upper-bound set of traffic parameters given in the Sender-Tspec.

RSVP-PATH

Session-Object	Protocol	UDP	The parameters identify the RSVP session, match the authorization previously sent by the GateController, and are also used for QoS classifiers.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Sender-Tspec	b	120	These give the Least-Upper-Bound for all of the individual traffic parameters for the two separate possible flows. This is a standard RSVP object, which will be interpreted by all intermediate routers in the path between the MTA and AN.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	40	
	VAD	off	
Gate-ID		37 125	Identity of gate that authorizes this request.
Component Tspec	b	120	These are the negotiated traffic parameters for the new CODEC being requested for this call. The AN calculates the actual upstream QoS parameters using these Tspec and Rspec parameters.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	40	
	VAD	off	
Forward Rspec	R	12 000	Rspec that corresponds to the immediately preceding Component Tspec.
	S	0	
Reverse-Session	Protocol	UDP	New RSVP objects that provides the AN with sufficient information to calculate downstream traffic parameters and to generate an RSVP-PATH message for the downstream flow.
	Destination Addr	MTAo	
	Destination port	7 120	
Reverse-Sender Templ	Source Address	MTAt	
	Source port	7 000	

RSVP-PATH

Reverse-Sender-Tspec	b	120	Negotiated traffic parameters for the new CODEC being requested for this call. The AN calculates the actual downstream QoS parameters using these Tspec and Rspec parameters. This is a new RSVP object, which will be ignored by intermediate routers.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	0	
	VAD	off	
Reverse-Rspec	R	12 000	
	S	0	
Component Tspec	b	80	These are the negotiated traffic parameters for the old CODEC currently being used for this call. The AN calculates the actual upstream QoS parameters using these Tspec and Rspec parameters.
	r	4 000	
	p	4 000	
	m	80	
	M	80	
	Hdr Suppression	40	
	VAD	off	
Forward Rspec	R	4 000	Rspec that corresponds to the immediately preceding Component Tspec.
	S	0	
Reverse-Session	Protocol	UDP	New RSVP objects that provides the AN with sufficient information to calculate downstream traffic parameters and to generate an RSVP-PATH message for the downstream flow.
	Destination Addr	MTAo	
	Destination port	7 120	
Reverse-Sender Templ	Source Address	MTAt	
	Source port	7 000	
Reverse-Sender-Tspec	b	80	Negotiated traffic parameters for the old CODEC currently being used for this call. The AN calculates the actual downstream QoS parameters using these Tspec and Rspec parameters. This is a new RSVP object, which will be ignored by intermediate routers.
	r	4 000	
	p	4 000	
	m	80	
	M	80	
	Hdr Suppression	0	
	VAD	off	
Reverse-Rspec	R	4 000	
	S	0	

2) The AN uses the RSVP-PATH message and calculates the new QoS parameters for the J.112 link. Since the G.728 stream fits completely within an allocation for G.711, there is no need for a separate Service Flow; therefore the existing Service Flows are modified to increase the admitted bandwidth. The AN sends the following DSC-REQ to the CM. This message is used to establish both upstream and downstream parameters. The Upstream Unsolicited Grant Size was calculated as 120 (from Tspec) plus 18 (Ethernet overhead) minus 40 (Header Suppression amount) plus 13 (J.112 overhead). Header suppression, being specified as length 40 in the RSVP-PATH, indicates the 42 bytes of Ethernet/IP/UDP. Contents of the suppressed header is taken from the RSVP packet.

DSC-REQ

TransactionID		1
UpstreamServiceFlow	ServiceFlowIdentifier	1 001
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	ServiceFlowScheduling	UGS (6)
	Request/TransmissionPolicy	0x00000017
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111
UpstreamServiceFlow	ServiceFlowIdentifier	1 001
	QoSParameterSetType	Active (4)
	TimeOutActive	10
	ServiceFlowScheduling	UGS (6)
	NominalGrantInterval	20 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	71
DownstreamServiceFlow	ServiceFlowIdentifier	2 001
	QoSParameterSetType	Admitted (2)
	TimeOutActive	10
	TrafficPriority	5
	MaximumSustainedRate	12 000
DownstreamServiceFlow	ServiceFlowIdentifier	2 001
	QoSParameterSetType	Active(4)
	TimeOutActive	10
	TrafficPriority	5
	MaximumSustainedRate	4 000
HMAC		

3) Simultaneous with message No. 2, the AN initiates any required backbone reservations for the requested quality of service. The content of this message is dependent on the particular backbone algorithms in use, and is outside the scope of the present document. The backbone router sends to the AN any required notification that the reservation is successful.

4) The CM checks the additional resources it is required to allocate (e.g. local network bandwidth). If the operation is successful it returns the DSC-RSP message stating the success.

DSC-RSP

TransactionID		1
ConfirmationCode		Success (0)
HMAC		

5) Upon receipt of the DSC-RSP, the AN acknowledges receipt with a DSA-ACK message.

DSC-ACK

TransactionID		1
ConfirmationCode		Success (0)
HMAC		

6) Once the J.112 reservation is complete, and the backbone reservation is successful, the AN responds to the RSVP-PATH message by sending an RSVP-RESV message. The message includes the Least-Upper-Bound of the two Sender-Tspecs, causing the intermediate routers to allocate resources sufficient to cover either flow. The RSVP-RESV message is sent with the source address of MTAT and destination address of MTAo. All intermediate routers will intercept, process, and forward this as a standard RSVP-RESV message.

RSVP-RESV

Session-Object	Protocol	UDP	These fields identify the IP flow for which the reservation is being established.
	Destination Address	MTAt	
	Destination port	7 000	
Filter-Spec	Source Address	MTAo	
	Source port	7 120	
Flowspec	b	120	These fields identify the resources being reserved for this flow. These values are the Least-Upper-Bound of the two Tspecs given in the RSVP-PATH.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	
ResourceID		1	Resource ID previously created for this reservation.

7) If the address of the previous hop differs from the Source Address, then the AN is required to generate a RSVP-PATH message to reserve downstream resources at all intermediate routers. This flag would only be set if the MTA was not immediately adjacent to the CM.

The AN constructs RSVP-PATH message using the Reverse Path info it received from the RSVP-PATH message and sends the message to the originating MTA. The message includes the ResourceID object.

RSVP-PATH

Session-Object	Protocol	UDP	The Session-Object and Sender-Template are faked as if the RSVP message had come from the far end.
	Destination Address	MTAo	
	Destination port	7 120	
Sender Templ	Source Address	MTAt	
	Source port	7 000	
Sender-Tspec	b	120	The Sender-Tspec came from the Reverse-Sender-Tspec in the RSVP-PATH message from MTAo. This identifies the resources that will be needed in the downstream direction (from MTAt to MTAo). This Tspec is the Least-Upper-Bound of the two individual Tspecs sent to the AN, causing all intermediate routers to allocate enough resources for either flow.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	40	
	VAD	off	
Forward Rspec	R	12 000	
	S	0	
ResourceID		1	Resource ID previously created for this reservation.

8) MTAo, in response to the RSVP-PATH (7), sends RSVP-RESV to MTAt. This message is sent with "router alert" set, and all intermediate routers intercept, process, and forward this message until it reaches the AN.

RSVP-RESV

Session-Object	Protocol	UDP	The Session-Object and Sender-Template are copied from the RSVP-PATH message received.
	Destination Address	MTAo	
	Destination port	7 120	
Filter-Spec	Source Address	MTAt	
	Source port	7 000	
Flowspec	b	120	These also are copied from the RSVP-PATH message, and specify the amount of resources being reserved for the flow.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	40	
	VAD	off	
	R	12 000	
	S	0	
ResourceID		1	Resource ID, copied from RSP-PATH.

9) In response to end-to-end signalling messages that indicate the resource reservation was successful at both ends, MTAo sends the COMMIT message to the AN. This message is directed to the AN at a UDP port determined via call signalling.

The Session-Object and Sender Template give the AN information to verify the Gate-ID and to identify which reserved resources are being committed.

COMMIT

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple must match those for the Gate-ID.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Gate-ID		37 125	

10) ANo sends the event record to the Record Keeping Server that a Commit has been received on this call. This message is only a sample of what might be included in a QoS-Start message.

QoS-START

Header	Timestamp	<time>	Time of the event being recorded.
	Billing Correlation ID	<string>	Correlation ID given in Gate-Set.
QoS Descriptor	Type	UGS	Description of the QoS provided for this connection.
	Grant interval	10 ms	
	Grant Jitter	2 ms	
	Grant/Interval	1	
	Grant Size	111	
MTA Port	Port	7 120	

11) The AN resolves which reservation is to be activated, and sends a DSC-REQ to the CM to activate the flow.

DSC-REQ

TransactionID		2
UpstreamServiceFlow	ServiceFlowIdentifier	1 001
	QoSParameterSetType	Admitted + Activated (6)
	TimeOutActive	10
	ServiceFlowScheduling	UGS (6)
	Request/TransmissionPolicy	0x00000017
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111
DownstreamServiceFlow	ServiceFlowIdentifier	2 001
	QoSParameterSetType	Admitted + Activated (6)
	TimeOutActive	10
	TrafficPriority	5
	MaximumSustainedRate	12 000
HMAC		

12) The CM sends a DSC-RSP message showing the operation was successful.

DSC-RSP

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

13) The AN sends DSC-ACK message to indicate that the DSC-RSP has been received and agreed.

DSC-ACK

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

14) The AN sends the gate co-ordination message to the remote AN to inform it that the resources at this end have been committed.

GATE-OPEN

Transaction ID		74	Identifier to match this message with its response.
Gate ID		1 273	Gate-ID at remote AN.
Tspec	b	120	These are the committed traffic parameters actually being utilized in the MTAo to MTAt direction.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
Reverse-Tspec	b	120	These are the expected traffic parameters being utilized in the MTAt to MTAo direction.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
HMAC			Security checksum for this message.

15) The remote AN responds to the GATE-OPEN with:

GATE-OPEN-ACK

Transaction ID		74	Identifier to match this message with its response.
HMAC			Security checksum for this message.

16) The AN acknowledges the COMMIT with:

COMMIT-ACK

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port may assist in matching the acknowledgement to the COMMIT message.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Gate-ID		37 125	

Annex H (informative): Sample Protocol Message Exchanges for Call Hold

Putting a call on Hold at an MTA is performed by sending an INVITE to the MTA with SDP parameters being zero. This results in the MTA sending a COMMIT message with a Flow Spec of 0. Also included is a Resource ID. This enables the AN to hold on to the admitted resources, but will now commit zero resources to the flow. This is performed with a MAC message exchange at the J.112 MAC level.

H.1 Example Call Flow with J.112 annex A messages

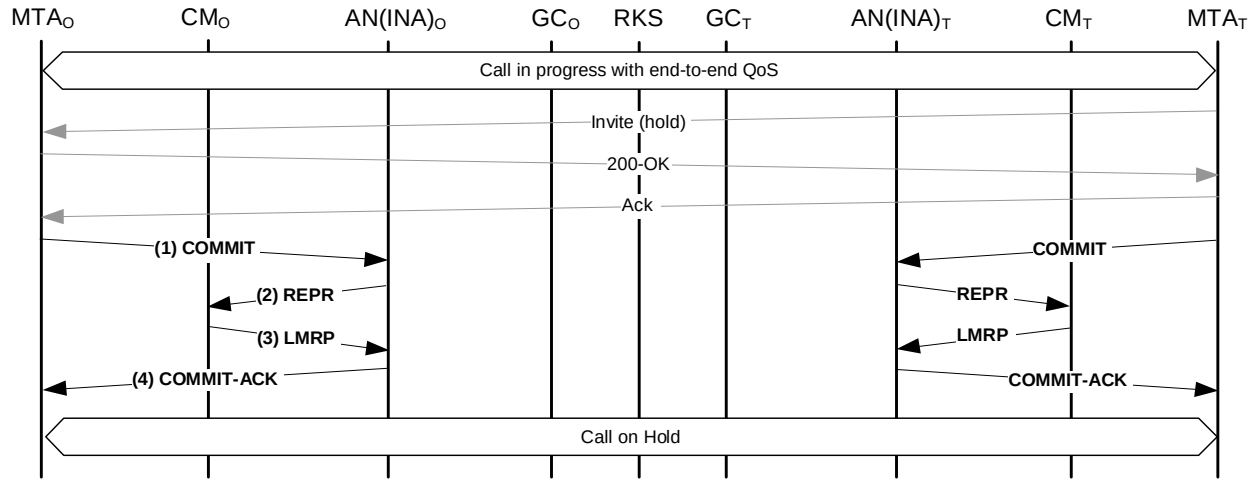


Figure 37: QoS Signalling for Call Hold

1) When MTAt decides that the current call is to be placed on hold it sends an INVITE message to MTAo. After further call signalling exchange MTAo sends a COMMIT message with an empty Flowspec.

COMMIT

Session-Object	Protocol	UDP	The Session-Object and Sender Template verify the gate identity.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Gate-ID		37 125	
Flowspec	b	0	A Flowspec is an optional object for a COMMIT message and indicates that the committed resources are for some amount different from the reserved resources; for Call Hold the committed upstream resources are changed to zero.
	r	0	
	p	0	
	m	0	
	M	0	
	R	0	
	S	0	

COMMIT

Reverse-Flowspec	b	0	A Flowspec is an optional object for a COMMIT message and indicates that the committed resources are for some amount different from the reserved resources; for Call Hold the committed downstream resources are changed to zero.
	r	0	
	p	0	
	m	0	
	M	0	
	R	0	
	S	0	

2) ANo sends a Reprovision message to CMt.

REPR

Reprovision_Control_Field	
Reprovision_Control_Aux_Field_included	0 <no>
Delete_Reservation_Ids	1 <yes>
New_Downstream_IB_Frequency_included	0 <no>
New_Downstream_OOB_Frequency_included	0 <no>
New_Upstream_Frequency_included	0 <no>
New_Frame_Length_included	1 <yes>
New_Cyclical_Assignment_included	1 <yes>
New_Slot_List_included	0 <no>
New_Frame_Length	0
Number_of_Connections	1
Connection_ID	37 125 <Gate ID>
Cyclic_Assignment	
Fixedrate_Start	0xFFFF
Fixedrate_Dist	0
Fixedrate_Stop	0xFFFF

3) CMt sends a Link Management Response message showing the operation was successful.

LMRP

Link_Management_Msg_Number	<Reprovision Message Type Value>

4) ANo acknowledges the COMMIT with a COMMIT-ACK message.

