

International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.365

(11/2006)

SERIES J: CABLE NETWORKS AND TRANSMISSION
OF TELEVISION, SOUND PROGRAMME AND OTHER
MULTIMEDIA SIGNALS

IPCablecom

IPCablecom2 application manager interface

ITU-T Recommendation J.365



ITU-T Recommendation J.365

IPCablecom2 application manager interface

Summary

This Recommendation defines an IPCablecom2 Application Manager (IPAM) interface which allows the Proxy Call Session Control Function (P-CSCF) to support Quality of Service (QoS) operations within an IPCablecom Multimedia-enabled DOCSIS-access network.

Source

ITU-T Recommendation J.365 was approved on 29 November 2006 by ITU-T Study Group 9 (2005-2008) under the ITU-T Recommendation A.8 procedure.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2007

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

CONTENTS

	Page
1 Scope	1
2 References.....	1
2.1 Normative references.....	1
2.2 Informative references.....	2
3 Terms and definitions	2
4 Abbreviations, acronyms and conventions.....	2
4.1 Abbreviations and acronyms	2
4.2 Conventions.....	3
5 Technical overview.....	3
5.1 Architecture	3
5.2 IPCablecom2 Application Manager overview	4
6 Interface description	5
6.1 Operations.....	5
6.2 Elements	6
6.3 Messages.....	9
6.4 Web Service profile.....	13
7 Application Manager requirements	13
7.1 Mapping SDP to FlowSpec	13
8 P-CSCF requirements	17
9 Security requirements	18
9.1 P-CSCF – IPAM interface.....	18
9.2 IPAM – Policy Server interface	18
Annex A – XML schema	19
Annex B – WSDL specification.....	21
Appendix I – Example call flows.....	23
I.1 On-net successful call.....	23
I.2 On-net unsuccessful call.....	25
I.3 Off-net (PSTN) successful call.....	28
I.4 Re-invite scenarios (hold, media changes).....	29
I.5 Call forking – Single line extension	31
I.6 3PCC.....	33
I.7 UEs behind same NAT	44

ITU-T Recommendation J.365

IPCablecom2 application manager interface

1 Scope

This Recommendation defines an IPCablecom2 Application Manager (IPAM) interface. This interface allows the Proxy Call Session Control Function (P-CSCF) to support Quality of Service (QoS) operations within an IPCablecom Multimedia-enabled DOCSIS-access network.

IPCablecom defines a framework to support SIP-based communication services. SIP messages may contain a Session Description Protocol (SDP) body that defines the attributes of the media streams being advertised. The interface defined in this Recommendation allows a P-CSCF to manage QoS resources by passing session identification information and SDP if available to an IPAM. It is the responsibility of the IPAM to translate the incoming messages into IPCablecom Multimedia messages and forward them to an IPCablecom Multimedia Policy Server.

It is an important objective of this work that interoperability between IPCablecom 2.0 and 3GPP IMS is provided. IPCablecom 2.0 is based upon 3GPP IMS, but includes additional functionality necessary to meet the requirements of cable operators. Recognizing developing converged solutions for wireless, wireline, and cable, it is expected that further development of IPCablecom 2.0 will continue to monitor and contribute to IMS developments in 3GPP, with the aim of alignment of 3GPP IMS and IPCablecom 2.0.

2 References

2.1 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- [ITU-T J.170] ITU-T Recommendation J.170 (2005), *IPCablecom security specification*.
- [ITU-T J.179] ITU-T Recommendation J.179 (2005), *IPCablecom support for multimedia*.
- [ITU-T J.361] ITU-T Recommendation J.361 (2006), *IPCablecom2 codec media*.
- [IETF RFC 3725] IETF RFC 3725 (2004), *Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)*.
- [IETF RFC 3890] IETF RFC 3890 (2004), *A Transport Independent Bandwidth Modifier for the Session Description Protocol (SDP)*.
- [SOAP 1.1] W3C Note 08 May 2000, *Simple Object Access Protocol (SOAP) 1.1*, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508>.
- [SOAP 1.2] W3C Recommendation 24 June 2003, *SOAP Version 1.2. Part 1: Messaging Framework*, <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>.
Part 2: Adjuncts, <http://www.w3.org/TR/2003/REC-soap12-part2-20030624/>.
- [XML 1.0] W3C Recommendation 04 February 2004, *Extensible Markup Language (XML) 1.0 (Third Edition)*, <http://www.w3.org/TR/2004/REC-xml-20040204>.

2.2 Informative references

- [ITU-T J.360] ITU-T Recommendation J.360 (2006), *IPCablecom2 architecture framework*.
- [ITU-T J.360 App.II] ITU-T Recommendation J.360 (2006), *IPCablecom2 architecture framework*. Appendix II: *Quality of Service architecture technical overview*.
- [IETF RFC 2327] IETF RFC 2327 (1998), *SDP: Session Description Protocol*.
- [IETF RFC 2617] IETF RFC 2617 (1999), *HTTP Authentication: Basic and Digest Access Authentication*.
- [IETF RFC 3264] IETF RFC 3264 (2002), *An Offer/Answer Model with the Session Description Protocol (SDP)*.
- [IETF RFC 3311] IETF RFC 3311 (2002), *The Session Initiation Protocol (SIP) UPDATE Method*.
- [WSDL] W3C Note 15 March 2001, *Web Services Description Language (WSDL) 1.1*. <http://www.w3.org/TR/wsdl>.

3 Terms and definitions

This Recommendation defines the following terms:

- 3.1 application manager:** A system that interfaces to Policy Server(s) for requesting QoS-based service on behalf of an end-user or network management system.
- 3.2 cable modem termination system:** Device at a cable head-end which implements the DOCSIS RFI MAC protocol and connects to CMs over an HFC network.
- 3.3 leg:** A single segment of a session associated with a UE (e.g., calling UE or called UE).
- 3.4 policy server:** A system that primarily acts as an intermediary between AM(s) and CMTS(s). It applies network policies to AM requests and proxies messages between the AM and CMTS.
- 3.5 quality of service:** Method used to reserve network resources and guarantee availability for applications.
- 3.6 request for comments:** Technical policy documents approved by the IETF which are available at <http://www.ietf.org/rfc.html>.

4 Abbreviations, acronyms and conventions

4.1 Abbreviations and acronyms

This Recommendation uses the following abbreviations:

AM	Application Manager
AS	Application Server
CMTS	Cable Modem Termination System
CSCF	Call Session Control Function
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol

IPAM	IPCablecom Application Manager (sometimes abbreviated as PAM)
MAC	Media Access Control
P-CSCF	Proxy Call Session Control Function
PS	Policy Server
QoS	Quality of Service
RFC	Request for Comments
SIP	Session Initiation Protocol
SSL	Secure Socket Layer
UA	User Agent
W3C	World Wide Web Consortium
WS	Web Service
WSDL	Web Services Description Language
XML	Extensible Markup Language
XSD	XML Schema Definition

4.2 Conventions

Throughout this Recommendation, the words that are used to define the significance of particular requirements are capitalized. These words are:

"MUST"	This word means that the item is an absolute requirement of this Recommendation.
"MUST NOT"	This phrase means that the item is an absolute prohibition of this Recommendation.
"SHOULD"	This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
"SHOULD NOT"	This phrase means that there may exist valid reasons in particular circumstances when the listed behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.
"MAY"	This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

5 Technical overview

This clause provides a high-level overview of the IPAM and how it fits in the overall IPCablecom architecture.

5.1 Architecture

The IPCablecom Multimedia Technical Report [ITU-T J.179] describes an architecture framework and reference model for IPCablecom Multimedia. The IPCablecom2 Architecture Framework Recommendation [ITU-T J.360] provides an overview of IPCablecom2 Reference Architecture and description of the various components. Within this architecture, the IPCablecom2 Application Manager is responsible for managing the QoS resources on behalf of the IPCablecom network.

Figure 1 illustrates the IPCablecom2 QoS architecture, and indicates the IPAM's interfaces.

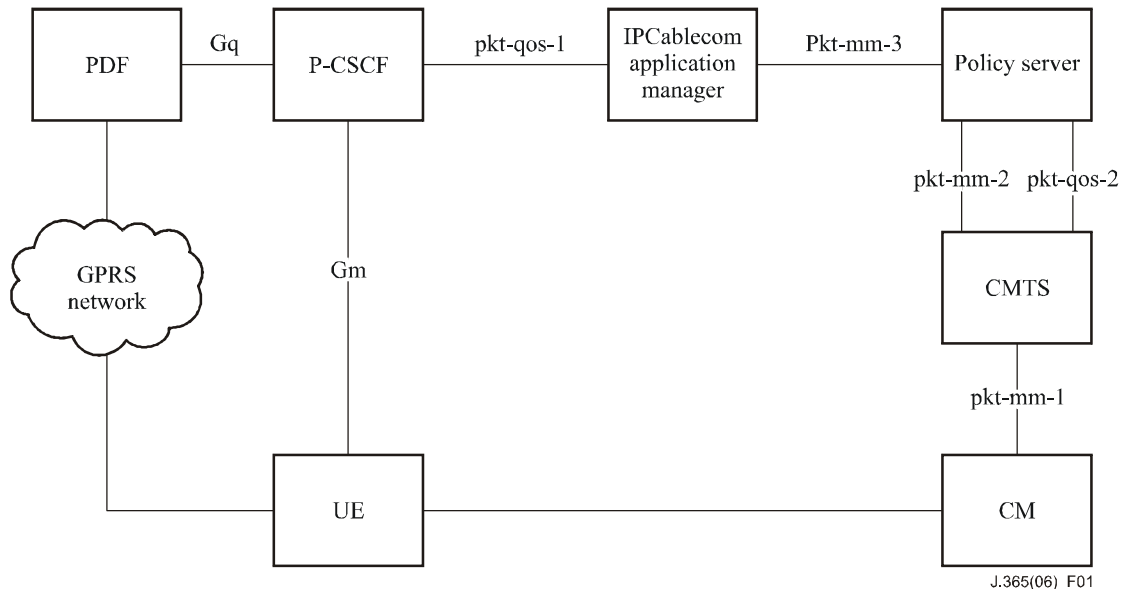


Figure 1/J.365 – IPCablecom2 QoS architecture

5.2 IPCablecom2 Application Manager overview

The IPAM enables a P-CSCF to manage QoS resources in the access network based on the Session Description Protocol (SDP) parameters contained in the offer and answer as defined in [IETF RFC 3264]. The IPAM uses the IPCablecom Multimedia pkt-mm-3 interface to communicate these requirements to an IPCablecom Multimedia Policy Server.

A typical SIP session set-up involves a UE (the calling UE) attempting to signal another UE (the called UE) of its desire to establish a session. In the simple case the following occurs:

- 1) The calling UE sends an INVITE message, containing an SDP offer, to the P-CSCF.
- 2) The P-CSCF uses the IPAM interface to reserve resources for the calling UE and then forwards the INVITE to the IMS. The reservation will be based initially on the SDP offered.
- 3) The INVITE is routed to the called UE's P-CSCF. The P-CSCF signals to its IPAM that resources need to be reserved for the called UE. The reservation will be based initially on an estimation of resources required using the caller's SDP (offer). The INVITE is then forwarded to the called UE.
- 4) When the called UE answers, a 200 OK message containing an SDP answer is sent back through the called UE's P-CSCF, routed through the IMS to the calling UE's P-CSCF, and then to the calling UE itself. During this process the P-CSCFs commit the previously reserved QoS, which is modified to reflect any changes in requirements based on the called UE's SDP (answer).
- 5) When the session completes, a BYE message is exchanged between the calling and called UEs through their respective P-CSCFs. During this process each UE's P-CSCF releases the previously committed QoS resources.

For each SIP offer/answer message received by a P-CSCF, the P-CSCF should signal the IPAM on behalf of the UE for which it is acting in processing the current message (e.g., the UE from which it received the SIP message or to which it will send the current SIP message received from the IMS). Intermediary proxies along the signalling path that are not directly responsible for any UEs involved

in the dialog should not signal any IPAM. Each P-CSCF has to signal the IPAM with the offer and answer, as both are needed to have a full view of the session's requirements.

For each session leg, the IPAM interprets the SDP contained in both the offer and answer, parsing the SDP and reading the media information including the media type, media codecs, source and destination IP addresses and ports, and then formulates an IPCablecom Multimedia Gate-Set message that is forwarded to the IPCablecom Multimedia Policy Server.

For each session leg, the IPAM creates IPCablecom Multimedia gate(s) for each media type specified. Based on the mode specified (sendonly, recvonly, sendrecv), the IPAM creates the appropriate upstream and downstream gates. For example, the SDP for a two-way video session would contain two media types: audio and video. For this type of session, the IPAM creates a total of four IPCablecom Multimedia gates consisting of an upstream gate for audio, a downstream gate for audio, an upstream gate for video and a downstream gate for video. While it is possible and may seem reasonable to create gates for the RTCP streams which are associated with the RTP streams, this is not normally done in a cable environment. This is because each gate requires the use of dedicated resources such as DOCSIS Service IDs (SIDs) which can be in very limited supply especially in legacy cable modems. Therefore, RTCP streams are normally carried on the default best-effort service flow.

6 Interface description

This clause defines the operations, elements, messages, and Web Service profile for the IPAM interface. The IPAM implements a document-style SOAP interface, each operation being associated with a specific input and output message. Operations and message types are defined within the Web Services Description Language (WSDL) and XML Schema Definition (XSD provided in Annexes A and B).

A P-CSCF invoking operations on this interface **MUST** send and receive messages as specified in the WSDL provided in Annex B.

An IPAM implementing the interface **MUST** use a document-style (document-literal) encoded SOAP interface as specified in the WSDL provided in Annex B.

An IPAM implementing the interface **MUST** process and respond to message types as specified in the WSDL provided in Annex B.

An IPAM wishing to support additional operations **MAY** extend the WSDL provided but **MUST NOT** modify the specified operations and message types.

6.1 Operations

This interface supports three operations: reserveQos, commitQos, and releaseQos. These operations allow a P-CSCF to manage QoS resources in an IPCablecom Multimedia-enabled cable access network. The operations supported by the IPAM allow a P-CSCF to optionally verify that resources are available in the access network using the reserveQos operation. The commitQos operation allows a P-CSCF to commit previously reserved resources, or to immediately commit resources that have not been previously reserved. The releaseQos operation allows a P-CSCF to free up resources that have previously been reserved or committed when a session is cancelled or has ended.

In order for a P-CSCF to manage QoS resources, it **MUST** execute one of the operations defined in Table 1 at the appropriate point in the session set-up. [IETF RFC 3264] defines an offer/answer model using SDP which is the basis for when a P-CSCF executes one of these operations. As specified, an offer/answer exchange is atomic, and therefore the P-CSCF does not have to maintain state across offer/answer exchanges.

[RFC 3311] defines an UPDATE method within SIP allowing a UE to update existing session parameters. The UPDATE may be issued while an offer/answer exchange is in progress; however, this can be treated as an update to a reservation.

An IPAM MUST update the reserved or committed resources for a given sessionID to reflect the most recently received reserveQos and commitQos operation parameters.

By providing discrete reserveQos and commitQos operations, the P-CSCF can initially request that resources are reserved based on an offer. This allows the P-CSCF to verify that resources are available before notifying the called UE. When an SDP answer is received from the called UE, the P-CSCF can update the previously reserved resources which are modified to reflect the negotiated session parameters. When a successful response (e.g., 200 OK) is received from the called UE, the P-CSCF can commit the previously reserved resources. Note that the SDP answer may be received in a 200 OK, in which case the resources can be immediately committed.

This Recommendation does not mandate the order in which the P-CSCF executes call processing (e.g., waiting for acknowledgement for a QoS operation before forwarding session signalling or not). Rather, such operation is dependent on the network operation and their desired customer experience, impacts to call establishment times, etc.

Table 1 outlines the available operations and their associated input and output messages. Each message is defined in detail in 6.3.

Table 1/J.365 – Web service operations

Operation name	Style	Messages	
reserveQos	Document	Input	reserveQosRequest (see 6.3.1)
		Output	reserveQosResponse (see 6.3.2)
commitQos	Document	Input	commitQosRequest (see 6.3.3)
		Output	commitQosResponse (see 6.3.4)
releaseQos	Document	Input	releaseQosRequest (see 6.3.5)
		Output	releaseQosResponse (see 6.3.6)

When a P-CSCF wishes to reserve QoS resources, it MUST issue a reserveQos request to the IPAM.

When a P-CSCF wishes to commit QoS resources, it MUST issue a commitQos request to the IPAM.

When the P-CSCF wishes to remove previously reserved or committed QoS resources, it MUST issue a releaseQos request to the IPAM.

6.2 Elements

This clause provides a definition for the elements passed within messages across the WS interface. Annex A contains an XSD for these element types.

6.2.1 partyInfo

The partyInfo element is a complex type which identifies the parameters associated with an individual leg of a session.

Within the IPAM, the partyInfo element is used to define the IPCablecom Multimedia Gates as well as providing information used in the execution of business policies and generation of billing events.

In IPCablecom Multimedia, QoS information for a media flow is conveyed/stored in an object called a Gate. A Gate is associated with a subscriber identifier (SubscriberID) which is either the

active cable modem or CPE IP address. The CMTS and the subscriber's cable modem filter traffic onto the flow associated with the Gate by using a traffic classifier. The classifier is defined by specifying a 5-tuple containing the source and destination IP addresses, source and destination ports, and the protocol.

The SubscriberID, Classifier, and TrafficProfile objects used to construct the IPCablecom Multimedia Gate for each flow that makes up a session are derived from the partyInfo elements contained in an array of partyInfo.

Table 2 defines the elements contained within the partyInfo type.

Table 2/J.365 – PartyInfo elements

Element name	Type	Definition
id	String	See 6.2.1.1
legId	String	See 6.2.1.2
isLocal	Boolean	See 6.2.1.3
sdp	String	See 6.2.1.4
signalingAddress	String	See 6.2.1.5

6.2.1.1 id

This is a unique identifier for a subscriber, typically the public identifier, e.g., alice@mso.net.

This id **MUST** be the same for a particular subscriber, no matter what location he/she is at when initiating or responding to a session request.

This id **SHOULD** contain an identifier that allows the IPAM to identify the associated SDP to a subscriber for the purpose of optionally evaluating subscriber-based policy rules.

6.2.1.2 legId

This is a unique identifier for a session leg (originating or terminating) that is needed to disambiguate legs that share the same session identifier and signalling address such as is the case when a session is forked. Typically, the P-CSCF will use the branch parameter from the SIP via the header they insert as the unique legId element, but any unique identifier can be generated or used by the P-CSCF.

The P-CSCF **MUST** use the same legId to identify a leg during the same transaction. This field **MUST** be supplied when this UE is local to the P-CSCF.

The legId is also used to release QoS resources for a particular leg, but not the whole session. For example, on a re-INVITE that is rejected by a 488 response, releasing the QoS resources for a particular leg by specifying the legId will only release resources reserved as a result of the re-INVITE and not resources reserved/committed for the previous offer/answer exchange.

6.2.1.3 isLocal

This is a boolean flag (true/false) that informs the IPAM whether or not this UE is local to the P-CSCF within the context of this operation request. A UE is local to a P-CSCF if the P-CSCF is the first point of contact in the signalling plane between the UE and the IMS.

The IPAM **MUST** assume a default value of false if this element is not supplied.