COMMIT-ACK

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port may assist in matching the acknowledgement to the COMMIT message.
	Destination Address	MTAt	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 120	
Gate-ID		37 125	

H.2 Example Call Flow with J.112 annex B/annex C messages

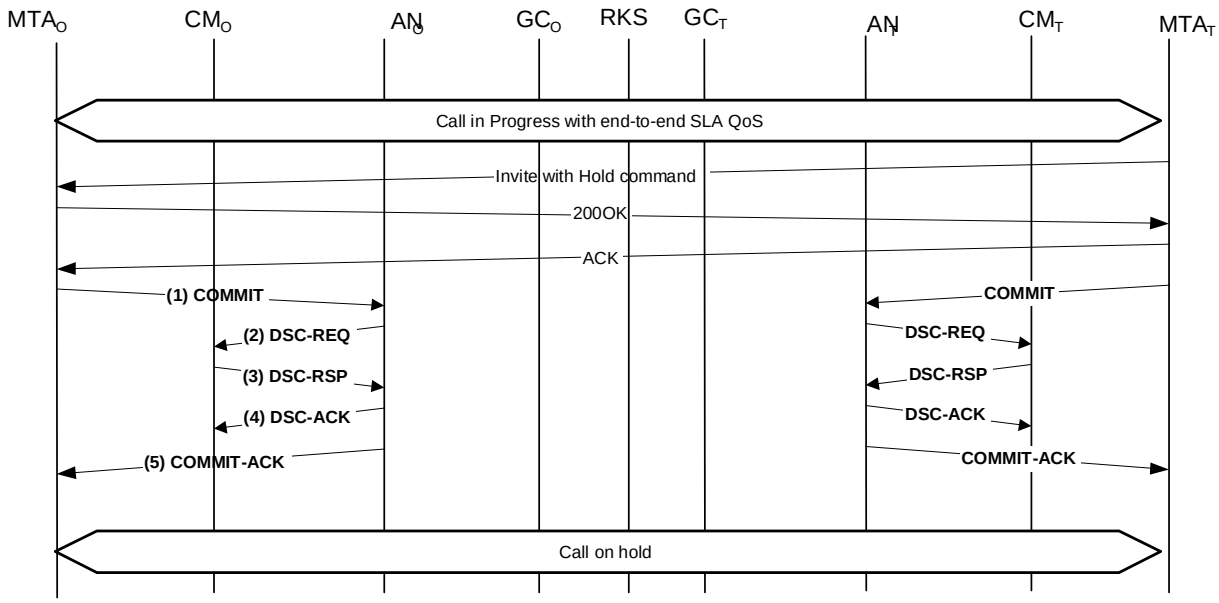


Figure 38: QoS Signalling for Call Hold

1) When MTA decides that the current call is to be placed in hold it sends a commit message with a bandwidth of zero. The MTA cannot change the active session ID during a call hold commit message.

COMMIT

Session-Object	Protocol	UDP	The Session-Object and Sender-Template verify the gate identity.
	Destination Address	MTAo	
	Destination port	7 120	
Sender Templ	Source Address	MTAt	
	Source port	7 000	
Gate-ID		37 125	
Flowspec	b	0	These are optional in a COMMIT message, and indicate the activation is for some amount different from the reservation; in this case the desired upstream activation is null.
	r	0	
	p	0	
	m	0	
	M	0	
	R	0	
	S	0	
Reverse-Flowspec	b	0	These are optional in a COMMIT message, and indicate the activation is for some amount different from the reservation; in this case the desired downstream activation is null.
	r	0	
	p	0	
	m	0	
	M	0	
	R	0	
	S	0	

2) The AN sends the CM a DSC message to deactivate the Service Flow and to deactivate the classifiers.

DSC-REQ

TransactionID		1
UpstreamServiceFlow	ServiceFlowIdentifier	1 001
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	ServiceFlowScheduling	UGS (6)
	Request/TransmissionPolicy	0x00000017
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111
DownstreamServiceFlow	ServiceFlowIdentifier	2 001
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	TrafficPriority	5
	MaximumSustainedRate	12 000
UpstreamPacketClassification	ServiceFlowIdentifier	1 001
	PacketClassifierIdentifier	3 001
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAo
	IPSourcePort	7 120
	IPDestinationAddress	MTAt
	IPDestinationPort	7 000
	IPProtocol	UDP (17)

DSC-REQ

DownstreamPacketClassification	ServiceFlowIdentifier	2 001
	PacketClassifierIdentifier	3 002
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAt
	IPSourcePort	7 000
	IPDestinationAddress	MTAo
	IPDestinationPort	7 124
	IPProtocol	UDP (17)
HMAC		

3) The CM sends a DSC-RSP message showing the operation was successful.

DSC-RSP

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

4) The AN sends DSC-ACK message to indicate that the DSC-RSP has been received and agreed.

DSC-ACK

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

5) The AN sends COMMIT_ACK message.

COMMIT-ACK

Session-Object	Protocol	UDP	The Session-Object and Sender-Template verify the gate identity.
	Destination Address	MTAo	
	Destination port	7 120	
Sender Templ	Source Address	MTAt	
	Source port	7 000	
Gate-ID		37 125	

Annex I (informative): Sample Protocol Message Exchanges for Call Waiting

I.1 Example Call Flow with J.112 annex A messages

For further study.

I.2 Example Call Flow with J.112 annex B/annex C messages

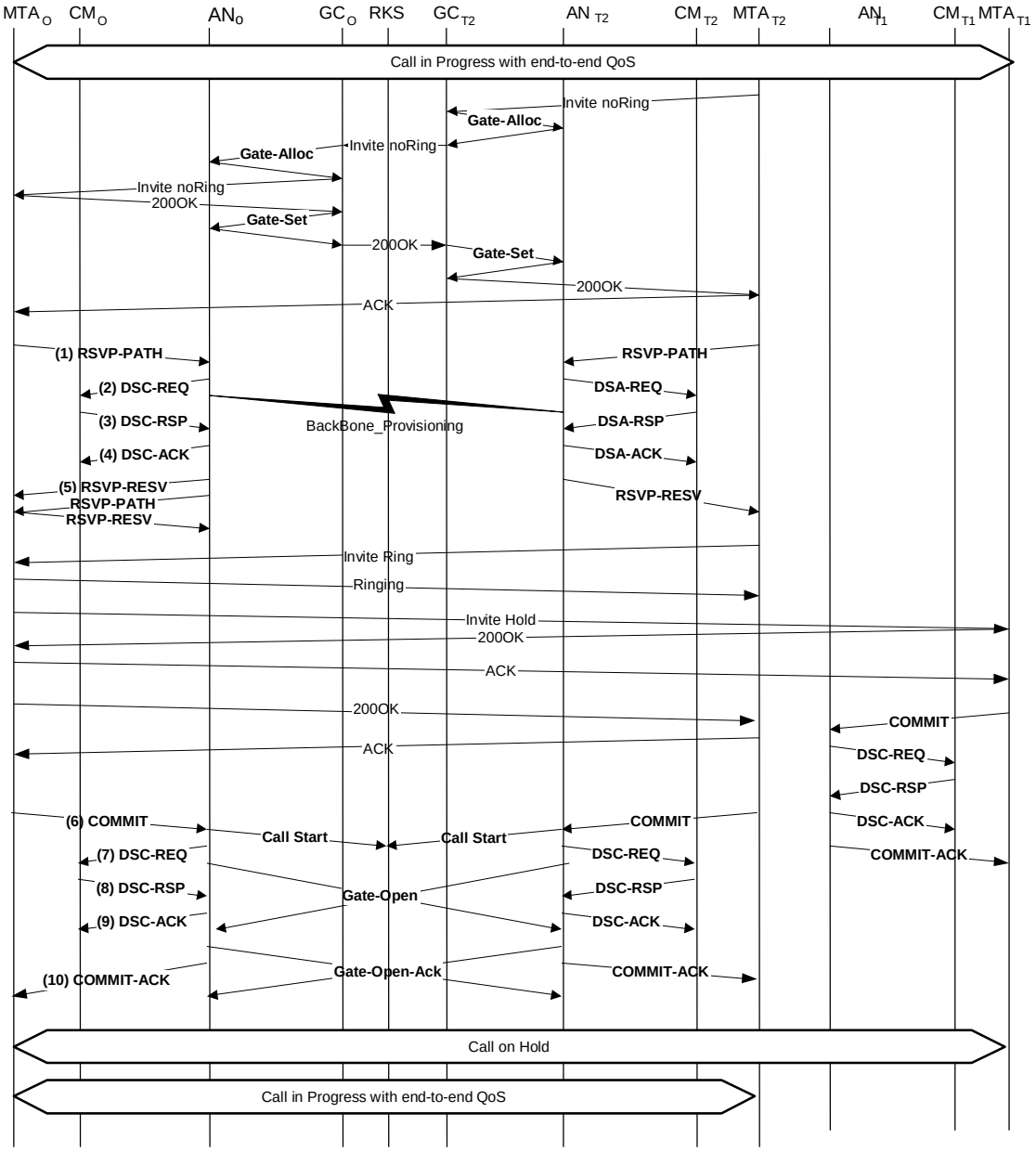


Figure 39: QoS Signalling for Call Waiting

1) MTAo is connected to MTAt1, and receives an incoming call from MTAt2. For this example, assume the call from MTAt1 had been using UDP port 7 120, and assigned ResourceID 472. Upon receipt of the call signalling information, MTAo sends an RSVP-PATH message, addressed to MTAt2, but with the Router-Alert bit set in the IP header. Intermediate routers in the home LAN intercept, process, and forward this message as a normal RSVP-PATH, thinking it is a separate flow and allocating separate resources for it.

RSVP-PATH

Session-Object	Protocol	UDP	The parameters form the classifier, matching the authorization previously sent by the GateController.
	Destination Address	MTAt2	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 122	
Sender-Tspec	b	120	These are the negotiated traffic parameters actually being requested for this call. The AN calculates the actual upstream QoS parameters using these Tspec and Rspec parameters. This is a standard RSVP object, which will be interpreted by all intermediate routers in the path between the MTA and AN.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	40	
	VAD	off	
Forward Rspec	R	12 000	
	S	0	
Reverse-Session	Protocol	UDP	New RSVP objects that provides the AN with sufficient information to calculate downstream traffic parameters and to generate an RSVP-PATH message for the downstream flow.
	Destination Addr	MTAo	
	Destination port	7 122	
Reverse-Sender Templ	Source Address	MTAt	
	Source port	0	
Reverse-Sender-Tspec	b	120	Negotiated traffic parameters actually being requested for this call. The AN calculates the actual downstream QoS parameters using these Tspec and Rspec parameters. This is a new RSVP object, which will be ignored by intermediate routers.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	Hdr Suppression	0	
	VAD	off	
Reverse-Rspec	R	12 000	
	S	0	

RSVP-PATH

ResourceID		472	Resource ID assigned for existing call.
Gate-ID		37 126	Gate-ID for this new call, take resources from old.

2) The AN uses the RSVP-PATH message and calculates the QoS parameters for the J.112 link. For this example, assume the previous call was also G.711, and therefore the bandwidth requirements are identical. Thus the existing ServiceFlow can be used for both packet streams. The AN sends the following DSC-REQ to the CM, which establishes the new classifiers. Header suppression, being specified as length 40 in the RSVP-PATH, indicates the 42 bytes of Ethernet/IP/UDP header. Contents of the suppressed header is taken from the RSVP packet.

DSC-REQ

TransactionID		1
UpstreamPacketClassification	ServiceFlowIdentifier	1 001
	PacketClassifierIdentifier	3 003
	ClassifierChangeAction	Add (0)
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAo
	IPSourcePort	7 122
	IPDestinationAddress	MTAt2
	IPDestinationPort	7 000
	IPProtocol	UDP (17)
DownstreamPacketClassification	ServiceFlowIdentifier	2 001
	PacketClassifierIdentifier	3 004
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAt2
	IPDestinationAddress	MTAo
	IPDestinationPort	7 122
	IPProtocol	UDP (17)

DSC-REQ

PayloadHeaderSuppression	ClassifierIdentifier	3 003
	ServiceFlowIdentifier	1 001
	HeaderSuppressionIndex	1
	HeaderSuppressionField	<42bytes>
	HeaderSuppressionMask	<42bits>
	HeaderSuppressionSize	42
	HeaderSuppressionVerify	Verify (0)
HMAC		

3) The CM checks the resources it is required to allocate (e.g. header suppression table space, Service Flow IDs, classifier table space, local network bandwidth), and installs the classifiers. If the operation is successful it returns the DSC-RSP message stating the success.

DSC-RSP

TransactionID		1
ConfirmationCode		Success (0)
HMAC		

4) Upon receipt of the DSC-RSP, the AN acknowledges receipt with a DSC-ACK message.

DSC-ACK

TransactionID		1
ConfirmationCode		Success (0)
HMAC		

5) Once the J.112 reservation is complete, and the backbone reservation is successful, the AN responds to the RSVP-PATH message by sending an RSVP-RESV message. The message includes the ResourceID that is assigned by the AN to this connection. The RSVP-RESV message is sent with the source address of MTAT and destination address of MTAo. All intermediate routers will intercept, process, and forward this as a standard RSVP-RESV message.

RSVP-RESV

Session-Object	Protocol	UDP	These fields identify the IP flow for which the reservation is being established.
	Destination Address	MTAt2	
	Destination port	7 000	
Filter-Spec	Source Address	MTAo	
	Source port	7 122	
Flowspec	b	120	These fields identify the resources being reserved for this flow.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	
ResourceID		472	Resource ID for this reservation.

6) In response to a hookflash, and after performing further signalling with both the previous and new parties, MTAo sends the COMMIT message to the AN. This message is directed to the AN at a UDP port determined via call signalling.

7) The Session-Object and Sender-Template give the AN enough information to identify the "gate" and to identify which reserved resources are being committed. Since no Tspecs are given in this message, all the reserved resources will be activated. All other flows assigned the same ResourceID will be deactivated.

COMMIT

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port must match those of the Gate ID.
	Destination Address	MTAt2	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 122	
Gate-ID		37 126	

8) The AN resolves which reservation is to be activated, and sends a DSC-REQ to the CM to activate the flow.

DSC-REQ

TransactionID		2
UpstreamServiceFlow	ServiceFlowIdentifier	1 001
	QoSParameterSetType	Admitted + Activated (6)
	TimeOutActive	10
	ServiceFlowScheduling	UGS (6)
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111
DownstreamServiceFlow	ServiceFlowIdentifier	2 001
	QoSParameterSetType	Admitted + Activated (6)
	TimeOutActive	10
	TrafficPriority	5
	MaximumSustainedRate	12 000
Upstream Classifier	ServiceFlowIdentifier	1 001
	PacketClassifierIdentifier	3 001
	ClassifierChangeAction	Replace (1)
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAo
	IPSourcePort	7 120
	IPDestinationAddress	MTAt
	IPDestinationPort	7 000
	IPProtocol	UDP (17)

DSC-REQ

Downstream Classifier	ServiceFlowIdentifier	2 001
	PacketClassifierIdentifier	3 002
	ClassifierChangeAction	Replace (1)
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAt
	IPSourcePort	7 000
	IPDestinationAddress	MTAo
	IPDestinationPort	7 120
	IPProtocol	UDP (17)
UpstreamPacketClassification	ServiceFlowIdentifier	1 001
	PacketClassifierIdentifier	3 003
	ClassifierChangeAction	Replace (1)
	ClassifierPriority	150
	ClassifierActivationState	Active (1)
	IPSourceAddress	MTAo
	IPSourcePort	7 122
	IPDestinationAddress	MTAt2
	IPDestinationPort	7 000
	IPProtocol	UDP (17)
DownstreamPacketClassification	ServiceFlowIdentifier	2 001
	PacketClassifierIdentifier	3 004
	ClassifierChangeAction	Replace (1)
	ClassifierPriority	150
	ClassifierActivationState	Active (1)
	IPSourceAddress	MTAt2
	IPSourcePort	7 000
	IPDestinationAddress	MTAo
	IPDestinationPort	7 122
	IPProtocol	UDP (17)
HMAC		

9) The CM sends a DSC-RSP message showing the operation was successful.

DSC-RSP

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

10) The AN sends DSC-ACK message to indicate that the DSC-RSP has been received and agreed.