The P-CSCF **MUST** set this flag to true if it is responsible for setting QoS for this UE as part of the operation being performed.

Note that once this flag is set to true within a session managed by an IPAM, it **MUST** remain true for the remainder of the session.

6.2.1.4 sdp

This is the SDP contained in the offer/answer.

The P-CSCF MUST pass the unchanged SDP contained in the offer/answer if available.

6.2.1.5 signalingAddress

This is the signalling transport address of the UE initiating or responding to a session request.

The P-CSCF MUST supply the signalingAddress if isLocal is set to true.

The IPAM MUST use this field as the subscriberId field in IPCablecom Multimedia messages if present.

6.2.2 sessionId

call-id SEMICOLON from-tag [SEMICOLON to-tag]

The call-id, from-tag and to-tag MUST be extracted from the corresponding SIP headers fields. If the to-tag is not present in the SIP message, the sessionId will only contain the call-id and the from-tag. Because the to- and from-tags can be reversed (depending on which UE is issuing a request), the IPAM MUST match sessionIds with the same call-id and same (from-tag to-tag) pair. For example, the two following sessionIds are equivalent:

123456-00e0953431@151.104.2.3;590432;276439

123456-00e0953431@151.104.2.3;276439;590432

6.2.3 arrayOfPartyInfo

An array of partyInfo elements (as defined in 6.2.1) representing the UEs involved in this dialog.

The P-CSCF MUST provide an array of partyInfo elements for each reserveQos and commitQos operation. The array MUST contain one partyInfo element for each leg of the session that QoS resources are being requested.

6.2.4 emergencyCall

A flag indicating if this session is an emergency call.

The IPAM MUST assume a default value of false if this element is not supplied.

When the emergencyCall flag is true, the IPAM MUST set the sessionClassId to 0x0F (priority 7, preempt) in the corresponding IPCablecom Multimedia Gate-Set messages.

6.2.5 icId

This optional parameter holds the IMS charging identifier (ICID).

The IPAM MUST store and associate this ICID value with the session identified by the sessionId.

6.2.6 bcId

An optional Billing Correlation ID MAY be returned by the IPAM and used by an RKS, or some other back-office application to correlate Event Messages that are generated by an IPCablecom Multimedia Policy Server and CMTS.

If configured, the IPAM MUST generate a unique BCID for all of the IPCablecom Multimedia gates associated with each leg of the session and include this BCID in an Event Generation Info object sent to the Policy Server over the pkt-mm-3 interface.

The IPAM MUST return the generated BCID to the P-CSCF if configured to do so.

6.3 Messages

6.3.1 reserveQosRequest

A reserveQosRequest message is sent when a P-CSCF wishes to verify that QoS resources are available following the receipt of an offer from a UE.

This message is sent when a SIP INVITE or UPDATE message is received by the P-CSCF and is used by the IPAM to initially reserve resources in the access network ensuring that those resources are available when the called UE ultimately answers the session establishment request. If resources are not available, the P-CSCF has the opportunity to signal back to the calling UE, if the operator wishes, to block sessions that will not have associated QoS resources.

A P-CSCF MAY send a reserveQosRequest message as part of a reserveQos operation when it receives an offer (INVITE or UPDATE) from a UE that it is responsible for, or from the IMS destined for a UE that it is responsible for. A P-CSCF MAY also send a reserveQosRequest message as part of a reserveQos operation when it receives an answer in a provisional (1XX) response from a UE that it is responsible for, or from the IMS destined for a UE that it is responsible for.

A P-CSCF MAY send a 503 Service Unavailable response to the calling UE if QoS resources are not available.

An IPAM MUST send IPCablecom Multimedia Gate-Set requests containing a reserved envelope over the pkt-mm-3 interface for all local UEs associated with this session when a reserveQosRequest message is received. The IPAM MUST generate the IPCablecom Multimedia flowspec envelope as specified in 7.1.

Table 3 below defines the contents of the reserveQosRequest message.

Table 3/J.365 – reserveQosRequest message elements

Element name	Requirement	Type	Definition
sessionId	Mandatory	String	See 6.2.2.
arrayOfPartyInfo	Mandatory	Array	See 6.2.3.
emergencyCall	Optional	Boolean	See 6.2.4.
icId	Optional	String	See 6.2.5.

6.3.2 reserveQosResponse

The reserveQosResponse message MUST be sent from the IPAM in response to a reserveQosRequest.

Table 4 defines the contents of the reserveQosResponse message.

Table 4/J.365 – reserveQosReponse message elements

Element name	Requirement	Type	Definition
responseCode	Mandatory	Integer	A response code representing the success or failure of the operation. The following defines the possible response codes returned: 0: Success 1: General failure 2: Resource unavailable 3: Failed to parse request message 4: Unknown UE 200-255: Vendor-specific error
Description	Optional	String	A string describing the reason for the failure. This is primarily meant for logging and debugging purposes and is provided as information for the operator. The content of this field is left undefined and is vendor specific.
bcId	Optional	Complex	Billing Correlation ID – The RKS, or some other back-office application, uses the BCID to correlate Event Messages that are generated for a single transaction. This is one of the fields in the Event Message Header.

6.3.3 commitQosRequest

A commitQosRequest message is sent when the P-CSCF receives an answer in the form of a 200 OK or 18X with SDP message. At this stage, the IPAM has all the info it needs and MUST commit resources by setting the state of IPCablecom Multimedia gates to committed as well as setting the classifiers and QoS resources to reflect the information contained in the updated SDP. Given that resources may have been reserved at the session initiation stage, the commitment of the resources can be expected to succeed (as long as the committed resources do not exceed the reserved ones).

A P-CSCF MUST send a commitQosRequest message as part of a commitQos operation when it receives an answer (OK or ACK with SDP) to a previously forwarded offer.

An IPAM MUST send IPCablecom Multimedia Gate-Set requests containing a committed envelope over the pkt-mm-3 interface for all resources associated with the sessionId specified. The IPAM MUST generate the IPCablecom Multimedia flowspec envelope as specified in 7.1.

Table 5 defines the contents of the commitQosRequest message.

Table 5/J.365 – commitQosRequest message elements

Element name	Requirement	Type	Definition
sessionId	Mandatory	String	See 6.2.2.
arrayOfPartyInfo	Optional	Array	See 6.2.3. An array of partyInfo elements as defined in 6.2.1 and containing any final SDP to be used to commit resources.
emergencyCall	Optional	Boolean	See 6.2.4.
icId	Optional	String	See 6.2.5.

6.3.4 commitQosResponse

The commitQosResponse message MUST be sent from the IPAM in response to a commitQosRequest.

Table 6 defines the contents of the commitQosResponse message.

Table 6/J.365 – commitQosResponse message elements

Element name	Requirement	Type	Definition
result	Mandatory	Integer	A response code representing the success or failure of the operation. The following defines the possible response codes returned: 0: Success 1: General failure 2: Resource unavailable 3: Failed to parse request message 4: Unknown UE 200-255: Vendor-specific errors
description	Optional	String	A string describing the reason for the failure. This is primarily meant for logging and debugging purposes and is provided as information for the operator. The content of this field is left undefined and is vendor specific.
bcId	Optional	Complex	Billing Correlation ID – The RKS, or some other back office application, uses the BCID to correlate Event Messages that are generated for a single transaction. This is one of the fields in the Event Message Header.

6.3.5 releaseQosRequest

A releaseQosRequest message is sent when the P-CSCF determines that a dialog has terminated (in re-INVITE scenarios, some negative final responses might not terminate a dialog, e.g., a 488 due to a re-INVITE). An exception to the rule above is when a non-2xx response is received as a final response to an INVITE, in which case a releaseQosRequest message MUST be sent specifying an additional argument to the sessionId, which is the legId of the UE for which resources need to be released. On an initial INVITE, resources will be released for that UE only, unless there are no other UEs left that might create a dialog with the originating UE. In the case where the initial INVITE was forked for instance, one terminating UE might reply with a final non-2xx response, while another terminating UE follows this reply with a 200 OK; thus, the P-CSCF needs to have a way to signal the IPAM not to release all resources for a session.

Similarly on a re-INVITE, the P-CSCF would signal to the IPAM that a re-INVITE failed by sending a releaseQosRequest message and specifying the legId of the UE for which the re-INVITE was not confirmed. This way, resources reserved for the new offer are released and resources reserved/committed prior to the re-INVITE are maintained.

A P-CSCF MUST send a releaseQosRequest message as part of a releaseQos operation when it receives a termination (BYE or CANCEL) from a UE that it is responsible for, or from the IMS destined for a UE that it is responsible for. In the case where a dialog is terminated upon receipt of a BYE request for instance, the legId SHOULD NOT be included, which will result in releasing all resources associated with the dialog identified by the sessionId. Note that when QoS for both the

originating and terminating UEs is controlled by the same IPAM, it is possible that the P-CSCF representing the originating UE and that representing the terminating UE will have reserved and committed QoS resources using equivalent sessionIds. In that case, the QoS resources for both the originating and terminating UEs will be released by the IPAM on receipt of the first releaseQosRequest as a result of the receipt of the BYE by the first P-CSCF. This can be alleviated by the P-CSCF including the legId in the releaseQosRequest message. If the P-CSCF chooses to include the legId when releasing resources as a result of a BYE, the P-CSCF MUST ensure that all resources associated with the session are released for the portion it is responsible for.

A P-CSCF MUST send a releaseQosRequest message as part of a releaseQos operation when it receives an error or redirect response (4XX, 5XX or 302 messages) from a UE that it is responsible for or from the IMS destined for a UE that it is responsible for.

An IPAM MUST send IPCablecom Multimedia Gate-Delete requests over the pkt-mm-3 interface for all IPCablecom Multimedia gate resources associated with the sessionId and legId specified. If no legId is supplied, the IPAM MUST release all IPCablecom Multimedia resources associated with this session. Table 7 defines the contents of the releaseQosRequest message.