DSC-ACK

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

11) The AN acknowledges the COMMIT with:

COMMIT-ACK

Session-Object	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple matches Gate ID.
	Destination Address	MTAt2	
	Destination port	7 000	
Sender Templ	Source Address	MTAo	
	Source port	7 122	
Gate-ID		37 126	

Annex J (informative): Sample Protocol Message Exchanges for Basic DCS On-Net to On-Net Call of an Embedded MTA

J.1 Example Call Flow with J.112 annex A messages

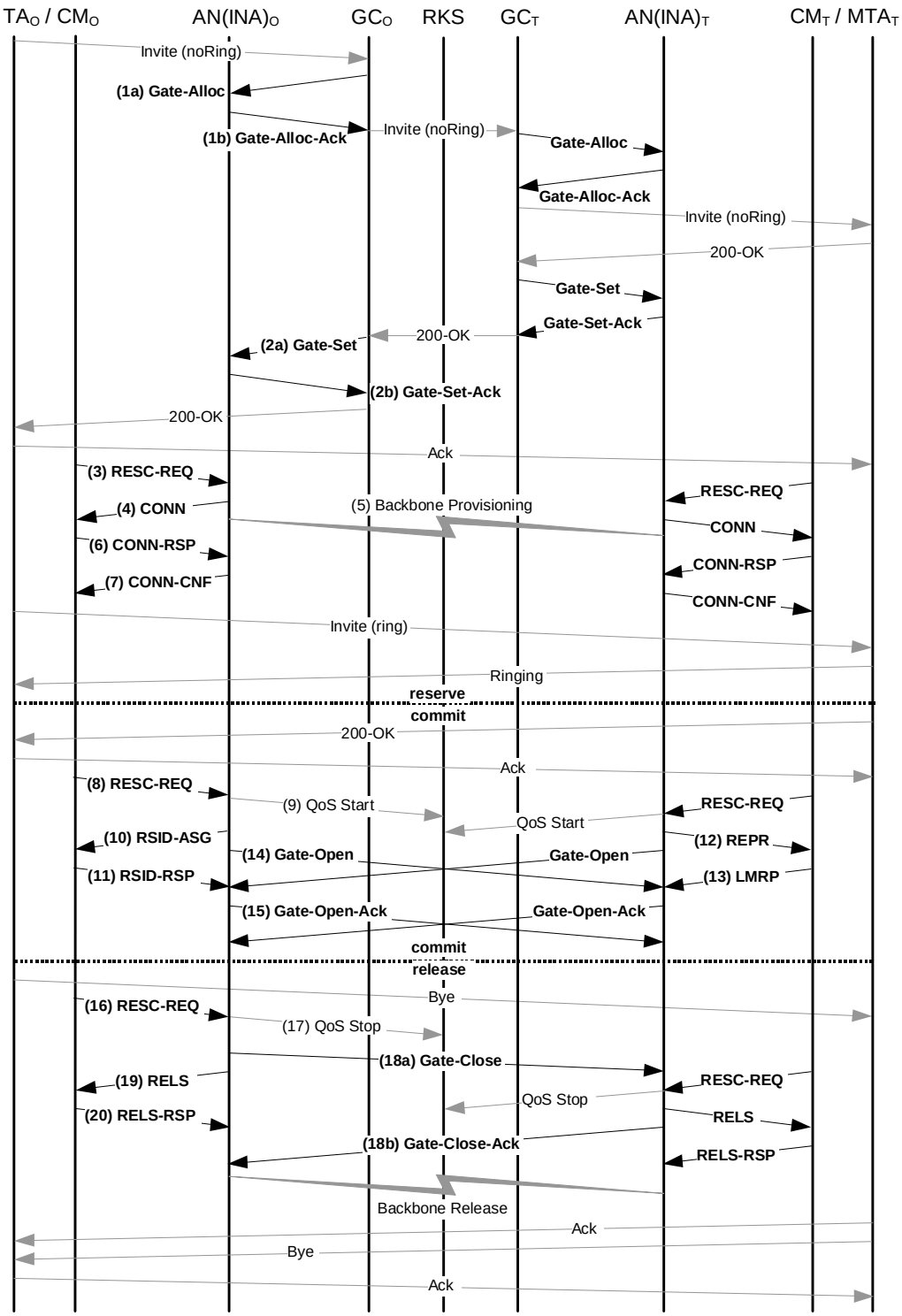


Figure 40: Basic Call Flow with J.112 annex A messages - DCS on Embedded MTAs

1) GCo, upon receipt of signalling information from MTAo, checks the current resource consumption of MTAo by consulting ANo (1a).

GATE-ALLOC

TransactionID		3 176	
Subscriber		MTAo	Request for total resources in use by this endpoint.
Activity-Count		4	Maximum number of gates allowed for this subscriber.

ANo checks current resource usage by MTAo, and responds telling the number of allocated gates (1b).

GATE-ALLOC-ACK

TransactionID		3 176	
Subscriber		MTAo	Answer to request for total resources in use by this endpoint.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		3	Total number of gates established for this subscriber.
Gate Co-ordination Port		4 104	UDP port at which AN will listen for gate co-ordination messages.

2) GCo, upon further signalling exchanges, gives ANo authorization to initiate the reserve phase of the resource allocation process for the new J.112 Flow (2a).

GATE-SET

Transaction ID		3 177	Unique Transaction ID for this message exchange.
Subscriber		MTAo	Request for the specification of the previously allocated gate.
Gate-ID		37 125	Identifier for allocated Gate.
Remote-Gate-Info	Address	ANt	Information needed to perform gate co-ordination.
	Port	2 052	
	Remote Gate-ID	1 273	
	Security Key	<key>	
Event-Generation-Info	RKS-Addr	RKS	Address of Record Keeping Server.
	RKS-Port	3 288	Port on Record Keeping Server.
	Billing Correlation ID	<id>	Opaque data that will be passed to RKS when resources are committed.

GATE-SET

Gate-Spec	Direction	up	
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAo	
	Destination Address	MTAt	
	Source port	0	
	Destination port	7 000	
	DSCP	6	Packet Type value for upstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

GATE-SET

Gate-Spec	Direction	Down	
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAt	
	Destination Address	MTAo	
	Source port	0	
	Destination port	7 120	
	DSCP	9	Packet Type value for downstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

ANo responds to the Gate-Set command with an acknowledgement (2b).

GATE-SET-ACK

TransactionID		3 177	
Subscriber		MTAo	Answer to request for the specification of the previously allocated gate.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		4	Total number of gates established for this subscriber.

3) MTAo, upon receiving call signalling information, calculates the QoS parameters for the J.112 link. It uses the MAC-layer interface to instruct CMo to send a Resource Request message to ANo. Assuming that an upstream rate of 3,088 Mbit/s is used and IP packets are encapsulated using DirectIP, the upstream resources are calculated as follows. An IP packet of size 120 Byte (from Tspec) including the 5-Byte AAL 5 Trailer fits in 3 ATM cells. Thus, using Reservation Access mode ANo has to grant 3 slots every 10 ms. In Fixed-rate Access mode, a Cyclic Assignment of 3 slots at a time is required with a Maximum Distance of 60 slots. The Requested Bandwidth is 360 slots per 1 200 ms.

RESC-REQ

Resource_Request_ID	0x01
Connection_ID	37 125 <Gate ID>
Field	
Aux_Control_Field_included	1 <yes>
Admit_Flag	1 <reservation requested>
Flowspec_DS_included	1 <yes>
Priority_included	0 <no>
Max_packet_size_included	1 <yes>
Session_binding_US_included	0 <no>
Release_requested	0 <no>
Reservation_ID_requested	0 <no>
Cyclic_Assignment_needed	1 <yes>
Requested_Bandwidth	360 <slots per 1 200 ms>
Maximum_Distance_Between_Slots	60 <slots>
Encapsulation	DirectIP (1)
Aux_Control_Field	
IPv6_Add	0 <no>
Flowspec_DS_included	1 <yes>
Session_binding_DS_included	0 <no>
Frame_Length	3
Flowspec_DS	
Max_Packet_Size	120
Average_Bitrate	12 000
Jitter	0 <ms>

4) ANo detects the Resource Request and cannot match the Connection ID included with an existing J.112 Flow. Thus, it checks the authorization by looking for a Gate ID that matches the Connection ID. If the Gate was already set, ANo is able to verify that the requested resources are within the authorized envelop. If this is the case, ANo sends the following Connect Message to CMo. This message is used to establish both upstream and downstream parameters. However, no resources are allocated in the Connect Message. That indicates to CMo that the resources for that J.112 Flow are reserved but not yet committed.

CONN

Connection_ID	37 125 <Gate ID>
Session_number	<not used>
Connection_Control_Field_Aux	
Connection_control_field2_included	1 <yes>
IPv6_add	0 <no>
Priority_included	0 <no>
Flowspec_DS_included	0 <no>
Session_binding_US_included	0 <no>
Session_binding_DS_included	0 <no>
Encapsulation_included	1 <yes>
DS_multiprotocol_CBD_included	0 <no>
Resource_number	0x01
Connection_Control_Field	
DS_ATM_CBD_included	0 <no>
DS_MPEG_CBD_included	1 <yes>
US_ATM_CBD_included	1 <yes>
Upstream_Channel_Number	0x1
Slot_list_included	0 <no>
Cyclic_assignment	0 <no>
Frame_Length	0 <no>
Maximum_Contention_Access_Message_Length	1 <slots>
Maximum_Reservation_Access_Message_Length	50 <slots>
Downstream_MPEG_CBD	
Downstream_Frequency	472 000 000 <Hz>
Program_Number	0xA437
Upstream_ATM_CBD	
Upstream_Frequency	20 000 000 <Hz>
Upstream_VPI	0x01
Upstream_VCI	0x54AC
MAC_Flag_Set	0x01
Upstream_Rate	Upstream_3.088M
Encapsulation	DirectIP (1)

CONN

Connection_control_field2	
Upstream_modulation_included	1 <yes>
Upstream_Modulation	QPSK (1)

5) Simultaneous with message No. 4, ANo initiates any required backbone reservations for the requested quality of service. The content of this message is dependent on the particular backbone algorithms in use, and is outside the scope of the present document. The backbone router sends to ANo any required notification that the reservation is successful.

6) CMo checks the resources it is required to allocate (e.g. header suppression context, Connection IDs, classifier context), and installs the classifiers. If the operation is successful it returns the Connect Response message stating the success.

CONN-RSP

Connection_ID	37 125 <Gate ID>

7) Upon receipt of the Connect Response message, ANo acknowledges receipt with a Connect Confirm message.

CONN-CNF

Connection_ID	37 125 <Gate ID>

8) In response to signalling messages that indicate the call set-up has completed (i.e. the other side has gone off-hook), MTAo uses the J.112 MAC-layer interface to initiate the commitment of the reserved resources. This is done via CMo sending a Resource Request message.

RESC-REQ

Resource_Request_ID	0x02
Connection_ID	37 125 <Gate ID>
Field	
Aux_Control_Field_included	1 <yes>
Admit_Flag	0 <commitment requested>
Flowspec_DS_included	1 <yes>
Priority_included	0 <no>
Max_packet_size_included	1 <yes>
Session_binding_US_included	0 <no>
Release_requested	0 <no>
Reservation_ID_requested	0 <no>
Cyclic_Assignment_needed	1 <yes>
Requested_Bandwidth	360 <slots per 1 200 ms>
Maximum_Distance_Between_Slots	60 <slots>

RESC-REQ

Encapsulation	DirectIP (1)
Aux_Control_Field	
IPv6_Add	0 <no>
Flowspec_DS_included	1 <yes>
Session_binding_DS_included	0 <no>
Frame_Length	3
Flowspec_DS	
Max_Packet_Size	120
Average_Bitrate	12 000
Jitter	0 <ms>

9) ANo sends the event record to the Record Keeping Server that enhanced Quality of Service has been granted to this call. The format of this message is described in [J.em].

10) The AN may commit the reserved resources either using Fixed-rate Access mode or Reservation Access mode. Upon reception of the Resource Request message, it needs to send the appropriate MAC-layer messages to complete the set-up of a J.112 Flow.

- For this example, it is assumed that ANo decides to use Reservation Access mode while ANt commits resources in Fixed-rate Access mode.
- Continuous Piggybacking is used to accomodate the CBR like characteristic of this traffic. To initiate the transmission ANo sends a Reservation ID Assignment message.

RSID-ASG

Connection_ID	37 125 <Gate ID>
Reservation_ID	0x1234
Grant_Protocol_Timeout	15 <ms>
Piggy_Back_Request_Values	
Continuous_Piggy_Back_Timeout	4 <36 ms>
GFC_11_Slots	9 <slots>
GFC_10_Slots	3 <slots>
GFC_01_Slots	1 <slots>

11) CMo sends a Reservation ID Response message showing the operation was successful.

RSID-RSP

Connection_ID	37 125 <Gate ID>
Reservation_ID	0x1234
Grant_Protocol_Timeout	15 <ms>

12) ANt at the terminating side of the call decided to provide the requested resources using Fixed-rate Access mode. To commit the resources and to initiate the transmission ANt sends a Reprovision message to CMt.

REPR

Reprovision_Control_Field	
Reprovision_Control_Aux_Field_included	0 <no>
Delete_Reservation_IDs	0 <no>
New_Downstream_IB_Frequency_included	0 <no>
New_Downstream_OOB_Frequency_included	0 <no>
New_Upstream_Frequency_included	0 <no>
New_Frame_Length_included	1 <yes>
New_Cyclical_Assignment_included	1 <yes>
New_Slot_List_included	0 <no>
New_Frame_Length	3
Number_of_Connections	1
Connection_ID	1 273 <Gate ID>
Cyclic_Assignment	
Fixedrate_Start	0x0000
Fixedrate_Dist	60
Fixedrate_Stop	0xFFFF

13) CMt sends a Link Management Response message showing the operation was successful.

LMRP

Link_Management_Msg_Number	<Reprovision Message Type Value>

14) ANo sends the gate co-ordination message to the remote ANt to inform it that the resources at the local end have been committed.

GATE-OPEN

Transaction ID		72	Identifier to match this message with its response.
Gate ID		1 273	Gate-ID at AN receiving this message.
Tspec	b	120	These are the traffic parameters actually being utilized for the resources committed to the flow in the MTAo to MTAt direction.
	r	12 000	
	p	12 000	
	m	120	
	M	120	

Reverse Tspec	b	120	These are the expected traffic parameters being utilized for the flow in the MTAt to MTAo direction.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
HMAC			Security checksum for this message.

15) Upon reception of the GATE-OPEN message from the remote ANt, ANo responds with a GATE-OPEN-ACK.

GATE-OPEN-ACK

Transaction ID		8 096	Identifier to match this message with its request.
HMAC			Security checksum for this message.

16) When the call is finished MTAo uses the J.112 MAC-layer interface to release the reserved resources. This is done via CMo sending a Resource Request message.

RESC-REQ

Resource_Request_ID	0x04
Connection_ID	37 125 <Gate ID>
Field	
Aux_Control_Field_included	0 <no>
Admit_Flag	0
Flowspec_DS_included	0 <no>
Priority_included	0 <no>
Max_packet_size_included	0 <no>
Session_binding_US_included	0 <no>
Release_requested	1 <yes>
Reservation_ID_requested	0 <no>
Cyclic_Assignment_needed	0 <no>
Requested_Bandwidth	0
Maximum_Distance_Between_Slots	0
Encapsulation	DirectIP (1)

17) ANo sends the notification to the Record Keeping Server that the call has ended. The format of this event message is described in [J.em].

18) ANo, upon receiving the request to release the resources, sends the gate co-ordination message to the address given in the GATE-SET command earlier, which in the case of DCS is ANt serving MTAt (18a).

GATE-CLOSE

Transaction ID		73	Identifier to match this message with its response.
Gate-ID		1 273	GateID at the network element receiving this message.
HMAC			Security checksum for this message.

ANt responds with a GATE-CLOSE-ACK message (18b).

GATE-CLOSE-ACK

Transaction ID		73	Identifier to match this message with its request.
HMAC			Security checksum for this message.

19) ANo answers the Resource Request message sending a Release message to CMo indicating the J.112 Flow that is to be deleted.

RELS

Number_of_Connections	1
Connection_ID	37 125 <Gate ID>

20) CMo releases the J.112 Flow and sends the Release Response to ANo.

RELS-RSP

Connection_ID	37 125 <Gate ID>

J.2 Example Call Flow with J.112 annex B/annex C messages

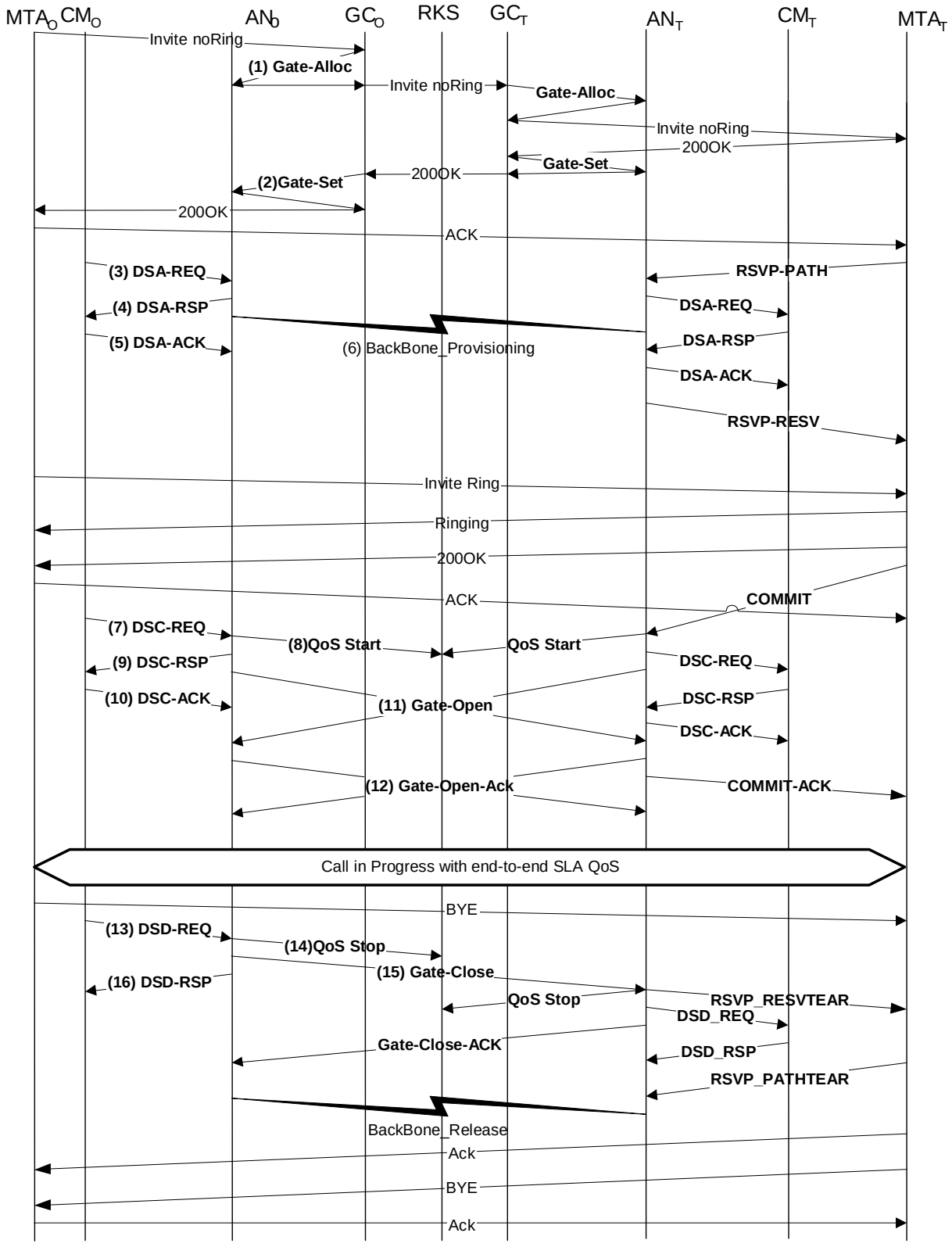


Figure 41: Basic Call Flow - Embedded MTA

1) GCo, upon receipt of signalling information from MTAo, checks the current resource consumption of MTAo by consulting ANo.

GATE-ALLOC

TransactionID		3 176	
Subscriber		MTAo	Request for total resources in use by this client.
Activity-Count		4	Maximum connections allowed by client.

ANo checks current resource usage by MTAo, and responds telling the number of connections active.

GATE-ALLOC-ACK

TransactionID		3 176	
Subscriber		MTAo	Request for total resources in use by this client.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		3	Total connections established by this client.

2) GCo, upon further signalling exchanges, gives ANo authorization to admit the new connection.

GATE-SET

Transaction ID		3 177	Unique Transaction ID for this message exchange.
Subscriber		MTAo	Request for total resources in use by this client.
Gate-ID		37 125	Identifier for allocated Gate.
Remote-Gate-Info	AN Address	ANt	Information needed to perform gate co-ordination.
	AN Port	2 052	
	Remote Gate-ID	1 273	
	Security Key	<key>	
Event-Generation-Info	RKS-Addr	RKS	Address of Record Keeping Server.
	RKS-Port	3 288	Port on Record Keeping Server.
	Billing Correlation ID	<id>	Opaque data that will be passed to RKS when resources are committed.

GATE-SET

Gate-Spec	Direction	up	
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAo	
	Destination Address	MTAt	
	Source port	0	
	Destination port	7 000	
	DSCP	6	Packet Type value for upstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

GATE-SET

Gate-Spec	Direction	down	
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAt	
	Destination Address	MTAo	
	Source port	0	
	Destination port	7 120	
	DSCP	9	Packet Type value for downstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

3) ANo respond to the Gate Setup command with an acknowledgement.

GATE-SET-ACK

TransactionID		3 177	
Subscriber		MTAo	Request for total resources in use by this client.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		4	Total connections established by this client.

4) MTAo, upon receiving call signalling information, calculates the QoS parameters for the J.112 link. It uses the Appendix E interface to the CM to send the following DSA-REQ to the AN. This message is used to establish both upstream and downstream parameters. The Upstream Unsolicited Grant Size was calculated as 120 (from SDP) plus 18 (Ethernet overhead) minus 40 (Header Suppression amount) plus 13 (J.112 overhead). Header suppression indicates the 42 bytes of Ethernet/IP/UDP header. Contents of the suppressed header is included in the DSA-REQ.

DSA-REQ

TransactionID		1
UpstreamServiceFlow	ServiceFlowReference	1
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	ServiceFlowScheduling	UGS (6)
	Request/TransmissionPolicy	0x00000017
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111
	AuthBlock	37 125
DownstreamServiceFlow	ServiceFlowReference	2
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	TrafficPriority	5
	MaximumSustainedRate	12 000
	AuthBlock	37 125
UpstreamPacketClassification	ServiceFlowReference	1
	PacketClassifierReference	1
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAo
	IPSourcePort	7 120
	IPDestinationAddress	MTAt
	IPDestinationPort	7 000
	IPProtocol	UDP (17)

DSA-REQ

DownstreamPacketClassification	ServiceFlowReference	2
	PacketClassifierReference	2
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAt
	IPSourcePort	7 000
	IPDestinationAddress	MTAo
	IPDestinationPort	7 124
	IPProtocol	UDP (17)
PayloadHeaderSuppression	ClassifierReference	1
	ServiceFlowReference	1
	HeaderSuppressionIndex	1
	HeaderSuppressionField	<42bytes>
	HeaderSuppressionMask	<42bits>
	HeaderSuppressionSize	42
	HeaderSuppressionVerify	Verify (0)
AuthorizationBlock		37 125
HMAC		

5) The AN checks the authorization, by looking for a gate with Gate-ID matching the value in AuthBlock, and checks the resources it is required to allocate (e.g. header suppression table space, Service Flow IDs, classifier table space), and installs the classifiers. If the operation is successful it returns the DSA-RSP message stating the success.

DSA-RSP

TransactionID		1
ConfirmationCode		Success (0)
UpstreamServiceFlow	ServiceFlowReference	1
	ServiceFlowIdentifier	1 001
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	ServiceFlowScheduling	UGS (6)
	Request/TransmissionPolicy	0x00000017
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111
	AuthBlock	37 125
DownstreamServiceFlow	ServiceFlowReference	2
	ServiceFlowIdentifier	2 001
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	TrafficPriority	5
	MaximumSustainedRate	12 000
	AuthBlock	37 125
UpstreamPacketClassification	ServiceFlowReference	1
	PacketClassifierReference	1
	PacketClassifierIdentifier	3 001
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAo
	IPSourcePort	7 120
	IPDestinationAddress	MTAt
	IPDestinationPort	7 000
	IPProtocol	UDP (17)

DSA-RSP

DownstreamPacketClassification	ServiceFlowReference	2
	PacketClassifierReference	2
	PacketClassifierIdentifier	3 002
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAt
	IPDestinationAddress	MTAo
	IPDestinationPort	7 124
	IPProtocol	UDP (17)
HMAC		

6) Upon receipt of the DSA-RSP, the CM acknowledges receipt with a DSA-ACK message.

DSA-ACK

TransactionID		1
ConfirmationCode		Success (0)
HMAC		

7) Simultaneous with message No. 4, the AN initiates any required backbone reservations for the requested quality of service. The content of this message is dependent on the particular backbone algorithms in use, and is outside the scope of the present document. The backbone router sends to the AN any required notification that the reservation is successful.

8) In response to signalling messages that indicate the call has completed (i.e. the other side has gone off-hook), MTAo uses the Appendix E interface to activate the admitted resources. This is done via a DSC-REQ command to the AN.

DSC-REQ

TransactionID		2
UpstreamServiceFlow	ServiceFlowIdentifier	1 001
	QoSParameterSetType	Admitted + Activated (6)
	TimeOutActive	10
	ServiceFlowScheduling	UGS (6)
	Request/TransmissionPolicy	0x00000017
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111
DownstreamServiceFlow	ServiceFlowIdentifier	2 001
	QoSParameterSetType	Admitted + Activated (6)
	TimeOutActive	10
	TrafficPriority	5
	MaximumSustainedRate	12 000
UpstreamPacketClassification	ServiceFlowIdentifier	1 001
	PacketClassifierIdentifier	3 001
	ClassifierChangeAction	Replace (1)
	ClassifierPriority	150
	ClassifierActivationState	Active (1)
	IPSourceAddress	MTAo
	IPSourcePort	7 120
	IPDestinationAddress	MTAt
	IPDestinationPort	7 000
	IPProtocol	UDP (17)

DSC-REQ

DownstreamPacketClassification	ServiceFlowIdentifier	2 001
	PacketClassifierIdentifier	3 002
	ClassifierChangeAction	Replace (1)
	ClassifierPriority	150
	ClassifierActivationState	Active (1)
	IPSourceAddress	MTAt
	IPSourcePort	7 000
	IPDestinationAddress	MTAo
	IPDestinationPort	7 124
	IPProtocol	UDP (17)
HMAC		

9) ANo sends the event record to the Record Keeping Server that a Commit has been received on this call. This message is only a sample of what might be included in a QoS-Start message:

QoS-START

Header	Timestamp	<time>	Time of the event being recorded.
	Billing Correlation ID	<string>	Correlation ID given in Gate-Set.
QoS Descriptor	Type	UGS	Description of the QoS provided for this connection.
	Grant interval	10 ms	
	Grant Jitter	2 ms	
	Grant/Interval	1	
	Grant Size	111	
MTA Port	Port	7 120	

10) The AN sends a DSC-RSP message showing the operation was successful.

DSC-RSP

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

11)The CM sends DSC-ACK message to indicate that the DSC-RSP has been received and agreed.

DSC-ACK

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

12)The AN sends the gate co-ordination message to the remote AN to inform it that the resources at this end have been committed.

GATE-OPEN

Transaction ID		72	Identifier to match this message with its response.
Gate ID		1 273	Gate-ID at remote AN.
Tspec	b	120	These are the committed traffic parameters actually being utilized in the MTAo to MTAt direction.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
Reverse-Tspec	b	120	These are the expected traffic parameters being utilized in the MTAt to MTAo direction.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
HMAC			Security checksum for this message.

13)The remote AN responds to the GATE-OPEN with:

GATE-OPEN-ACK

Transaction ID		72	Identifier to match this message with its response.
HMAC			Security checksum for this message.

14) When the call is finished the MTA uses the Appendix E interface to delete the Service Flows, sending a DSD-REQ message to the AN.

DSD-REQ

TransactionID		3
ServiceFlowID		1 001
HMAC		

DSD-REQ

TransactionID		4
ServiceFlowID		2 001
HMAC		

15) The AN sends the notification to the Record Keeping Server that the call has ended. This message is only a sample of what might be included in a QoS-Stop message (refer to J.ev).

QoS-Stop

TimeStamp		<time>	The time of the event being recorded.
Header	Time Stamp	<time>	Time of the event being recorded.
	Billing Correlation ID	<string>	Correlation ID from Gate-Set message.
SF-ID	SF-ID	1 001	Service Flow Identifier.

16) The AN, upon receiving RSVP-PATH-TEAR, sends the gate co-ordination message to its corresponding AN serving MTAt.

GATE-CLOSE

Transaction ID		73	Identifier to match this message with its response.
Gate-ID		1 273	This identifies the GateID at the remote AN.
HMAC			Security checksum for this message.

The remote AN responds with:

GATE-CLOSE-ACK

Transaction ID		73	Identifier to match this message with its response.
HMAC			Security checksum for this message.

17) The AN deletes the Service Flow IDs and sends the response to the CM.