Table 7/J.365 – releaseQosRequest message elements

Element name	Requirement	Type	Definition
sessionId	Mandatory	String	See 6.2.2.
legId	Optional	String	See 6.2.1.2. This parameter reflects the legId of the UE for which resources need to be released (for a re-INVITE, this will only mean resources reserved for the rejected offer).

6.3.6 releaseQosResponse

The releaseQosResponse message MUST be sent from the IPAM in response to a releaseQosRequest.

Table 8 defines the contents of the releaseQosResponse message.

Table 8/J.365 – releaseQosReponse message elements

Element name	Requirement	Type	Definition
Result	Mandatory	Integer	A response code representing the success or failure of the operation. The following defines the possible response codes returned: 0: Success 1: General failure 2: Unknown sessionId 3: Unknown legId 200-255: Vendor-specific error
description	Optional	String	A string describing the reason for the failure. This is primarily meant for logging and debugging purposes and is provided as information for the operator. The content of this field is left undefined and is vendor specific.

6.4 Web Service profile

This clause defines P-CSCF and IPAM requirements related to the Web Services technologies used within the interface.

6.4.1 XML

The P-CSCF and IPAM MUST support XML 1.0 (Third Edition) as specified in [XML 1.0] and required by SOAP 1.1 and 1.2 standards.

6.4.2 SOAP

The P-CSCF and IPAM MUST support SOAP 1.1 as defined in [SOAP 1.1].

A P-CSCF or IPAM MAY support SOAP 1.2 as defined in [SOAP 1.2] provided that compatibility is maintained with SOAP 1.1 by using an HTTP transport and making use of HTTP POST messages.

6.4.3 Encoding

The IPAM interface MUST support a document-style encoding as defined in the WSDL provided in Annex B Transport layer.

The IPAM MUST support an HTTP transport for exchanging messages.

The IPAM MUST support an HTTPS transport for exchanging messages as defined in clause 9.

The IPAM MUST support HTTP/1.1 and HTTPS/1.1 connections.

The P-CSCF MUST use an HTTP/1.1 or HTTPS/1.1 connection to allow for persistent TCP connections and therefore increase performance.

7 Application Manager requirements

An IPAM MUST conform to the pkt-qos-1 interface requirements defined in clause 6.

An IPAM MUST conform to the pkt-mm-3 interface requirements as specified in the IPCablecom Multimedia Recommendation [ITU-T J.179] for communication to an IPCablecom Multimedia Policy Server.

Additionally, the following IPAM requirements define how the IPAM translates parameters passed over the pkt-qos-1 interface into IPCablecom Multimedia messages passed over the pkt-mm-3 interface.

7.1 Mapping SDP to FlowSpec

The 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 DOCSIS cable 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.

Integrated Services currently defines two types of service, i.e., 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:

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

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 and 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:

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

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.

For well-known audio and video codecs, the IPAM MUST use the FlowSpec parameters defined in the IPCablecom CODEC and Media Recommendation [ITU-T J.361].

For non-well-known codecs, the IPAM MUST generate a FlowSpec via the following mechanisms:

If the b=TIAS:## and a=maxprate parameters are supplied, the IPAM MUST determine the bandwidth necessary for the session as follows:

- 1) Determine which lower layers will be used and calculate the sum of the sizes of the headers in bits (h-size). In cases of variable header sizes, the average size MUST be used. For RTP-transported media, the lower layers MUST include the RTP header with header extensions, if used, the CSRC list, and any profile-specific extensions.
- 2) Retrieve the maximum packet rate from the SDP (prate = maxprate).
- 3) Calculate the transport overhead by multiplying the header sizes by the packet rate (t-over = h-size * prate).
- 4) Round the transport overhead up to nearest integer in bits (t-over = CEIL(t-over)).
- 5) Add the transport overhead to the transport independent bandwidth value (total bit-rate = bw-value + t-over).

When the above calculation is performed using the "maxprate", the bit-rate value will be the absolute maximum the media stream may use over the transport assumed in the calculations.

If the b=TIAS parameter is not supplied, the IPAM MUST use the value specified in the b=AS:## bandwidth parameter as a substitution for the value calculated above.

Using the calculated bandwidth parameter (B), the IPAM MUST set the flowspec values to the following:

$$b = \text{Bucket Size} = B / \text{maxprate}$$

$$p = r = R = B$$

$$m = b$$

$$M = 1522 \text{ bytes}$$

7.1.1 Multiple codecs

When using the SIP and SDP standards, a UE can use any of the negotiated codecs at any point during the flow. In order to inform the CMTS of the QoS requirements of the current flow and allow it to properly schedule the resources, the application must be aware of the flow parameters being used at any instant. The SIP framework, however, does not mandate this level of session awareness, i.e., SIP does not mandate that the UE notifies the signalling layer when making a change in the flow. Therefore, the IPAM may not know at a certain point in time what flow parameters a UE is using in a session.

Due to the fact that the IPAM does not necessarily know the flow parameters being used at point in time on a flow, the IPAM MUST commit the Least Upper Bound (LUB) when the flow is in the active state. By doing so, the IPAM can be assured that the UE will always have the resources necessary for any codec they use in the negotiated set.

7.1.1.1 Calculation of the Least Upper Bound (LUB)

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 order to support this functionality, the IPAM MUST send a FlowSpec which contains the Least Upper Bound (LUB) of the necessary flow parameters for the individual flows.

The Least Upper Bound (LUB) of two flows A and B, LUB(A, B), is the "smallest" envelope that can carry both of the flows A, B non-simultaneously. LUB(A, B) is calculated on a parameter-by-parameter basis as follows:

Define the TSpec values for a flow α . Also define the period P_α as M_α/r_α . Then LUB(A, B) is given by:

$$\begin{aligned} \text{LUB}(A, B) \equiv \{ & b_{\text{LUB}(A, B)} \equiv \text{MAX}(b_A, b_B), \\ & r_{\text{LUB}(A, B)} \equiv (M_{\text{LUB}(A, B)} / P_{\text{LUB}(A, B)}), \\ & p_{\text{LUB}(A, B)} \equiv \text{MAX}(p_A, p_B, r_{\text{LUB}(A, B)}), \\ & m_{\text{LUB}(A, B)} \equiv \text{MAX}(m_A, m_B), \\ & M_{\text{LUB}(A, B)} \equiv \text{MAX}(M_A, M_B) \\ & \} \end{aligned}$$

where:

$$P_{LUB(A, B)} \equiv GCF(P_A, P_B);$$

MAX(x, y) means "take the higher of the pair (x, y)";

$$MAX(x, y, z) \equiv MAX(MAX(x, y), z);$$

GCF(x, y) means "take the Greatest Common Factor of the pair (x, y)".

The LUB of n flows ($n \neq 2$), $LUB(n_1, n_2, \dots)$, is defined recursively as:

$$LUB(n_1, n_2, \dots, N) \equiv LUB(n_1, LUB(n_2, \dots, N))$$

In addition, the slack term in the corresponding RSpec must allow any component flow to use the resources. In order to ensure that this criterion is met, the RSpec for the flow is set to the minimum value of the RSpec values in the component flows. That is:

$$S_{LUB(A, B)} \equiv MIN(S_A, S_B)$$

Where MIN(x, y) means "take the lower of the pair (x, y)".

The following example shows how TSpec parameters are determined using LUB algorithm specified above:

1) As the result of codec negotiation, the following codecs are selected for a session:

G711(20 ms) and G728(10 ms)

2) The LUB bucket depth for the selected codecs is:

$$G711(20 \text{ ms}) = (8000/50) + 40 = 200 \text{ bytes}$$

$$G728(10 \text{ ms}) = (2000/100) + 40 = 60 \text{ bytes}$$

$$b[LUB] = m[LUB] = M[LUB] = MAX(200, 60) = 200 \text{ bytes}$$

3) The LUB bucket rate for selected codecs is:

$$P[LUB] = GCF(10 \text{ ms}, 20 \text{ ms}) = 10 \text{ ms} = 0.01 \text{ second}$$

$$r[LUB] = M * 1/P = 200 * 1/0.01 = 20,000 \text{ bytes per second}$$

$$r[G711(20 \text{ ms})] = 200 * 1/0.02 = 10,000 \text{ bytes per second}$$

$$r[G728(10 \text{ ms})] = 60 * 1/0.01 = 6,000 \text{ bytes per second}$$

$$p[LUB] = MAX(10000, 6000, 20000) = 20,000 \text{ bytes per second}$$

7.1.2 Mapping SDP to classifiers

Classifiers MUST be built as specified in SDP. The IPAM MUST generate the classifiers for the gate requests from a series of one or more reserveQos and/or commitQos operations from the P-CSCF. A classifier is composed of two fields: address and port. The IPAM MUST generate the address portion of the classifier from the "a=Local-TURN" attribute in the SDP of the QoS operation when present. If the "a=Local-TURN" is not present, then the IPAM MUST use the "signalingAddress" field of the QoS operation. If neither the "a=Local-TURN" nor "signalingAddress" has been supplied in any of the QoS operations, then the IPAM MUST use the "c=", connection data, field's <connection address> in the supplied SDP.

The IPAM MUST use the media descriptor, "m=", field's <port> subfield in SDP for the respective classifier.

If the QoS request, reserveQos or commitQos, does not specify all the information to fully qualify the packet classifier the IPAM SHOULD use the IPCablecom Multimedia classifier wildcard value, 0, initially for the Gate-Set and issue a Gate-Set modify when the remaining classifier information is learned.

7.1.3 Understanding of Session Mode

For each media descriptor line in the SDP the IPAM MUST create one or two gates. The number of gates for each media descriptor MUST be determined by inspecting the media direction attribute: "a:sendonly", "a:recvonly", or "a:sendrecv".