DSD-RSP

TransactionID		3
ServiceFlowID		1 001
ConfirmationCode		Success (0)
HMAC		

DSD-RSP

TransactionID		4
ServiceFlowID		2 001
ConfirmationCode		Success (0)
HMAC		

Annex K (informative): Sample Protocol Message Exchanges for Basic NCS Call for Embedded MTA

K.1 Example Call Flow with J.112 annex A messages

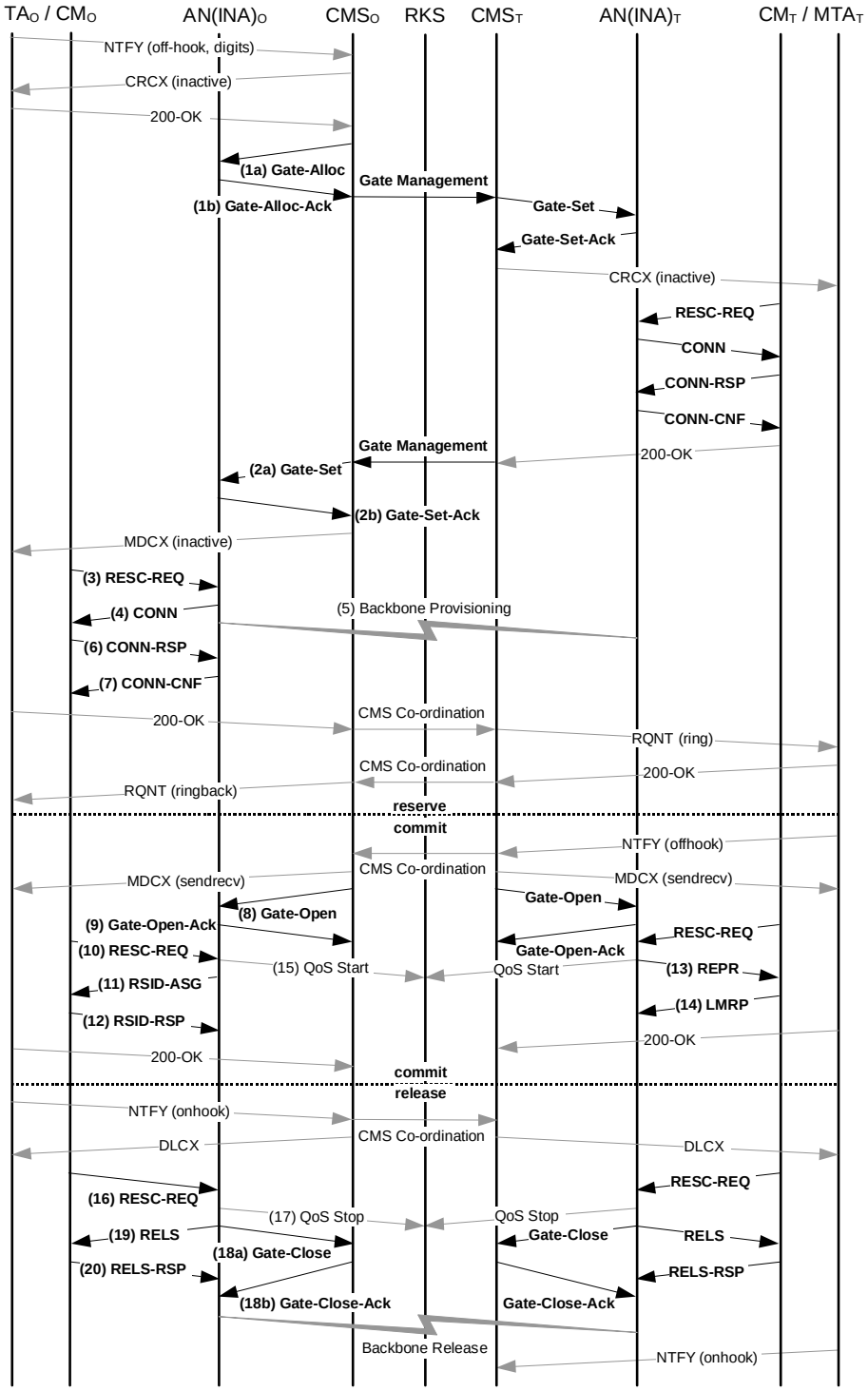


Figure 42: Basic Call Flow with J.112 annex A messages - NCS on Embedded MTAs

1) GCo/CMSo, upon receipt of signalling information from MTAo, checks the current resource consumption of MTAo by consulting ANo (1a).

GATE-ALLOC

TransactionID		3 176	
Subscriber		MTAo	Request for total resources in use by this endpoint.
Activity-Count		4	Maximum number of gates allowed for this subscriber.

ANo checks current resource usage by MTAo, and responds telling the number of allocated gates (1b).

GATE-ALLOC-ACK

TransactionID		3 176	
Subscriber		MTAo	Answer to request for total resources in use by this endpoint.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		3	Total number of gates established for this subscriber.

2) GCo/CMSo, upon further signalling exchanges, gives ANo authorization to initiate the reserve phase of the resource allocation process for the new J.112 Flow (2a).

GATE-SET

Transaction ID		3 177	Unique Transaction ID for this message exchange.
Subscriber		MTAo	Request for the specification of the previously allocated gate.
Gate-ID		37 125	Identifier for allocated Gate.
Remote-Gate-Info	Address	CMSo	Information needed to perform gate co-ordination. Note that CMSo has given itself as the entity for exchanging gate co-ordination messages. Flag value indicates that the AN should not send a Gate-Open message when it receives a COMMIT from the MTA, but still expect to receive a Gate-Open message from CMSo.
	Port	2 052	
	Remote Gate-ID	8 095	
	Security Key	<key>	
	Flag	No-gate-open	
Event-Generation-Info	RKS-Addr	RKS	Address of Record Keeping Server.
	RKS-Port	3 288	Port on Record Keeping Server.
	Billing Correlation ID	<id>	Opaque data that will be passed to RKS when resources are committed.

GATE-SET

Gate-Spec	Direction	up	
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAo	
	Destination Address	MTAt	
	Source port	0	
	Destination port	7 000	
	DSCP	6	Packet Type value for upstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

GATE-SET

Gate-Spec	Direction	Down	
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAt	
	Destination Address	MTAo	
	Source port	0	
	Destination port	7 120	
	DSCP	9	Packet Type value for downstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

ANo responds to the Gate Setup command with an acknowledgement (2b).

GATE-SET-ACK

TransactionID		3 177	
Subscriber		MTAo	Request for the specification of the previously allocated gate.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		4	Total number of gates established for this subscriber.

3) MTAo, upon receiving a Modify-Connection command, calculates the QoS parameters for the J.112 link. It uses the MAC-layer interface to instruct CMo to send a Resource Request message to ANo. Assuming that an upstream rate of 3,088 Mbit/s is used and IP packets are encapsulated using DirectIP, the upstream resources are calculated as follows. An IP packet of size 120 Byte (from Tspec) including the 5-Byte AAL 5 Trailer fits in 3 ATM cells. Thus, using Reservation Access mode the AN has to grant 3 slots every 10 ms. In Fixed-rate Access mode, a Cyclic Assignment of 3 slots at a time is required with a Maximum Distance of 60 slots. The Requested Bandwidth is 360 slots per 1 200 ms.

RESC-REQ

Resource_Request_ID	0x01
Connection_ID	37 125 <Gate ID>
Field	
Aux_Control_Field_included	1 <yes>
Admit_Flag	1 <reservation requested>
Flowspec_DS_included	1 <yes>
Priority_included	0 <no>
Max_packet_size_included	1 <yes>
Session_binding_US_included	0 <no>
Release_requested	0 <no>
Reservation_ID_requested	0 <no>
Cyclic_Assignment_needed	1 <yes>
Requested_Bandwidth	360 <slots per 1 200 ms>
Maximum_Distance_Between_Slots	60 <slots>
Encapsulation	DirectIP (1)
Aux_Control_Field	
IPv6_Add	0 <no>
Flowspec_DS_included	1 <yes>
Session_binding_DS_included	0 <no>
Frame_Length	3
Flowspec_DS	
Max_Packet_Size	120
Average_Bitrate	12 000
Jitter	0 <ms>

4) ANo detects the Resource Request and cannot match the Connection ID included with an existing J.112 Flow. Thus, it checks the authorization by looking for a Gate ID that matches the Connection ID. If the Gate was already set, ANo is able to verify that the requested resources are within the authorized envelop. If this is the case, ANo sends the following Connect Message to CMo. This message is used to establish both upstream and downstream parameters. However, no resources are allocated in the Connect Message. That indicates to CMo that the resources for that J.112 Flow are reserved but not yet committed.

CONN

Connection_ID	37 125 <Gate ID>
Session_number	<not used>
Connection_Control_Field_Aux	
Connection_control_field2_included	1 <yes>
IPv6_add	0 <no>
Priority_included	0 <no>
Flowspec_DS_included	0 <no>
Session_binding_US_included	0 <no>
Session_binding_DS_included	0 <no>
Encapsulation_included	1 <yes>
DS_multiprotocol_CBD_included	0 <no>
Resource_number	0x01
Connection_Control_Field	
DS_ATM_CBD_included	0 <no>
DS_MPEG_CBD_included	1 <yes>
US_ATM_CBD_included	1 <yes>
Upstream_Channel_Number	0x1
Slot_list_included	0 <no>
Cyclic_assignment	0 <no>
Frame_Length	0 <no>
Maximum_Contention_Access_Message_Length	1 <slots>
Maximum_Reservation_Access_Message_Length	50 <slots>
Downstream_MPEG_CBD	
Downstream_Frequency	472 000 000 <Hz>
Program_Number	0xA437
Upstream_ATM_CBD	
Upstream_Frequency	20 000 000 <Hz>
Upstream_VPI	0x01
Upstream_VCI	0x54AC
MAC_Flag_Set	0x01
Upstream_Rate	Upstream_3.088M
Encapsulation	DirectIP (1)

CONN

Connection_control_field2	
Upstream_modulation_included	1 <yes>
Upstream_Modulation	QPSK (1)

5) Simultaneous with message No. 4, ANo initiates any required backbone reservations for the requested quality of service. The content of this message is dependent on the particular backbone algorithms in use, and is outside the scope of the present document. The backbone router sends to ANo any required notification that the reservation is successful.

6) CMo checks the resources it is required to allocate (e.g. header suppression context, Connection IDs, classifier context), and installs the classifiers. If the operation is successful it returns the Connect Response message stating the success.

CONN-RSP

Connection_ID	37 125 <Gate ID>

7) Upon receipt of the Connect Response message, ANo acknowledges receipt with a Connect Confirm message.

CONN-CNF

Connection_ID	37 125 <Gate ID>

8) CMSo sends the gate co-ordination message to ANo to inform it that the resources should be committed. If ANo does not receive a Resource Request message from CMo within a reasonable time, it will revoke the gate authorization.

GATE-OPEN

Transaction ID		8 096	Identifier to match this message with its response.
Gate ID		37 125	Gate-ID at AN receiving this message.
HMAC			Security checksum for this message.

9) ANo responds to the GATE-OPEN with a GATE-OPEN-ACK.

GATE-OPEN-ACK

Transaction ID		8 096	Identifier to match this message with its request.
HMAC			Security checksum for this message.

10) In response to a Modify-Connection command, which indicates the call set-up has completed (i.e. the other side has gone off-hook), MTAo uses the J.112 MAC-layer interface to initiate the commitment of the reserved resources. This is done via CMo sending a Resource Request message.

RESC-REQ

Resource_Request_ID	0x02
Connection_ID	37 125 <Gate ID>
Field	
Aux_Control_Field_included	1 <yes>
Admit_Flag	0 <commitment requested>
Flowspec_DS_included	1 <yes>
Priority_included	0 <no>
Max_packet_size_included	1 <yes>
Session_binding_US_included	0 <no>
Release_requested	0 <no>
Reservation_ID_requested	0 <no>
Cyclic_Assignment_needed	1 <yes>
Requested_Bandwidth	360 <slots per 1 200 ms>
Maximum_Distance_Between_Slots	60 <slots>
Encapsulation	DirectIP (1)
Aux_Control_Field	
IPv6_Add	0 <no>
Flowspec_DS_included	1 <yes>
Session_binding_DS_included	0 <no>
Frame_Length	3
Flowspec_DS	
Max_Packet_Size	120
Average_Bitrate	12 000
Jitter	0 <ms>

11) The AN may commit the reserved resources either using Fixed-rate Access mode or Reservation access mode. Upon reception of the COMMIT message, it needs to send the appropriate MAC-layer messages to complete the set-up of a J.112 Flow.

For this example, it is assumed that ANo decides to use Reservation Access mode while ANt commits resources in Fixed-rate Access mode.

Continuous Piggybacking is used to accomodate the CBR like characteristic of this traffic. To initiate the transmission ANo sends a Reservation ID Assignment message.

RSID-ASG

Connection_ID	37 125 <Gate ID>
Reservation_ID	0x1234
Grant_Protocol_Timeout	15 <ms>
Piggy_Back_Request_Values	
Continuous_Piggy_Back_Timeout	4 <36 ms>
GFC_11_Slots	9 <slots>
GFC_10_Slots	3 <slots>
GFC_01_Slots	1 <slots>

12) CMo sends a Reservation ID Response message showing the operation was successful.

RSID-RSP

Connection_ID	37 125 <Gate ID>
Reservation_ID	0x1234
Grant_Protocol_Timeout	15 <ms>

13) ANt at the terminating side of the call decided to provide the requested resources using Fixed-rate Access mode. To commit the resources and to initiate the transmission ANt sends a Reprovision message to CMt.

REPR

Reprovision_Control_Field	
Reprovision_Control_Aux_Field_included	0 <no>
Delete_Reservation_IDs	0 <no>
New_Downstream_IB_Frequency_included	0 <no>
New_Downstream_OOB_Frequency_included	0 <no>
New_Upstream_Frequency_included	0 <no>
New_Frame_Length_included	1 <yes>
New_Cyclical_Assignment_included	1 <yes>
New_Slot_List_included	0 <no>
New_Frame_Length	3
Number_of_Connections	1
Connection_ID	8 095 <Gate ID>
Cyclic_Assignment	
Fixedrate_Start	0x0000
Fixedrate_Dist	60
Fixedrate_Stop	0xFFFF

14) CMt sends a Link Management Response message showing the operation was successful.

LMRP

Link_Management_Msg_Number	<Reprovision Message Type Value>

15) ANo sends the event record to the Record Keeping Server that enhanced Quality of Service has been granted to this call. The format of this message is described in [J.em].

16) When the call is finished, in response to a Delete-Connection command, MTAo uses the J.112 MAC-layer interface to release the reserved resources. This is done via CMo sending a Resource Request message.

RESC-REQ

Resource_Request_ID	0x04
Connection_ID	37 125 <Gate ID>
Field	
Aux_Control_Field_included	0 <no>
Admit_Flag	0
Flowspec_DS_included	0 <no>
Priority_included	0 <no>
Max_packet_size_included	0 <no>
Session_binding_US_included	0 <no>
Release_requested	1 <yes>
Reservation_ID_requested	0 <no>
Cyclic_Assignment_needed	0 <no>
Requested_Bandwidth	0
Maximum_Distance_Between_Slots	0
Encapsulation	DirectIP (1)

17) ANo sends the event record to the Record Keeping Server that the call has ended. The format of this message is described in [J.em].

18) ANo, upon receiving the Resource Request message, sends the gate co-ordination message to the address given in the GATE-SET command earlier, which in the case of NCS is the Call Agent (18a).

GATE-CLOSE

Transaction ID		73	Identifier to match this message with its response.
Gate-ID		8 095	GateID at the network element (here: CMS) receiving this message.
HMAC			Security checksum for this message.

CMSo responds with a GATE-CLOSE-ACK message (18b).

GATE-CLOSE-ACK

Transaction ID		73	Identifier to match this message with its request.
HMAC			Security checksum for this message.

19) ANo answers the Resource Request message sending a Release message to CMo indicating the J.112 Flow that is to be deleted.

RELS

Number_of_Connections	1
Connection_ID	37 125 <Gate ID>

20) CMo releases the J.112 Flow and sends the Release Response to ANo.

RELS-RSP

Connection_ID	37 125 <Gate ID>

K.2 Example Call Flow with J.112 annex B/annex C messages

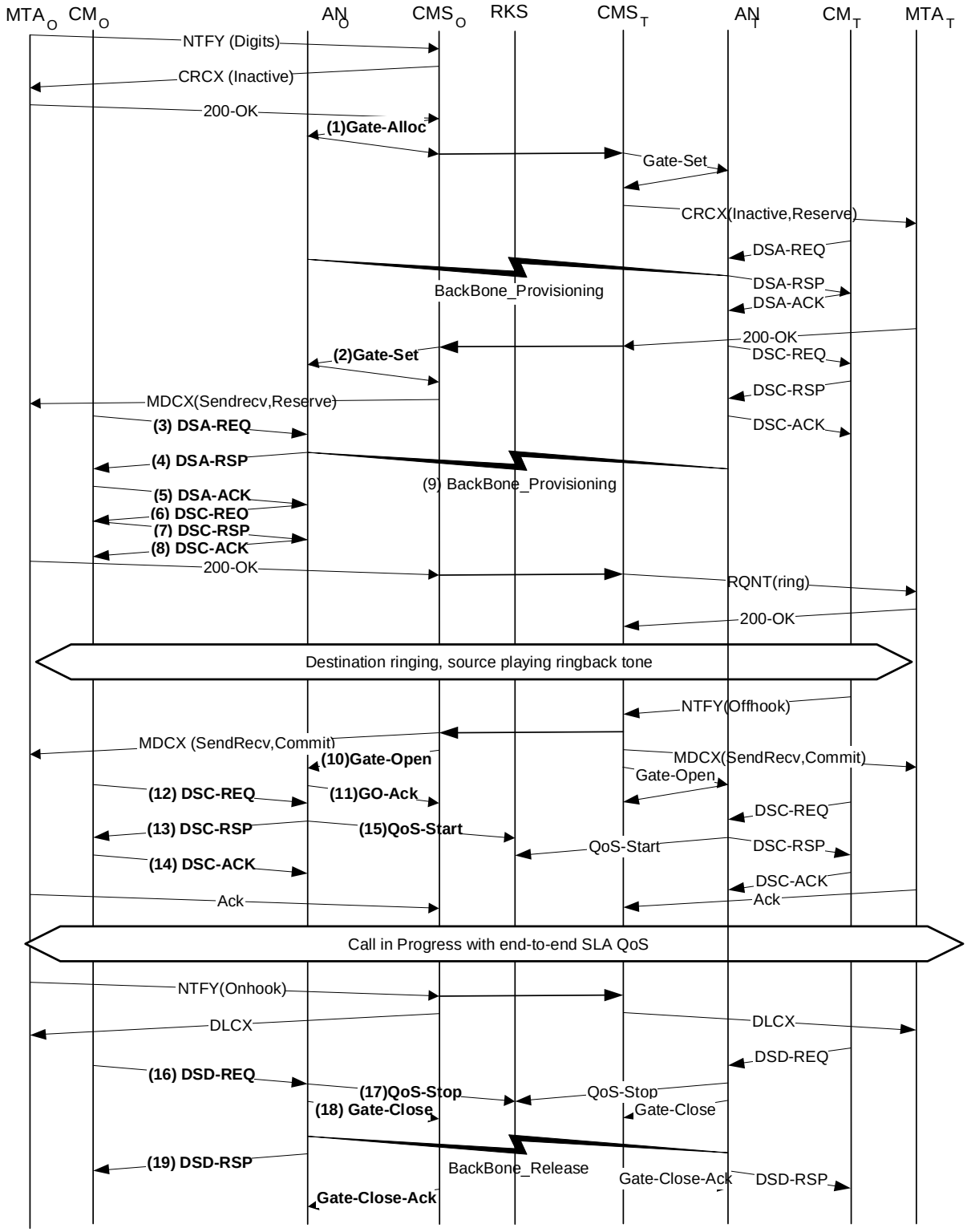


Figure 43: On-Net to On-Net Embedded NCS Call

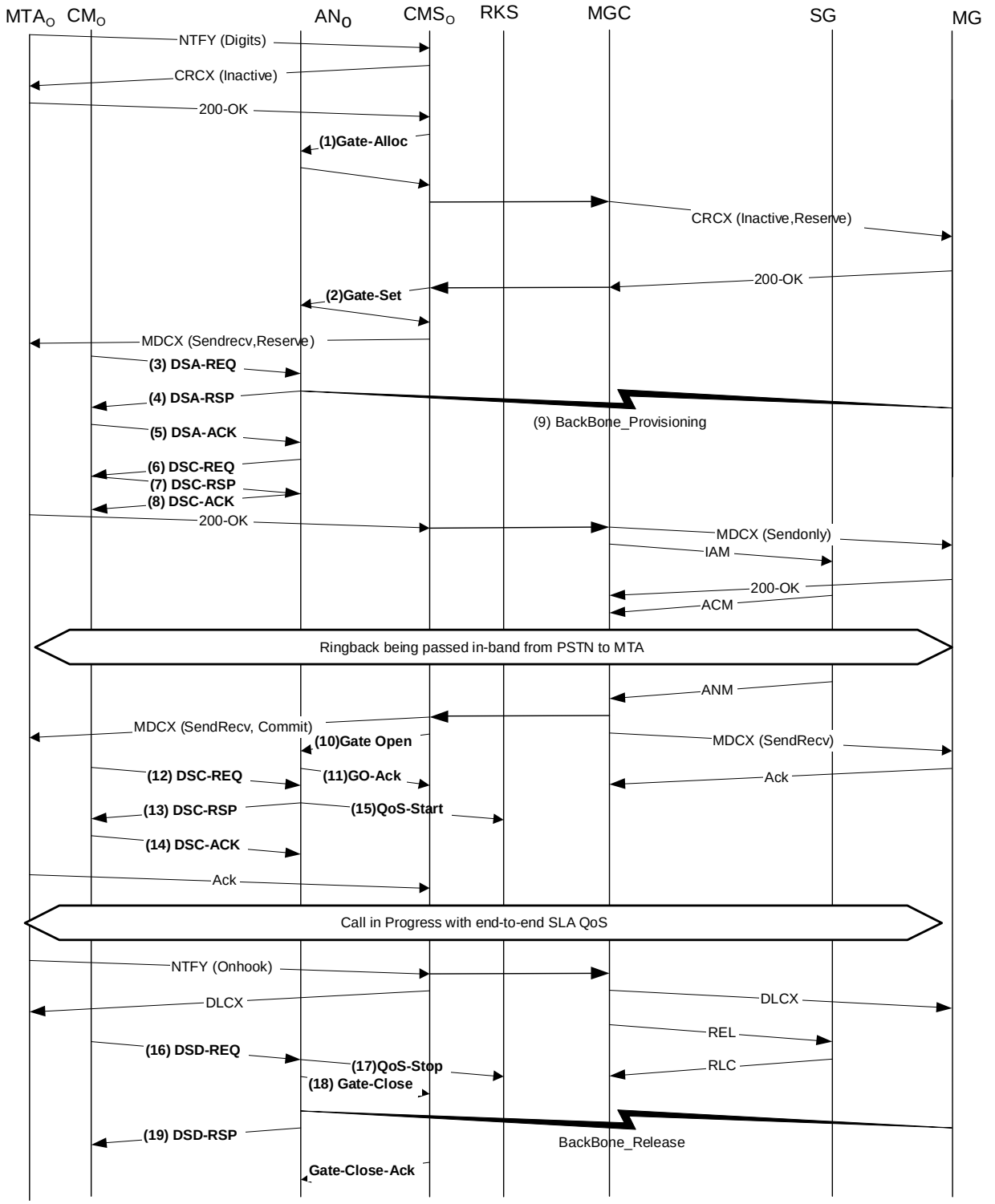


Figure 44: On-net to Off-net Embedded NCS

1) CMSo, upon receipt of signalling information from MTAo, checks the current resource consumption of MTAo by consulting ANo.

GATE-ALLOC

TransactionID		3 176	
Subscriber		MTAo	Request for total resources in use by this client.
Activity-Count		4	Maximum connections allowed by client.

ANo checks current resource usage by MTAo, and responds telling the number of connections active.

GATE-ALLOC-ACK

TransactionID		3 176	
Subscriber		MTAo	Request for total resources in use by this client.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		3	Total connections established by this client.

2) CMSo, upon further signalling exchanges, gives ANo authorization to admit the new connection.

GATE-SET

Transaction ID		3 177	Unique Transaction ID for this message exchange.
Subscriber		MTAo	Request for total resources in use by this client.
Gate-ID		37 125	Identifier for allocated Gate.
Remote-Gate-Info	AN Address	CMSo	Information needed to perform gate co-ordination. Note that CMS has given itself as the entity for exchanging gate co-ordination messages.
	AN Port	2 052	
	Remote Gate-ID	8 095	
	Security Key	<key>	
	Flag	No-gate-open	
Event-Generation-Info	RKS-Addr	RKS	Address of Record Keeping Server.
	RKS-Port	3 288	Port on Record Keeping Server.
	Billing Correlation ID	<id>	Opaque data that will be passed to RKS when resources are committed.

GATE-SET

Gate-Spec	Direction	up	
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAo	
	Destination Address	MTAt	
	Source port	0	
	Destination port	7 000	
	DSCP	6	Packet Type value for upstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

GATE-SET

Gate-Spec	Direction	down	
	Flag	Auto-commit	Flag to activate resources on the Reserve operation.
	Protocol	UDP	The protocol, Destination Address, Source Address, and Destination Port quadruple are used for QoS classifiers.
	Source Address	MTAt	
	Destination Address	MTAo	
	Source port	0	
	Destination port	7 120	
	DSCP	9	Packet Type value for downstream packets.
	T1	180 000	Maximum time between reserve and commit.
	T2	2 000	Maximum time for gate co-ordination to complete.
	b	120	These are the maximum bandwidth parameters that MTAo is authorized to request for this conversation.
	r	12 000	
	p	12 000	
	m	120	
	M	120	
	R	12 000	
	S	0	

ANo responds to the Gate Setup command with an acknowledgement.

GATE-SET-ACK

TransactionID		3 177	
Subscriber		MTAo	Request for total resources in use by this client.
Gate-ID		37 125	Identifier for allocated Gate.
Activity Count		4	Total connections established by this client.

3) MTAo, upon receiving call signalling information, calculates the QoS parameters for the J.112 link. It uses the Appendix E interface to the CM to send the following DSA-REQ to the AN. This message is used to establish both upstream and downstream parameters. The Upstream Unsolicited Grant Size was calculated as 120 (from SDP) plus 18 (Ethernet overhead) minus 40 (Header Suppression amount) plus 13 (J.112 overhead). Header suppression indicates the 42 bytes of Ethernet/IP/UDP header. Contents of the suppressed header is included in the DSA-REQ.

DSA-REQ

TransactionID		1
UpstreamServiceFlow	ServiceFlowReference	1
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	ServiceFlowScheduling	UGS (6)
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111
	AuthBlock	37 125
DownstreamServiceFlow	ServiceFlowReference	2
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	TrafficPriority	5
	MaximumSustainedRate	12 000
	AuthBlock	37 125
UpstreamPacketClassification	ServiceFlowReference	1
	PacketClassifierReference	1
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAo
	IPSourcePort	7 120
	IPDestinationAddress	MGt
	IPDestinationPort	7 000
	IPProtocol	UDP (17)

DSA-REQ

DownstreamPacketClassification	ServiceFlowReference	2
	PacketClassifierReference	2
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MGt
	IPDestinationAddress	MTAo
	IPDestinationPort	7 124
	IPProtocol	UDP (17)
PayloadHeaderSuppression	ClassifierReference	1
	ServiceFlowReference	1
	HeaderSuppressionIndex	1
	HeaderSuppressionField	<42bytes>
	HeaderSuppressionMask	<42bits>
	HeaderSuppressionSize	42
	HeaderSuppressionVerify	Verify (0)
AuthorizationBlock		37 125
HMAC		

4) The AN checks the authorization, by looking for a gate with gate-ID matching the value in AuthBlock, and checks the resources it is required to allocate (e.g. header suppression table space, Service Flow IDs, classifier table space), and installs the classifiers. If the operation is successful it returns the DSA-RSP message stating the success.

DSA-RSP

TransactionID		1
ConfirmationCode		Success (0)
UpstreamServiceFlow	ServiceFlowReference	1
	ServiceFlowIdentifier	1 001
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	ServiceFlowScheduling	UGS (6)
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111
	AuthBlock	37 125
DownstreamServiceFlow	ServiceFlowReference	2
	ServiceFlowIdentifier	2 001
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	TrafficPriority	5
	MaximumSustainedRate	12 000
	AuthBlock	37 125
UpstreamPacketClassification	ServiceFlowReference	1
	PacketClassifierReference	1
	PacketClassifierIdentifier	3 001
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAo
	IPSourcePort	7 120
	IPDestinationAddress	MGt
	IPDestinationPort	7 000
	IPProtocol	UDP (17)

DSA-RSP

DownstreamPacketClassification	ServiceFlowReference	2
	PacketClassifierReference	2
	PacketClassifierIdentifier	3 002
	ClassifierPriority	150
	ClassifierActivationState	Active (1)
	IPSourceAddress	MGt
	IPDestinationAddress	MTAo
	IPDestinationPort	7 124
	IPProtocol	UDP (17)
HMAC		

5) Upon receipt of the DSA-RSP, the CM acknowledges receipt with a DSA-ACK message.

DSA-ACK

TransactionID		1
ConfirmationCode		Success (0)
HMAC		

6) Upon receipt of the DSA-ACK from the CM, the AN sends a DSC-REQ message to the CM to activate the resources for the downstream service flow. The AN does this because the Auto-commit flag is enabled in the GATE-SET from the CMS for the downstream gate.

DSC-REQ

TransactionID		2
UpstreamServiceFlow	ServiceFlowIdentifier	1 001
	QoSParameterSetType	Admitted (2)
	TimeOutAdmitted	200
	ServiceFlowScheduling	UGS (6)
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111

DSC-REQ

DownstreamServiceFlow	ServiceFlowIdentifier	2 001
	QoSParameterSetType	Admitted + Activated (6)
	TimeOutActive	10
	TrafficPriority	5
	MaximumSustainedRate	12 000
Upstream Classifier	ServiceFlowIdentifier	1 001
	PacketClassifierIdentifier	3 001
	ClassifierChangeAction	Replace (1)
	ClassifierPriority	150
	ClassifierActivationState	Inactive (0)
	IPSourceAddress	MTAo
	IPSourcePort	7 120
	IPDestinationAddress	MGt
	IPDestinationPort	7 000
	IPProtocol	UDP (17)
Downstream Classifier	ServiceFlowIdentifier	2 001
	PacketClassifierIdentifier	3 002
	ClassifierChangeAction	Replace (1)
	ClassifierPriority	150
	ClassifierActivationState	Active (1)
	IPSourceAddress	MGt
	IPDestinationAddress	MTAo
	IPDestinationPort	7 124
	IPProtocol	UDP (17)
HMAC		

7) Upon receipt of the DSC-REQ from the AN, the CM sends a DSC-RSP to the AN.

DSC-RSP

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

8) Upon receipt of the DSC-RSP from the CM, the AN sends a DSC-ACK to the CM.

DSC-ACK

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

9) Simultaneous with message No. 4, the AN initiates any required backbone reservations for the requested quality of service. The content of this message is dependent on the particular backbone algorithms in use, and is outside the scope of the present document. The backbone router sends to the AN any required notification that the reservation is successful.

10) The CMS sends the gate open message to the AN to inform it that the resources should be committed. If the AN does not receive the DSC-REQ from MTAO within a short time, it should revoke the gate authorization.

GATE-OPEN

Transaction ID		72	Identifier to match this message with its response.
Gate ID		37 125	Gate-ID at AN.
HMAC			Security checksum for this message.

11) The AN responds to the GATE-OPEN with:

GATE-OPEN-ACK

Transaction ID		72	Identifier to match this message with its response.
HMAC			Security checksum for this message.

12) In response to signalling messages that indicate the call has completed (i.e. the other side has gone off-hook), MTAo uses the Appendix E interface to activate the admitted resources. This is done via a DSC-REQ command to the AN.