When the media descriptor includes "a:sendrecv" the IPAM MUST generate two gates: one gate in the upstream direction and one gate in the downstream direction for the media.

When the media descriptor is "a:sendonly", the IPAM MUST generate an upstream gate request.

When the media descriptor is "a:recvonly", the IPAM MUST generate a downstream gate request.

The IPAM MUST reserve and/or commit the direction of the gate based on the isLocal attribute as included in the QoS request.

The Session Mode may also change mid-session when a call is put on hold. In this case a re-INVITE would be generated by the UE with an updated SDP where the Session Mode is set to recvonly. Depending upon an operator-configured option, the gates associated with the media could remain unchanged, be put into a reserved state, or deleted. When the call is taken off hold and the UE generates another re-INVITE with the Session Mode changed to sendrecv, the gates would be left unchanged, put into a committed state or recreated. The method of configuring the IPAM is out of scope of this Recommendation.

7.1.4 Handling NAT

When two UEs are located behind the same NAT, the signalling address specified within the partyInfo object will be the same but each UE will be identified by a unique legId.

When the UEs support ICE and are behind the same NAT, they will re-invite with their private addresses in the m/c lines but with the same signalling address. The IPAM MUST recognize that the UEs share the same signalling address, but have different private media addresses and release any IPCablecom Multimedia resources allocated for this dialog as the media will now be flowing over the local interfaces.

8 P-CSCF requirements

A P-CSCF MUST conform to the pkt-qos-1 interface requirements defined in clause 6.

Additionally, the following P-CSCF requirements define how the P-CSCF MUST use the pkt-qos-1 interface to reserve, commit, and release resources.

8.1.1 Reserving resources

A P-CSCF MUST send a reserveQosRequest message as part of a reserveQos operation when it receives an offer (SIP INVITE or UPDATE) from a UE that it is responsible for.

Upon receipt of a reserveQosResponse message with a non-zero response code, a P-CSCF MAY send a 503 Service Unavailable response to the calling UE.

When reserving resources, the P-CSCF MUST include the IP address associated with the flow for which the offer is associated with, in the signalingAddress attribute of the partyInfo element.

8.1.2 Committed resources

A P-CSCF MUST send a commitQosRequest message as part of a commitQos operation when it receives an answer (OK or ACK with SDP) to a previously forwarded offer.

When committing resources, the P-CSCF MUST include the IP address associated with the flow for which the answer is associated with, in the signalingAddress attribute of the partyInfo element.

8.1.3 Removing resources

A P-CSCF MUST send a releaseQosRequest message as part of a releaseQos operation when it receives a termination (BYE or CANCEL) for a UE that it is responsible for.

A P-CSCF MUST send a releaseQosRequest message as part of a releaseQos operation when it receives an error or redirect response (4XX, 5XX or 302 messages) for a UE that it is responsible for.

9 Security requirements

9.1 P-CSCF – IPAM interface

The P-CSCF – IPAM interface MUST be secured using the IPCablecom Profile for TLS as described in 6.9 of the IPCablecom Security Specification, see [ITU-T J.170]. This provides for mutual endpoint authentication and confidentiality of messages.

9.2 IPAM – Policy Server interface

The IPAM MUST support the IPCablecom Multimedia Security Requirement as defined in 8.3 of the IPCablecom Multimedia Recommendation [ITU-T J.179].

Annex A

XML schema

```
<schema xmlns:tns="http://www.cablelabs.com/namespaces/PacketCable/R2/XSD/PAMI" xmlns:soap11-
enc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/" xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.cablelabs.com/namespaces/PacketCable/R2/XSD/PAMI">
  <complexType name="partyInfo">
    <sequence>
      <element name="id" type="string" nillable="true"
minOccurs="0"/>
      <element name="legId" type="string" nillable="true"
minOccurs="0"/>
      <element name="isLocal" type="boolean" minOccurs="0"/>
      <element name="sdp" nillable="true" minOccurs="0">
        <simpleType>
          <restriction base="string">
            <whiteSpace value="preserve"/>
          </restriction>
        </simpleType>
      </element>
      <element name="signalingAddress" type="string"
nillable="true" minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="BCIDType">
    <sequence>
      <element name="BCID">
        <simpleType>
          <restriction base="string">
            <length value="48"/>
          </restriction>
        </simpleType>
      </element>
    </sequence>
  </complexType>
  <complexType name="commitQosRequestType">
    <sequence>
      <element name="sessionId" type="string" nillable="true"/>
      <element name="arrayOfPartyInfo" type="tns:partyInfo"
nillable="true" maxOccurs="unbounded"/>
      <element name="emergencyCall" type="boolean" minOccurs="0"/>
      <element name="icId" type="string" minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="commitQosResponseType">
    <sequence>
      <element name="responseCode" type="int"/>
      <element name="description" type="string" minOccurs="0"/>
      <element name="bcid" type="tns:BCIDType" minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="releaseQosRequestType">
    <sequence>
      <element name="sessionId" type="string" nillable="true"/>
      <element name="legId" type="string" nillable="true"
minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="releaseQosResponseType">
    <sequence>
      <element name="result" type="int"/>
      <element name="description" type="string" minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="reserveQosRequestType">
    <sequence>
      <element name="sessionId" type="string" nillable="true"/>
      <element name="arrayOfPartyInfo" type="tns:partyInfo"
nillable="true" maxOccurs="unbounded"/>
      <element name="emergencyCall" type="boolean" minOccurs="0"/>
      <element name="icId" type="string" minOccurs="0"/>
    </sequence>
  </complexType>
```

```
<complexType name="reserveQosResponseType">
  <sequence>
    <element name="result" type="int"/>
    <element name="description" type="string" minOccurs="0"/>
    <element name="bcid" type="tns:BCIDType" minOccurs="0"/>
  </sequence>
</complexType>
<element name="commitQosRequest" type="tns:commitQosRequestType"/>
<element name="commitQosResponse" type="tns:commitQosResponseType"/>
<element name="releaseQosRequest" type="tns:releaseQosRequestType"/>
<element name="releaseQosResponse" type="tns:releaseQosResponseType"/>
<element name="reserveQosRequest" type="tns:reserveQosRequestType"/>
<element name="reserveQosResponse" type="tns:reserveQosResponseType"/>
</schema>
```

Annex B

WSDL specification

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:pc="http://www.cablelabs.com/namespaces/PacketCable/R2/XSD/PAMI"
xmlns:tns="http://www.cablelabs.com/namespaces/PacketCable/R2/WSDL/PAMI"
targetNamespace="http://www.cablelabs.com/namespaces/PacketCable/R2/WSDL/PAMI">
  <types>
    <xs:schema
targetNamespace="http://www.cablelabs.com/namespaces/PacketCable/R2/XSD/PAMI">
      <xs:include schemaLocation=
"http://www.cablelabs.com/namespaces/PacketCable/R2/XSD/PAMI.xsd"/>
    </xs:schema>
  </types>
  <message name="reserveQosRequest">
    <part name="parameter" element="pc:reserveQosRequest"/>
  </message>
  <message name="reserveQosResponse">
    <part name="parameter" element="pc:reserveQosResponse"/>
  </message>
  <message name="commitQosRequest">
    <part name="parameter" element="pc:commitQosRequest"/>
  </message>
  <message name="commitQosResponse">
    <part name="parameter" element="pc:commitQosResponse"/>
  </message>
  <message name="releaseQosRequest">
    <part name="parameter" element="pc:releaseQosRequest"/>
  </message>
  <message name="releaseQosResponse">
    <part name="parameter" element="pc:releaseQosResponse"/>
  </message>
  <portType name="SOAPport">
    <operation name="reserveQos">
      <input message="tns:reserveQosRequest"/>
      <output message="tns:reserveQosResponse"/>
    </operation>
    <operation name="commitQos">
      <input message="tns:commitQosRequest"/>
      <output message="tns:commitQosResponse"/>
    </operation>
    <operation name="releaseQos">
      <input message="tns:releaseQosRequest"/>
      <output message="tns:releaseQosResponse"/>
    </operation>
  </portType>
  <binding name="pcAMbinding" type="tns:SOAPport">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="reserveQos">
      <soap:operation soapAction="urn:#reserveQos" style="document"/>
      <input>
        <soap:body parts="parameter" use="literal"/>
      </input>
      <output>
        <soap:body parts="parameter" use="literal"/>
      </output>
    </operation>
    <operation name="commitQos">
      <soap:operation soapAction="urn:#commitQos" style="document"/>
      <input>
        <soap:body parts="parameter" use="literal"/>
      </input>
      <output>
        <soap:body parts="parameter" use="literal"/>
      </output>
    </operation>
    <operation name="releaseQos">
      <soap:operation soapAction="urn:#releaseQos" style="document"/>
      <input>
```

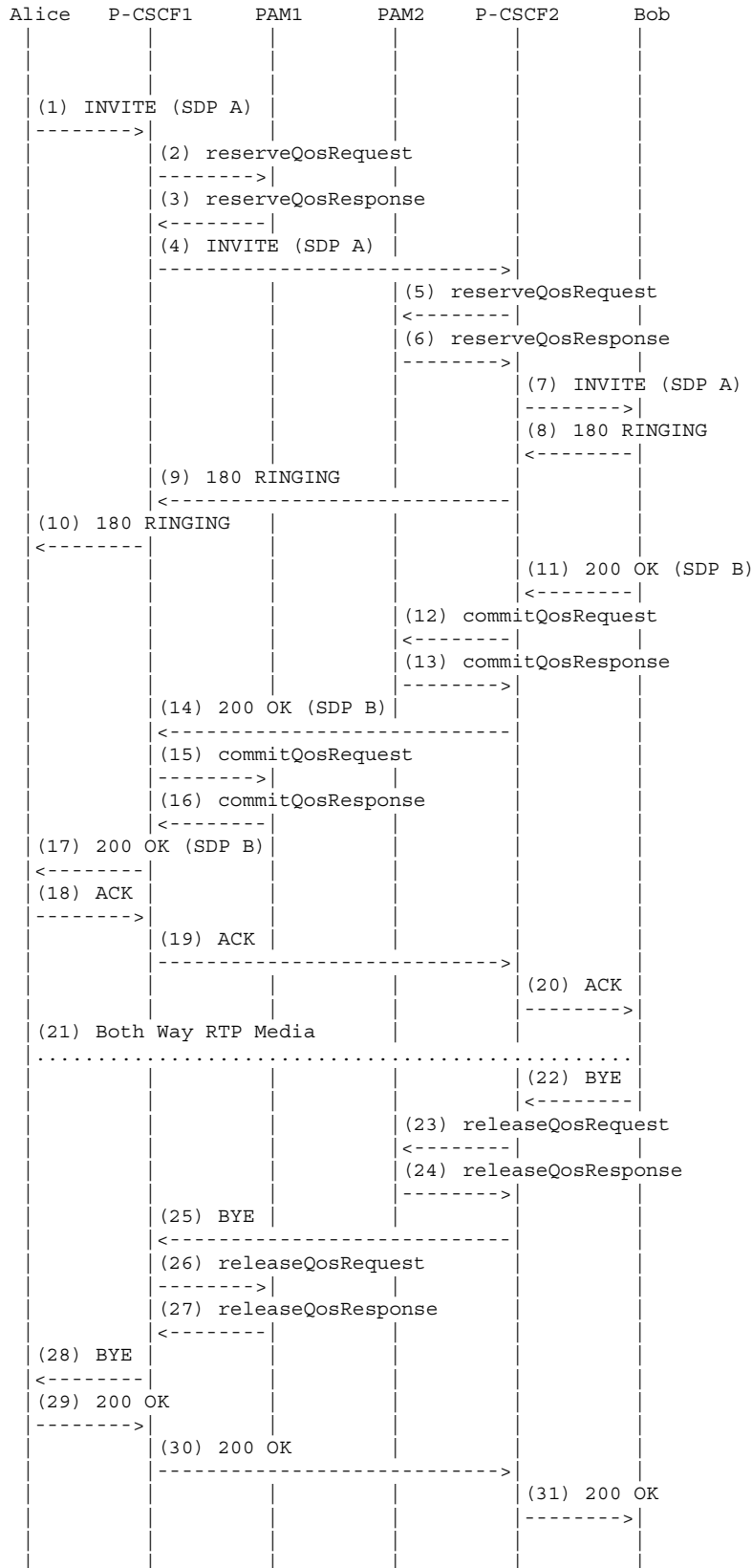
```
        <soap:body parts="parameter" use="literal"/>
    </input>
    <output>
        <soap:body parts="parameter" use="literal"/>
    </output>
</operation>
</binding>
<service name="pcAM">
    <port name="pcAMport" binding="tns:pcAMbinding">
        <soap:address location="http://youraddress"/>
    </port>
</service>
</definitions>
```

Appendix I

Example call flows

This clause provides example operational behaviour based on the interfaces and requirements defined within this Recommendation. The Call flows provided in this clause are for reference only.

I.1 On-net successful call