DSC-REQ

TransactionID		2
UpstreamServiceFlow	ServiceFlowIdentifier	1 001
	QoSParameterSetType	Admitted + Activated (6)
	TimeOutActive	10
	ServiceFlowScheduling	UGS (6)
	NominalGrantInterval	10 ms
	ToleratedGrantJitter	2 ms
	GrantsPerInterval	1
	UnsolicitedGrantSize	111
DownstreamServiceFlow	ServiceFlowIdentifier	2 001
	QoSParameterSetType	Admitted + Activated (6)
	TimeOutActive	10
	TrafficPriority	5
	MaximumSustainedRate	12 000
UpstreamPacketClassification	ServiceFlowIdentifier	1 001
	PacketClassifierIdentifier	3 001
	ClassifierChangeAction	Replace (1)
	ClassifierPriority	150
	ClassifierActivationState	Active (1)
	IPSourceAddress	MTAo
	IPSourcePort	7 120
	IPDestinationAddress	MGt
	IPDestinationPort	7 000
	IPProtocol	UDP (17)

DSC-REQ

DownstreamPacketClassification	ServiceFlowIdentifier	2 001
	PacketClassifierIdentifier	3 002
	ClassifierChangeAction	Replace (1)
	ClassifierPriority	150
	ClassifierActivationState	Active (1)
	IPSourceAddress	MGt
	IPDestinationAddress	MTAo
	IPDestinationPort	7 124
	IPProtocol	UDP (17)
HMAC		

13) The AN sends a DSC-RSP message showing the operation was successful.

DSC-RSP

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

14) The CM sends DSC-ACK message to indicate that the DSC-RSP has been received and agreed.

DSC-ACK

TransactionID		2
ConfirmationCode		Success (0)
HMAC		

15) ANo sends the event record to the Record Keeping Server that a Commit has been received on this call. This message is only a sample of what might be included in a QoS-Start message:

QoS-START

Header	Timestamp	<time>	Time of the event being recorded.
	Billing Correlation ID	<string>	Correlation ID given in Gate-Set.
QoS Descriptor	Type	UGS	Description of the QoS provided for this connection.
	Grant interval	10 ms	
	Grant Jitter	2 ms	
	Grant/Interval	1	
	Grant Size	111	
MTA Port	Port	7 120	

16) When the call is finished the MTA uses the Appendix E interface to delete the Service Flows, sending a DSD-REQ message to the AN.

DSD-REQ

TransactionID		3
ServiceFlowID		1 001
HMAC		

DSD-REQ

TransactionID		4
ServiceFlowID		2 001
HMAC		

17) The AN sends the notification to the Record Keeping Server that the call has ended. This message is only a sample of what might be included in a QoS-Stop message:

QoS-Stop

TimeStamp		<time>	The time of the event being recorded.
Header	Time Stamp	<time>	Time of the event being recorded.
	Billing Correlation ID	<string>	Correlation ID from Gate-Set message.
SF-ID	SF-ID	1 001	Service Flow Identifier.

18) The AN, upon receiving RSVP-PATH-TEAR, sends the gate co-ordination message to the CMS (identified in the Gate-Set message).

GATE-CLOSE

Transaction ID		73	Identifier to match this message with its response.
Gate-ID		8 095	This identifies the GateID at the CMS.
HMAC			Security checksum for this message.

The CMS responds with:

GATE-CLOSE-ACK

Transaction ID		73	Identifier to match this message with its response.
HMAC			Security checksum for this message.

19) The AN deletes the Service Flow IDs and sends the response to the CM.

DSD-RSP

TransactionID		3
ServiceFlowID		1 001
ConfirmationCode		Success (0)
HMAC		

DSD-RSP

TransactionID		4
ServiceFlowID		2 001
ConfirmationCode		Success (0)
HMAC		

Annex L (informative): Theft of Service Scenarios

We outline here several possible theft of service scenarios to highlight the need for the dynamic authorization, the need for the 2-phase resource reservation protocol, the need for gates, and the need for gate co-ordination. The system design places much of the session control intelligence at the clients, where it can easily scale with technology and provide new and innovative services. While this "future-proofing" is a goal of the design, we must recognize that it leaves open a wide range of fraud possibilities. This annex discusses some of those possibilities, and how the QoS signalling architecture prevents them.

The basic assumption is that the MTA is not immune to customer tampering, and that the significant incentive for free service will lead to some very sophisticated attempts to thwart any network controls placed on the MTA. This customer tampering includes, but is not limited to, opening the box and replacing ROMs, replacing integrated circuit chips, probing and reverse engineering of the MTA design, and even total replacement of the MTA with a special black-market version. While technical solutions exist to the physical security of the MTA (e.g. booby trapping the box with lethal gas), they are not considered acceptable.

Since the MTA can be distinguished only by its communication over a J.112 network, it is possible, and quite likely, that PC software will be written that will emulate the behaviour of a MTA. Such a PC may be indistinguishable from a real MTA. The software behaviour in this case is under the total control of the customer.

Further, it is intended that new services will be implemented in the MTA, and that software control of those new services will be provided by a variety of vendors. This updated software will be downloaded into the MTA, leaving open the possibility of customers downloading special hacked versions that provide free service. We do not concern ourselves here with the problem of "trojan horses" in such downloaded software, as this is considered identical to the problem today of customers giving away their credit card numbers and/or PINs. We are concerned with the customer intentionally downloading special software that does only what is in his/her best interest.

L.1 Scenario No. 1: Customers establishing high QoS Connections themselves

The MTA, with sufficient intelligence, can remember past destinations dialled and the destination address, or use some other mechanism to determine the IP address of a destination. It can then signal that destination itself (with some co-operation of the other client), and negotiate a high quality-of-service connection via the RSVP mechanism or via the Appendix E interface for an embedded client. Since no network agent is used in initiating the session, there will be no billing record produced. Prevention of this scenario is done by requiring dynamic authorization at the AN; without the authorization the attempt to obtain the high quality-of-service will fail.

The above scenario required the co-operation of two altered MTAs. Similar theft of service could be accomplished with only the originator being modified. If the originating MTA used the network agent to establish the session, thereby informing the destination in the standard manner of an incoming session, but again negotiated the high quality-of-service itself, there would be no billing record generated and the originator would get a free session. Again, the solution is to require the use of gates in the ANs.

L.2 Scenario No. 2: Customers using provisioned QoS for non-voice applications

Statically provisioned QoS can only identify a customer as one who is authorized for high Quality of Service. There is no restriction on the usage of the service. In particular, a customer who has subscribed for a commercial-grade voice communications service, and is therefore authorized to activate high-bandwidth low-latency connections through the J.112 network, can use this ability for web surfing or other PC applications. Prevention of this scenario is done by requiring dynamic authorization at the AN; without the authorization the attempt to obtain the high quality-of-service will fail.

L.3 Scenario No. 3: MTA non-co-operation for billing

One can easily imagine what would happen if there was a message from the MTA on session establishment that said, "OK, callee has answered, start billing me now," or a message on hangup that said, "session has completed, stop billing now." However, there are more subtle ways that a user could have the same affect as tinkering with such messages if they existed.

It is essential in providing a commercial-grade voice communications service using IPCablecom to ensure network capacity exists prior to signalling the CPE at the receiving party's location. This function is done with the RESERVE messages. If the RESERVE message were to actually allocate the bandwidth (i.e. combining the RESERVE and COMMIT mechanisms), then there would be no incentive for the MTA to ever issue the COMMIT. The MTA could merely start transmitting voice packets immediately, and the destination could start transmitting voice packets as soon as the phone is answered. The COMMIT message becomes, in effect, the billing start message above. It is therefore essential that the RESERVE not actually allocate the bandwidth, but rather it must check all current allocations and pending reservations to ensure that the bandwidth will be available at the time of a COMMIT message.

L.4 Scenario No. 4: MTA altering the destination address in voice packets

Another example is when two MTAs, which are far apart, each establish a local session. Once the bandwidth and connection are established, the MTAs then change the IP addresses in the RTP streams to point to each other. The billing system continues to bill each of them for a local session, while the customers are actually engaged in a long distance session. This requires us to have mechanisms at the ANs that provide access to higher QoS only based on packet filters previously authorized. Thus, in addition to the 2-phase resource management, this scenario motivates the need for packet filters at the gates.

L.5 Scenario No. 5: Use of half-connections

This is an example of theft of service that could occur in the absence of gate co-ordination. Suppose one client in a session sends a COMMIT message and the other does not. For example, say the terminating client sends a COMMIT, but fails to send the proper signalling message, so the originator never sends a COMMIT. In this case, only one gate is opened, and the users and network are left with a half-connection. Given that the originator did not send a COMMIT message, the network cannot legitimately bill the user for the half-connection. However, it is possible for two colluding clients to set up two half-connections, neither of which is billable, which can be combined to give a full connection between the two parties. This results in a free session. Fraud of this type can only be prevented by synchronizing the operation of the two gates.

L.6 Scenario No. 6: Early termination leaving a half-connection

Gate co-ordination is also required on completion of the session. Suppose that MTA_O calls MTA_T and pays for the session. Since MTA_O is being charged for the session, it clearly has an incentive to issue a RELEASE message to AN_O to close its gate and stop the billing. However, if MTA_T does not issue the RELEASE message to close the gate at AN_T, a half-connection remains. In this case MTA_T can continue to send voice and/or data to MTA_O without billing for the session. Hence, a GATE-CLOSE message must be issued from the originating side gate at AN_O to close the terminating side gate at AN_T.

L.7 Scenario No. 7: Forged Gate Co-ordination messages

Each MTA knows the identity of its AN, and knows the 5-tuple that its AN uses to identify the GateID. MTAs can do various kinds of end-to-end negotiation before asking for resources; in particular they can easily exchange the information about their GateID. Then the MTA can fake the GATE-OPEN message being sent to the non-paying end, and get a non-billed one-way connection. Doing this twice gets a full non-billed connection. One solution to this problem is for the GateController to give the AN a key to use for the AN-AN messages, on a per-session (or per-gate) basis.

L.8 Scenario No. 8: Fraud directed against unwanted callers

Due to details of the call setup sequence, it is possible that the bandwidth authorization at the destination will be more generous than that at the source. Given this, it is then possible for a called party to reserve and allocate bandwidth far in excess of the final negotiated amount, resulting in the calling party being charged for more than expected. If available, this would likely be used against telemarketers, fighting back for unwanted calls during dinner.

Gate co-ordination, which was used previously to guard against half-connections, also protects from this type of fraud. The GATE-OPEN message tells the bandwidth that was allocated as a result of the COMMIT, and the COMMIT-ACK sent to the originator tells exactly what bandwidth will be charged for the session. If the originator detects anything amiss, he can immediately terminate the session.

Annex M (informative): COPS (Common Open Policy Service)

M.1 COPS Procedures and Principles

This annex provides a brief description of COPS procedures and principles, and how COPS relates to other protocols such as LDAP. COPS is currently defined in an Internet draft-ietf-rap-cops-07 (see Bibliography).

Common Open Policy Service (COPS) protocol is a client/server protocol being defined in the IETF RSVP admission policy (rap) working group for use in admission control in RSVP/IntServ and DiffServ QoS networks. COPS runs over TCP/IP, using a well-known port number 3 288. COPS entities would reside at a network edge device and a policy server. Three functional entities are defined for the rap framework:

- Policy Decision Point (PDP) - the server entity in COPS, which makes the final decision on session admission or rejection, based on policy information that it has access to. This is expected to be implemented as an application on a standalone server device.
- Policy Enforcement Point (PEP) - the client entity in COPS, which consults with the PDP to make policy decisions or to obtain policy information that it may itself use to make admission control decisions; the PEP may receive requests for service and initiate a query to the PDP that will result in a go/no-go response, or the PEP may inform the PDP that it wishes to receive decisions and policy related information on an unsolicited basis.
- Local Decision Point (LDP) - a local version of the PDP that can make decisions based on local information or information cached from previous decisions. A PDP decision always takes precedence over the LDP.

A COPS sequence, as used in an RSVP/IntServ environment, is shown below.

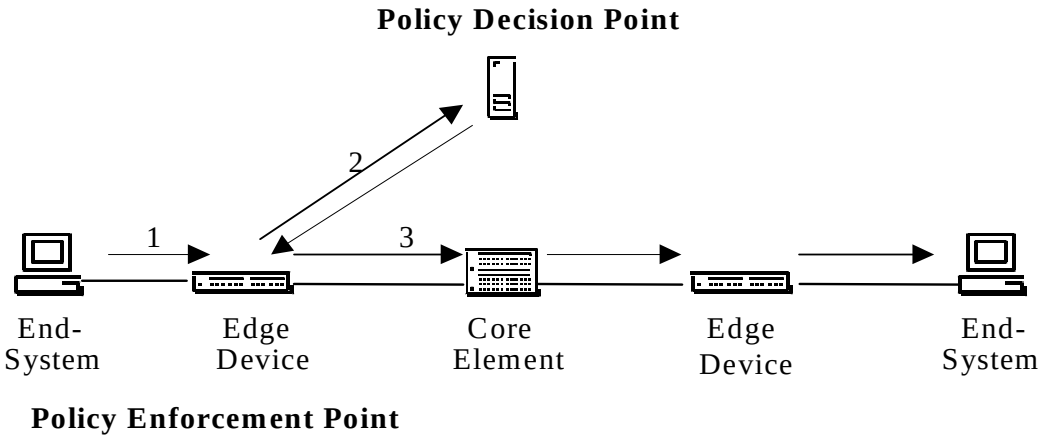


Figure 45: COPS Protocol

In the COPS sequence, the client PEP is responsible for initially establishing a session with the PDP, using information that is either configured in the PEP or determined by some other means. Once the session is established, if the Edge Device receives an RSVP message (1), it generates a request for handling to the PDP (2) that describes the context of the request and carries information about the request. The PDP then responds (3) with a decision to accept or reject the request, and if it is accepted, the Edge Device continues by forwarding the RSVP message out into the network (4).

Each session is maintained by a Keep Alive message that verifies that the session is active in case no message has been received recently. Each RSVP or other request is identified by a Handle, which can be used to associate the response, subsequent unsolicited responses, and clearing.

The protocol messages are extensible to other tasks. They consist of an Op Code identifying if the message is a Request, Response, or other type, followed by self-identifying objects, each containing an object class and version identifier. Each object includes a Class Number that defines what the object is, for example, a Timer object, or a Decision object, plus a Class Type that identifies the subtype or version of the Class that is being used.

Other object classes include Bandwidth allocation Data needed for identifying the resources requested by the user, and Policy objects that can be passed down from the PDP to be included in the RSVP message when it is sent out into the network.

Comparison of COPS and LDAP for Policy

Both COPS and LDAP have been associated with Policy-Based Management, however, they would provide very different functions.

COPS is designed for the client to request a decision from a Policy Decision Point and to interact with the PDP to actively participate in the management of policy and policy-related issues. The PEP that makes the request may have no actual knowledge of policies, and relies on the PDP to make decisions based on its knowledge of policies. The protocol allows the PEP to pass information about the request to the PDP, and the PDP to pass back a decision to allow or reject the request.

LDAP is designed for the client to request a directory record from a directory. The function for using the record is dependent on the client, which must be capable of understanding the retrieved record and deciding how to use the information. The server must be capable of finding the correct record based on information in the request, which may involve a search function, or retrieval of multiple records.