```

(2) reserveQosRequest P-CSCF1 -> PAM1
<reserveQosRequest>
  <sessionId>1234@mso.net;alicetag</sessionId>
  <ArrayOfPartyInfo>
    <PartyInfo>
      <id>alice@mso.net</id>
      <legId>z9hG4bK74bf9</legId>
      <isLocal>true</isLocal>
      <sdp>SDP A</sdp>
    </PartyInfo>
  </ArrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>

(3) reserveQosResponse PAM1 -> P-CSCF1
<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>

(5) reserveQosRequest P-CSCF2 -> PAM2
<reserveQosRequest>
  <sessionId>1234@mso.net;alicetag</sessionId>
  <ArrayOfPartyInfo>
    <PartyInfo>
      <isLocal>>false</isLocal>
      <sdp>SDP A</sdp>
    </PartyInfo>
    <PartyInfo>
      <id>bob@mso.net</id>
      <legId>z9hG4bK7a3d4</legId>
      <isLocal>true</isLocal>
      <signalingAddress>Bob IP Address</signalingAddress>
    </PartyInfo>
  </ArrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>

(6) reserveQosResponse PAM2 -> P-CSCF2
<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>

(12) commitQosRequest P-CSCF2 -> PAM2
<commitQosRequest>
  <sessionId>1234@mso.net;alicetag;bobtag</sessionId>
  <ArrayOfPartyInfo>
    <legId> z9hG4bK7a3d4</legId>
    <isLocal>true</isLocal>
    <sdp>SDP B</sdp>
  </ArrayOfPartyInfo>
</commitQosRequest>

(13) commitQosResponse PAM2 -> P-CSCF2
<commitQosResponse>
  <result>0</result>
</commitQosResponse>

(15) commitQosRequest P-CSCF1 -> PAM1
<commitQosRequest>
  <sessionId>1234@mso.net;alicetag;bobtag</sessionId>
  <ArrayOfPartyInfo>
    <isLocal>>false</isLocal>
    <sdp>SDP B</sdp>
  </ArrayOfPartyInfo>
</commitQosRequest>

(16) commitQosResponse PAM1 -> P-CSCF1
<commitQosResponse>
  <result>0</result>
</commitQosResponse>

(23) releaseQosRequest P-CSCF2 -> PAM2
<releaseQosRequest>
  <sessionId>1234@mso.net;bobtag;alicetag</sessionId>
</releaseQosRequest>

```

```

(24) releaseQosResponse PAM2 -> P-CSCF2
<releaseQosResponse>
  <result>0</result>
</releaseQosResponse>

(26) releaseQosRequest P-CSCF1 -> PAM1
<releaseQosRequest>
  <sessionId>1234@mso.net;bobtag;alicetag</sessionId>

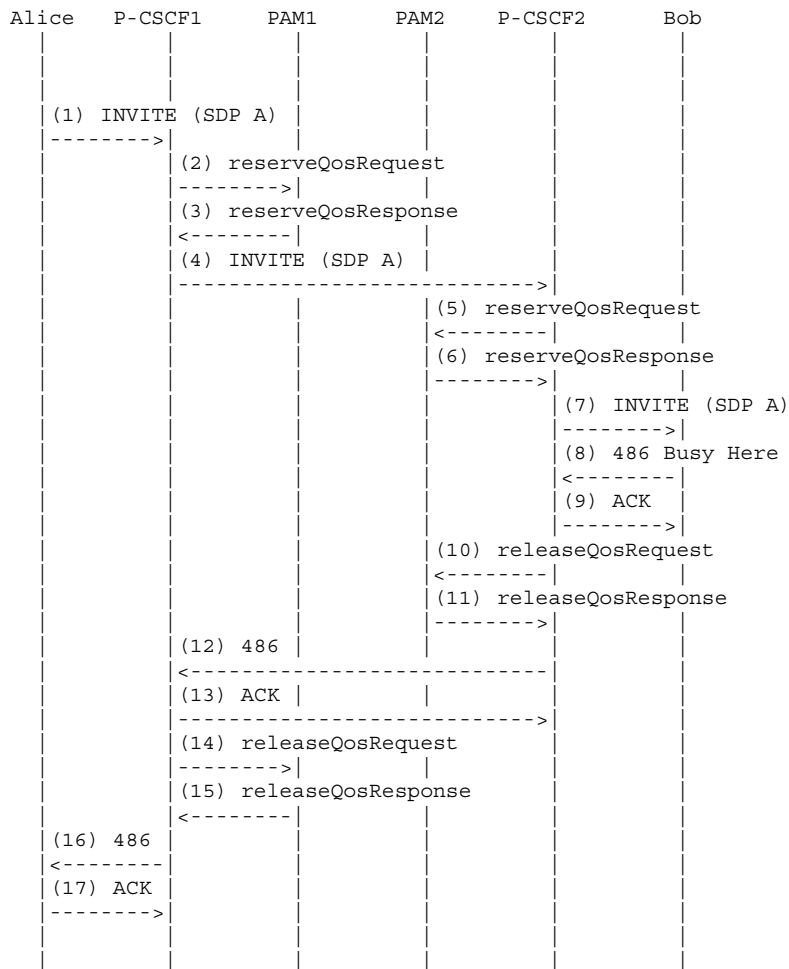
</releaseQosRequest>

(27) releaseQosResponse PAM1 -> P-CSCF1
<releaseQosResponse>
  <result>0</result>
</releaseQosResponse>

```

I.2 On-net unsuccessful call

I.2.1 Callee busy



```

(2) reserveQosRequest P-CSCF1 -> PAM1
<reserveQosRequest>
  <sessionId>1234@mso.net;alicetag</sessionId>
  <arrayOfPartyInfo>
    <PartyInfo>
      <id>alice@mso.net</id>
      <legId>z9hG4bK74bf9</legId>
      <isLocal>true</isLocal>
      <sdp>SDP A</sdp>
    </PartyInfo>
  </arrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>

```

```

(3) reserveQosResponse PAM1 -> P-CSCF1
<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>

(5) reserveQosRequest P-CSCF2 -> PAM2
<reserveQosRequest>
  <sessionId>1234@mso.net;alictag</sessionId>
  <arrayOfPartyInfo>
    <PartyInfo>
      <isLocal>>false</isLocal>
      <sdp>SDP A</sdp>
    </PartyInfo>
    <PartyInfo>
      <id>bob@mso.net</id>
      <legId>z9hG4bK7a3d4</legId>
      <isLocal>>true</isLocal>
      <signalingAddress>Bob IP Address</signalingAddress>
    </PartyInfo>
  </arrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>

(6) reserveQosResponse PAM2 -> P-CSCF2
<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>

(10) releaseQosRequest P-CSCF2 -> PAM2
<releaseQosRequest>
  <sessionId>1234@mso.net;alictag;bobtag</sessionId>
  <legId>z9hG4bK7a3d4</legId>
</releaseQosRequest>

(11) releaseQosResponse PAM2 -> P-CSCF2
<releaseQosResponse>
  <result>0</result>
</releaseQosResponse>

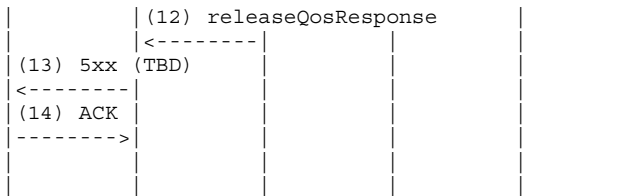
(14) releaseQosRequest P-CSCF1 -> PAM1
<releaseQosRequest>
  <sessionId>1234@mso.net;alictag;bobtag</sessionId>
  <legId>z9hG4bK74bf9</legId>
</releaseQosRequest>

(15) releaseQosResponse PAM1 -> P-CSCF1
<releaseQosResponse>
  <result>0</result>
</releaseQosResponse>

```

I.2.2 Resource unavailable





(2) reserveQosRequest P-CSCF1 -> PAM1

```
<reserveQosRequest>
  <sessionId>1234@mso.net;alictag</sessionId>
  <arrayOfPartyInfo>
    <PartyInfo>
      <id>alice@mso.net</id>
      <legId>z9hG4bK74bf9</legId>
      <isLocal>true</isLocal>
      <sdp>SDP A</sdp>
    </PartyInfo>
  </arrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>
```

(3) reserveQosResponse PAM1 -> P-CSCF1

```
<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>
```

(5) reserveQosRequest P-CSCF2 -> PAM2

```
<reserveQosRequest>
  <sessionId>1234@mso.net;alictag</sessionId>
  <arrayOfPartyInfo>
    <PartyInfo>
      <isLocal>>false</isLocal>
      <sdp>SDP A</sdp>
    </PartyInfo>
    <PartyInfo>
      <id>bob@mso.net</id>
      <legId>z9hG4bK7a3d4</legId>
      <isLocal>true</isLocal>
      <signalingAddress>Bob IP Address</signalingAddress>
    </PartyInfo>
  </arrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>
```

(6) reserveQosResponse PAM2 -> P-CSCF2 (QoS Reservation Failure)

```
<reserveQosResponse>
  <result>3</result>
</reserveQosResponse>
```

(7) releaseQosRequest P-CSCF2 -> PAM2

```
<releaseQosRequest>
  <sessionId>1234@mso.net;alictag</sessionId>
  <legId>z9hG4bK7a3d4</legId>
</releaseQosRequest>
```

(8) releaseQosResponse PAM2 -> P-CSCF2

```
<releaseQosResponse>
  <result>0</result>
</releaseQosResponse>
```

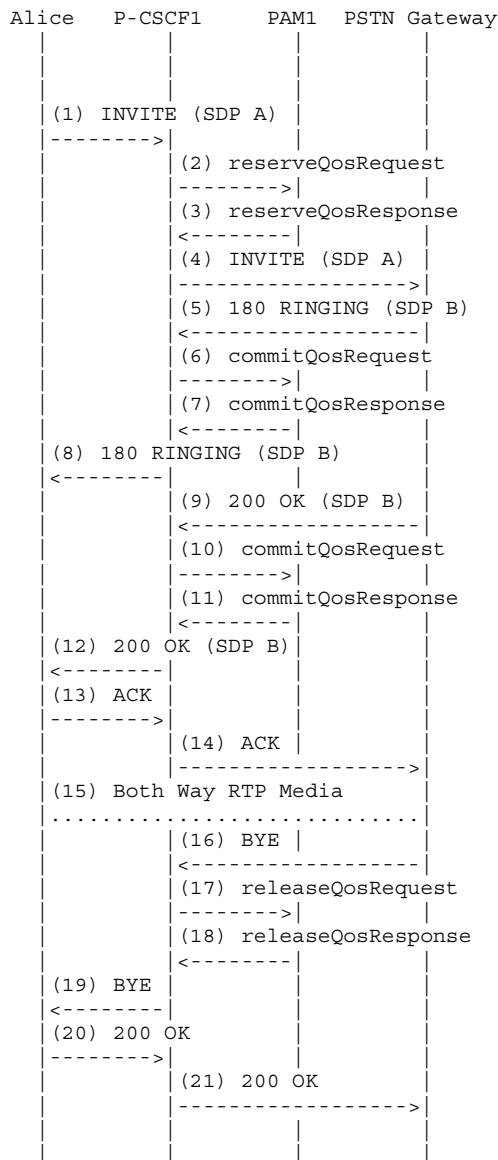
(11) releaseQosRequest P-CSCF1 -> PAM1

```
<releaseQosRequest>
  <sessionId>1234@mso.net;alictag</sessionId>
  <legId>z9hG4bK74bf9</legId>
</releaseQosRequest>
```

(12) releaseQosResponse PAM1 -> P-CSCF1

```
<releaseQosResponse>
  <result>0</result>
</releaseQosResponse>
```

I.3 Off-net (PSTN) successful call



(2) reserveQosRequest P-CSCF1 -> PAM1

```
<reserveQosRequest>
  <sessionId>1234@mso.net;alictag</sessionId>
  <arrayOfPartyInfo>
    <PartyInfo>
      <id>alice@mso.net</id>
      <legId>z9hG4bK74bf9</legId>
      <isLocal>true</isLocal>
      <sdp>SDP A</sdp>
    </PartyInfo>
  </arrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>
```

(3) reserveQosResponse PAM1 -> P-CSCF1

```
<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>
```

(6) commitQosRequest P-CSCF1 -> PAM1

```
<commitQosRequest>
  <sessionId>1234@mso.net;alictag;psntag</sessionId>
  <arrayOfPartyInfo>
    <isLocal>>false</isLocal>
    <sdp>SDP B</sdp>
  </arrayOfPartyInfo>
</commitQosRequest>
```

```

(7) commitQosResponse PAM1 -> P-CSCF1
<commitQosResponse>
  <result>0</result>
</commitQosResponse>

(10) commitQosRequest P-CSCF1 -> PAM1
<commitQosRequest>
  <sessionId>1234@mso.net;alicetag;pstntag</sessionId>
  <arrayOfPartyInfo>
    <isLocal>>false</isLocal>
    <sdp>SDP B</sdp>
  </arrayOfPartyInfo>
</commitQosRequest>

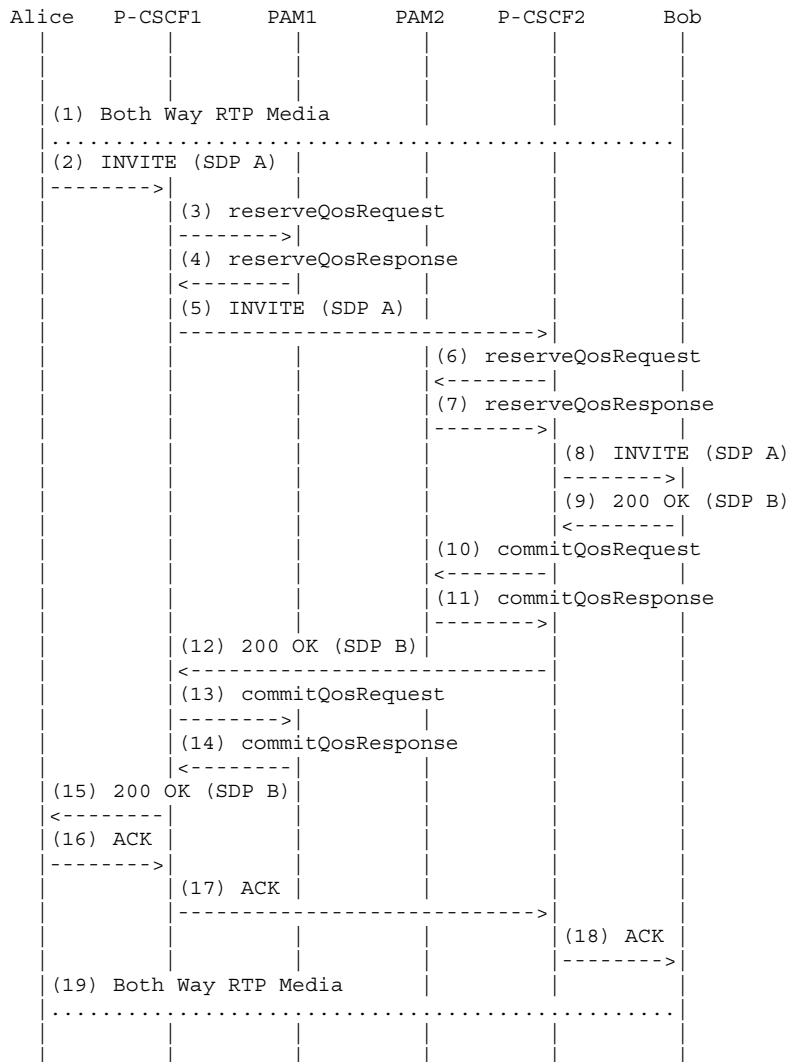
(11) commitQosResponse PAM1 -> P-CSCF1
<commitQosResponse>
  <result>0</result>
</commitQosResponse>

(17) releaseQosRequest P-CSCF1 -> PAM1
<releaseQosRequest>
  <sessionId>1234@mso.net;pstntag;alicetag</sessionId>
</releaseQosRequest>

(18) releaseQosResponse PAM1 -> P-CSCF1
<releaseQosResponse>
  <result>0</result>
</releaseQosResponse>

```

I.4 Re-invite scenarios (hold, media changes)