Both COPS and LDAP could be used in the context of RSVP admission control. COPS would be used between the PEP and PDP to forward a request for policy-based analysis. LDAP would be used between the PDP and a Directory Server to retrieve policy records associated with the originating and destination addresses for the RSVP request. The PDP would then make a decision based on the retrieved policy information, and use COPS to pass that decision back to the PEP.

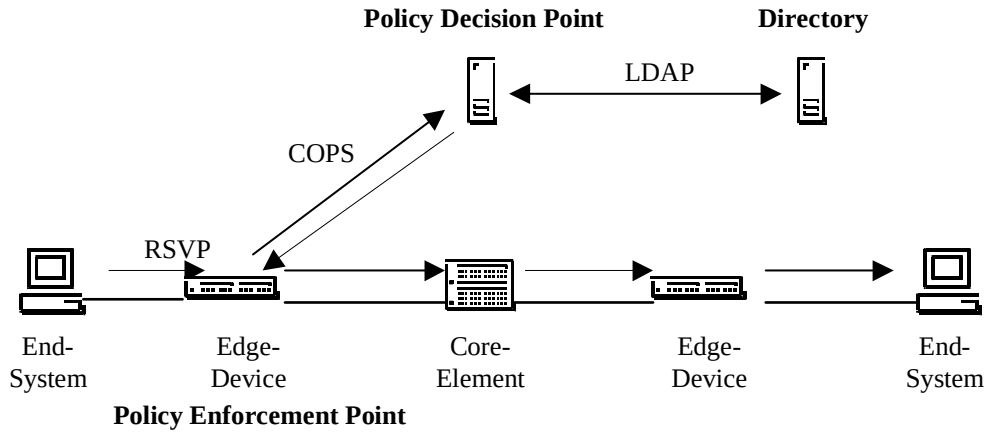


Figure 46: COPS and LDAP model

Annex N (informative): RSVP (Resource Reservation Protocol)

N.1 RSVP Procedures and Principles

This annex provides a brief description of RSVP procedures and principles. RSVP is currently defined in RFC 2205.

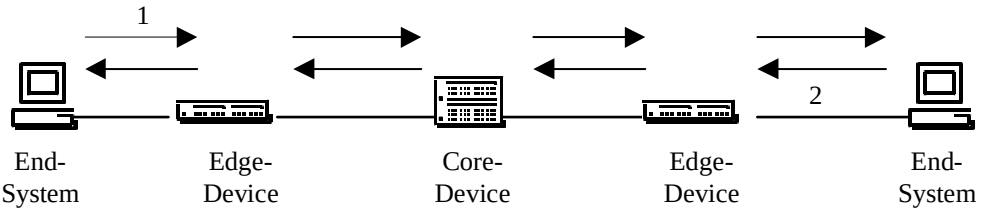


Figure 47: RSVP

RSVP was developed in the IETF for resource reservation to support information flows across the Internet. Some of the main characteristics of RSVP are:

- resource reservation hop by hop to support end-to-end information flows;
- state information kept at every participating router;
- non-participating routers treat RSVP messages like normal IP packets;
- soft state - reservation must be refreshed periodically or it automatically cancels;
- request driven - an initial PATH message establishes state in the router. A RESV message from the receiver actually results in reservation of resources.

In RSVP, the source initiates a session by sending out a PATH message (1). This is routed through the network based on its destination address (which may be multicast) and creates a flow state at every RSVP-supporting router that it passes through. The PATH message is routed using the same procedures as other IP packets with that destination address, so that it duplicates the route to be followed by data packets. As it progresses, it records the address of the last RSVP-capable router, and this is added to the state information at the next router.

At the receiving end, the receiver joins the session by sending out a RESV message (2) that identifies a flow or flows that this receiver wishes to receive out of the different flows supported in the session. The RESV message traces back the sequence followed by the PATH message, using the records of the last RSVP-capable router, and causes resources to be reserved at each hop. If multiple RESV messages are received at the same router, they may be merged into a singe RESV message with combined resource reservation request.

The process requires state establishment at multiple internal nodes and resource reservation at the same nodes. It establishes a fixed path for the information flow. However, it ensures that resources have been allocated at all RSVP-supporting points in the path.

RSVP flowspec

An elementary RSVP reservation request consists of a "flowspec" together with a "Filter-Spec"; this pair is called a "flow descriptor". The flowspec specifies a desired QoS. The filterspec, together with a session specification, defines the set of data packets - the "flow" - to receive the QoS defined by the flowspec. The flowspec is used to set parameters in the node's packet scheduler or other link layer mechanism, while the Filter-Spec is used to set parameters in the packet classifier. Data packets that are addressed to a particular session but do not match any of the Filter-Specs for that session are handled as best effort traffic.

The flowspec in a reservation request will generally include a service class and two sets of numeric parameters: (1) an "Rspec" (R for "reserve") that defines the desired QoS, and (2) a "Tspec" (T for "traffic") that describes the data flow.

It is important to note that the formats and contents of Tspecs and Rspecs are determined by the integrated service models RFC 2210 defined in the intserv working group of the IETF, and are generally opaque to RSVP itself. RSVP defines the signalling mechanism, and not the traffic model.

Annex O (informative): TCP Considerations

The present document defines an interface between a Gate Controller (GC) and an Access Node (AN) to be used for gate authorization, which basically supports a transaction based protocol where each transaction is independent. TCP may be used as a transport for this messaging. However, there were concerns raised about the performance implications of using TCP. This annex examines a few of these concerns and proposes some potential solutions that can provide an acceptable transport through implementation optimizations and tuning of the TCP implementation.

The design of the network should support the desired degree of reliability and real time performance.

O.1 Requirements

Let us first consider requirements on the transport protocol for the interaction between the GC and AN:

1) Reliable message delivery for messages exchanged between the GC and AN is required.

2) The message exchange should have low latency (of the order of milliseconds), in the normal case (without packet loss). We also need it to have reasonably low latency even under packet loss (of the order of tens of milliseconds).

3) We want multiple requests to be outstanding concurrently. This is because multiple call set-ups are likely to be in progress concurrently.

4) If there is likely to be head-of-the-line (HOL) blocking, this should be avoided.

5) There is likely to be a long-standing association (at least of the order of several minutes) between the GC and the AN. However, when there is a failure of a GC, the process of establishing a new connection to the AN should not take excessive time. This is especially true when the establishment of a new connection occurs during the time that a call is being setup.

O.2 Recommended Changes

Briefly, the changes we recommend to a vanilla TCP implementation are the following:

1) Modify the time-out mechanism for connection establishment (make it more aggressive).

2) Allow for a larger window after connection establishment.

3) Have multiple TCP connections per GC-AN pair to work around potential HOL problems (e.g. use them on a round-robin basis).

4) Lower the 500 ms granularity of the time-out.

5) Disable Nagle's algorithm on the transmit end so as to reduce the latency.

6) Have a non-blocking interface between the application and the TCP stack.

The remainder of this annex gives details of how these may be implemented.

O.3 TCP Connection Establishment impacting Post-dial Delay

TCP connection establishment uses a three-way handshake as follows:

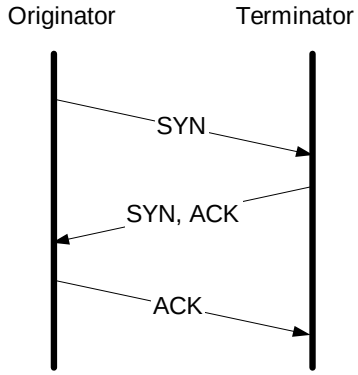


Figure 48: TCP Connection Establishment

TCP retransmits segments assumed to be lost based on a round-trip time estimate, A, and a mean deviation, D, from A. The retransmission timeout value (RTO) is generally calculated using the formula:

$$RTO = A + 4D$$

but the initial RTO is calculated using the formula:

$$RTO = A + 2D$$

where A and D are initialized to 0 and 3 s respectively. When a retransmission occurs, an exponential backoff using a multiplier of 2 is applied to the current RTO value. Thus, for the first segment, the RTO is calculated as

$$RTO = 0 + 2 \times 3 = 6$$

Thus, if the initial SYN segment is lost, a retransmission will not occur until 6 s later. At that time, RTO will be calculated as:

$$RTO = 0 + 4 \times 3 = 12$$

and an exponential backoff of 2 applied, leading to a new retransmission timeout value of 24 s. Thus, should the retransmission be lost as well, a total of 30 s will have elapsed before the third retransmission occurs.

The importance of this problem entirely depends on the frequency with which GC->AN connection establishment falls during the post-dial-delay period. In the currently envisioned scenarios, this occurrence should be very much the exception rather than the rule. The connection setup delay impacting the post-dial delay is an important reason to avoid having connection establishment in the post-dial-delay period. Diffserv marking of the packets to reduce both latency and loss probability, analogous to what is done with routing traffic today, could be used to reduce connection setup delays due to lost packets.

O.4 Need Low Latency for packets between the GC and AN, even under loss

Requirement (2), which deals with recovery of packet loss needs a few remedies available for TCP to recover from loss quickly. When there are only a few packets being transmitted, and the receiver is unable to generate a sufficient number of duplicate ACKs, the recovery from packet loss is from a retransmission time-out. The TCP retransmission algorithm is based on a smoothed average of the observed round-trip time (RTT), A, and a smoothed average of the mean deviation in RTT. As described above, the retransmission time-out value is then set to:

RTO = A + 4D

and if the timer fires, the segment in question is retransmitted, and RTO is backed off exponentially using a multiplier of 2 (see note) until an upper limit of 64 s for RTO. Once a segment has been passed to TCP, the segment is either transmitted successfully to the destination or the connection is closed after some period of time has passed (generally a large period of time, e.g. 2 to 9 minutes).

NOTE: TCP furthermore uses duplicate ACKs to trigger retransmission of potentially lost segments, however we will ignore that for this part of the discussion.

While the above retransmission strategy is deemed desirable, we believe it has two (related) problems for the interface considered:

1) If the segment is not delivered successfully within a small period of time, the call that is in the process of being set up will most likely be abandoned and the transaction should therefore be able to be aborted.

2) The 64-second cap on the retransmission timeout is ill-suited for real time communication and should be set much lower.

A separate, but related issue is that of the granularity of RTO. While the TCP specification itself does not specify the granularity of RTO, it is very common to have a granularity of 500 ms in commercial operating systems. Thus, a lost segment will generally not be detected within less than 500 ms, and two lost segments will not be detected within less than 500 ms + 1 000 ms = 1,5 s.

To recover rapidly from packet loss in a sequence of packets (without having to depend on multiple duplicate ACKs to trigger fast retransmit or having to wait for the RTO timer to fire), it may be desirable to implement TCP-SACK, which aids recovery even if the fast-retransmit threshold is not reached. It is also recommended that the TCP implementation use a smaller timer granularity (possibly less than 500 ms).

O.5 Head of Line Blocking

Head of line blocking refers to the fact, that TCP provides an in-order data delivery service where a lost segment can block later segments from being delivered to the application. Thus, if segments 1 and 2 are sent from A to B, and segment 1 is lost, segment 2 cannot be delivered to the application until segment 1 has been successfully retransmitted.

For the interface considered, this head of line blocking can probably be overcome reasonably well by having multiple TCP connections established between the GC and AN, and then use the set of TCP connections in e.g. a round-robin fashion for the transactions. Thus, if a segment is lost on one connection, it will not affect segments, i.e. transactions sent on other connections.

The downside to this approach is that a lost segment is not likely to be retransmitted until its retransmission timer fires (as opposed to a duplicate ACK being received), since there should not be any additional segments to transmit until then.

O.6 TCP Slow Start

TCP's ability to start transmitting a stream of data packets is sometimes limited by the TCP slow start mechanism, especially when the stream is a small number (greater than 1) of data packets. It is desirable to choose an initial window that is larger than 1 (both at the beginning of the life of the connection as well as after congestion recovery from a single packet loss). Choosing an initial window size of 2 to 4 MSS is considered desirable. It is important however to ensure that this initial window not exceed 4 MSS, because of the potential to cause congestion itself.

O.7 Delaying of packets: Nagle's Algorithm

TCP/IP was originally designed for supporting multiple user sessions over a slow network. In order to optimize network utilization, the Nagle algorithm was introduced for keyboard input users. Essentially, this algorithm delays the transmission of a packet until a sufficiently large transmit buffer is accumulated or until a certain period of time (usually around 200 ms) elapses.

Due to the real time nature of this traffic, it is advisable to disable the Nagle algorithm for GC-AN communication. On most Unix based platforms, Nagle's algorithm can be disabled by issuing the following system call on the socket's file descriptor:

EXAMPLE 1: Setting the TCP_NODELAY Option

```
/* set TCP No-delay flag (disable Nagle algorithm) */

int flag = 1;

setsockopt(fd, IPPROTO_TCP, TCP_NODELAY, &flag,
sizeof(flag));
```

Most other languages and platforms have a similar feature to disable the Nagle algorithm, usually known as the TCP_NODELAY option.

O.8 Non-Blocking Interface

By default, most operating systems provide a blocking interface for TCP/IP sockets. Although it may allow for an improved error recovery scheme, it has an impact on the performance of the communication channel.

Essentially, a system call such as send() with blocking interface never returns until the operating system confirms that the message was successfully stored in the transmit buffer.

It may be desirable to use a non-blocking interface in order to improve performance and to support asynchronous events using the select() function call on a UNIX based architecture. A non-blocking socket interface can be setup by using the following call on the newly created socket.

EXAMPLE 2: Setting the O_NONBLOCK Option

```
/* set the socket to non blocking */

fcntl( fd, F_SETFL, O_NONBLOCK );
```

Most other languages and platforms have a similar feature.

Annex P (informative): Bibliography

IEEE 1394: "IEEE Standard for a High Performance Serial Bus".

RFC 791 (1981): "Internet Protocol".

RFC 2139 (1997): "RADIUS Accounting".

RFC 2749 (2000): "COPS usage for RSVP".

RFC 2750 (2000): "RSVP Extensions for Policy Control.

RFC 2753 (2000): "A Framework for Policy Based Admission Control".

ITU-T Recommendation G.114: "One-way transmission time".

ITU-T Recommendation G.721: "32 kbit/s adaptive differential pulse code modulation (ADPCM)".

ITU-T Recommendation G.729, annex E: "11.8 kbit/s CS-ACELP speech coding algorithm".

ITU-T Recommendation G.728: "Coding of speech at 16 kbit/s using low-delay code excited linear prediction".

ITU-T Recommendation J.160: "Architectural framework for the delivery of time critical services over cable Television networks using cable modems".

ITU-T Recommendation J.161: "Audio codec requirements for the provision of bidirectional audio service over cable television networks using cable modems".

ITU-T Recommendation J.162: "Network call signalling protocol for the delivery of time critical services over cable television networks using cable modems".

ITU-T Recommendation J.em: "Event Message requirements for the support of real time services over cable television networks using cable modems".

ITU-T Recommendation J.sec: "IPCablecom Security specification".

PacketCable Distributed Call Signalling Specifications, at http://www.packetcable.com/specifications.html.

Draft-ietf-rsvp-refresh-reduct-05 (April 2001): "RSVP Refresh Overhead Reduction Extensions".

Draft-ietf-mpls-rsvp-lsp-tunnel-08 (February 2001): "RSVP-TE: Extensions to RSVP for LSP Tunnels".

Draft-ietf-rap-pr-05 (March 2001): "COPS Usage for Policy Provisioning (COPS-PR)".

NOTE: See http://www.ietf.org/internet-drafts/.

Draft-ietf-rap-cops-07 on the COPS (Common Open Policy Service) Protocol.

History

Document history		
V1.1.1	August 2001	Publication