```

(3) reserveQosRequest P-CSCF1 -> PAM1
<reserveQosRequest>
  <sessionId>1234@mso.net;alictag;bobtag</sessionId>
  <ArrayOfPartyInfo>
    <PartyInfo>
      <legId>z9hG4bK74bf9</legId>
      <sdp>SDP A</sdp>
    </PartyInfo>
  </ArrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>

(4) reserveQosResponse PAM1 -> P-CSCF1
<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>

(6) reserveQosRequest P-CSCF2 -> PAM2
<reserveQosRequest>
  <sessionId>1234@mso.net;alictag;bobtag </sessionId>
  <ArrayOfPartyInfo>
    <PartyInfo>
      <sdp>SDP A</sdp>
    </PartyInfo>
  </ArrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>

(7) reserveQosResponse PAM2 -> P-CSCF2
<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>

(10) commitQosRequest P-CSCF2 -> PAM2
<commitQosRequest>
  <sessionId>1234@mso.net;alictag;bobtag</sessionId>
  <ArrayOfPartyInfo>
    <legId>z9hG4bK7a3d4</legId>
    <isLocal>>true</isLocal>
    <sdp>SDP B</sdp>
  </ArrayOfPartyInfo>
</commitQosRequest>

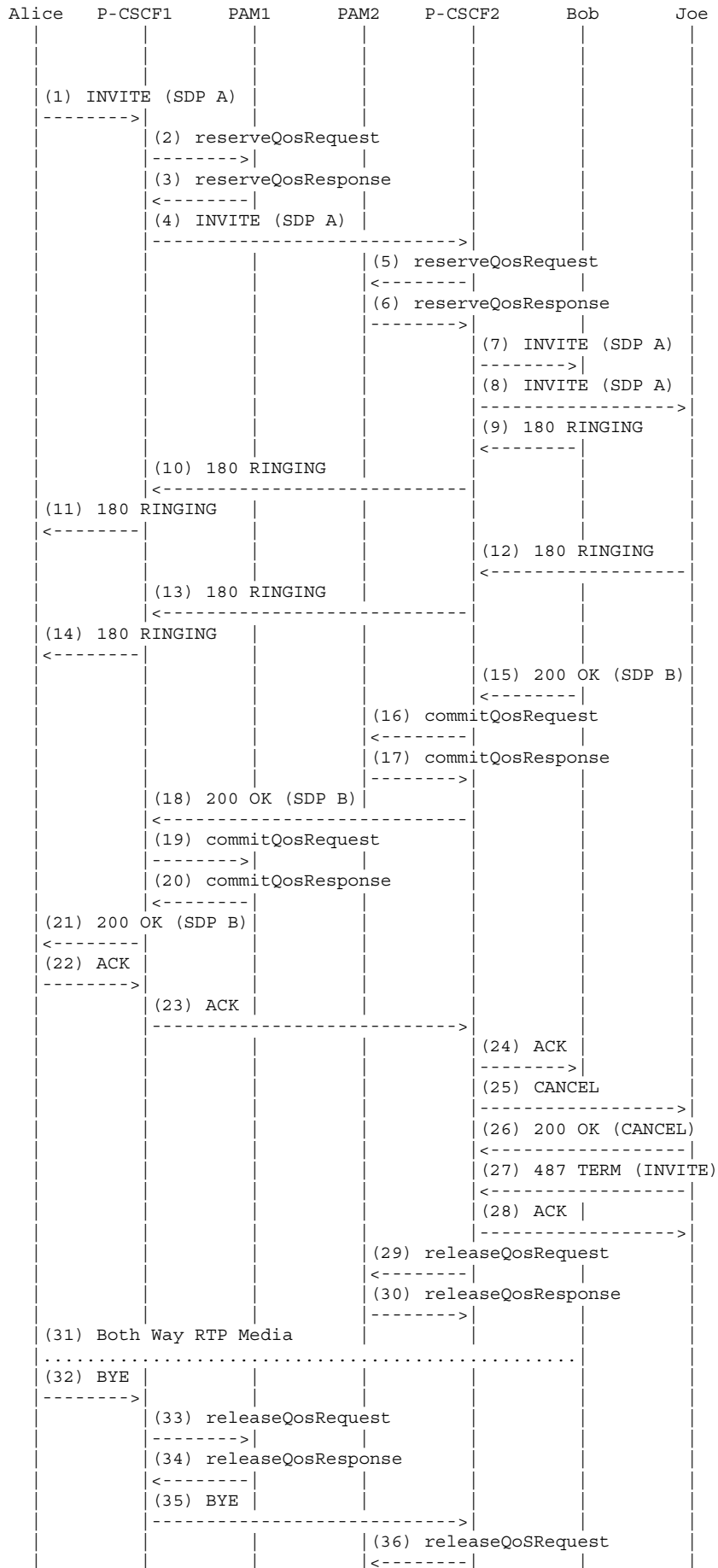
(11) commitQosResponse PAM2 -> P-CSCF2
<commitQosResponse>
  <result>0</result>
</commitQosResponse>

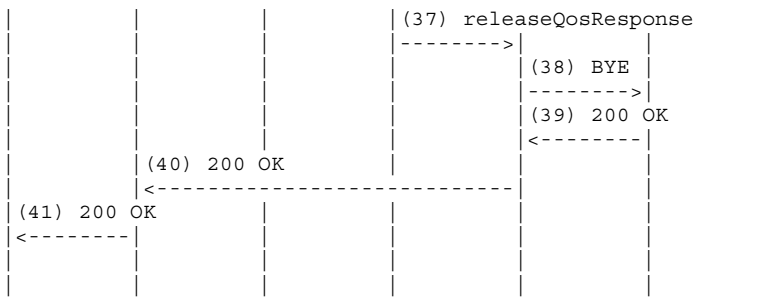
(13) commitQosRequest P-CSCF1 -> PAM1
<commitQosRequest>
  <sessionId>1234@mso.net;alictag;bobtag</sessionId>
  <ArrayOfPartyInfo>
    <isLocal>>false</isLocal>
    <sdp>SDP B</sdp>
  </ArrayOfPartyInfo>
</commitQosRequest>

(14) commitQosResponse PAM1 -> P-CSCF1
<commitQosResponse>
  <result>0</result>
</commitQosResponse>

```

I.5 Call forking – Single line extension





(2) reserveQosRequest P-CSCF1 -> PAM1

```

<reserveQosRequest>
  <sessionId>1234@mso.net;alicetag</sessionId>
  <arrayOfPartyInfo>
    <PartyInfo>
      <id>alice@mso.net</id>
      <legId>z9hG4bK74bf9</legId>
      <isLocal>true</isLocal>
      <sdp>SDP A</sdp>
    </PartyInfo>
  </arrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>
  
```

(3) reserveQosResponse PAM1 -> P-CSCF1

```

<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>
  
```

(5) reserveQosRequest P-CSCF2 -> PAM2

```

<reserveQosRequest>
  <sessionId>1234@mso.net;alicetag</sessionId>
  <arrayOfPartyInfo>
    <PartyInfo>
      <isLocal>>false</isLocal>
      <sdp>SDP A</sdp>
    </PartyInfo>
    <PartyInfo>
      <id>bob@mso.net</id>
      <legId>z9hG4bK7a3d4</legId>
      <isLocal>true</isLocal>
      <signalingAddress>Bob IP Address</signalingAddress>
    </PartyInfo>
    <PartyInfo>
      <id>joe@mso.net</id>
      <legId>z9hG4bK71a2b</legId>
      <isLocal>true</isLocal>
      <signalingAddress>Joe IP Address</signalingAddress>
    </PartyInfo>
  </arrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>
  
```

(6) reserveQosResponse PAM2 -> P-CSCF2

```

<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>
  
```

(16) commitQosRequest P-CSCF2 -> PAM2

```

<commitQosRequest>
  <sessionId>1234@mso.net;alicetag;bobtag</sessionId>
  <arrayOfPartyInfo>
    <legId>z9hG4bK7a3d4</legId>
    <isLocal>true</isLocal>
    <sdp>SDP B</sdp>
  </arrayOfPartyInfo>
</commitQosRequest>
  
```

(17) commitQosResponse PAM2 -> P-CSCF2

```

<commitQosResponse>
  <result>0</result>
</commitQosResponse>
  
```

(19) commitQosRequest P-CSCF1 -> PAM1

```

<commitQosRequest>
  <sessionId>1234@mso.net;alicetag;bobtag</sessionId>
  
```

```

    <arrayOfPartyInfo>
      <isLocal>false</isLocal>
      <sdp>SDP B</sdp>
    </arrayOfPartyInfo>
  </commitQosRequest>

(20) commitQosResponse PAM1 -> P-CSCF1
<commitQosResponse>
  <result>0</result>
</commitQosResponse>

(29) releaseQosRequest P-CSCF2 -> PAM2
<releaseQosRequest>
  <sessionId>1234@mso.net;alictag;joetag</sessionId>
  <legId>z9hG4bK71a2b</legId>
</releaseQosRequest>

(30) releaseQosResponse PAM2 -> P-CSCF2
<releaseQosResponse>
  <result>0</result>
</releaseQosResponse>

(33) releaseQosRequest P-CSCF1 -> PAM1
<releaseQosRequest>
  <sessionId>1234@mso.net;alictag;bobtag</sessionId>
</releaseQosRequest>

(34) releaseQosResponse PAM1 -> P-CSCF1
<releaseQosResponse>
  <result>0</result>
</releaseQosResponse>

(36) releaseQosRequest P-CSCF2 -> PAM2
<releaseQosRequest>
  <sessionId>1234@mso.net;alictag;bobtag </sessionId>
</releaseQosRequest>

(37) releaseQosResponse PAM2 -> P-CSCF2
<releaseQosResponse>
  <result>0</result>
</releaseQosResponse>

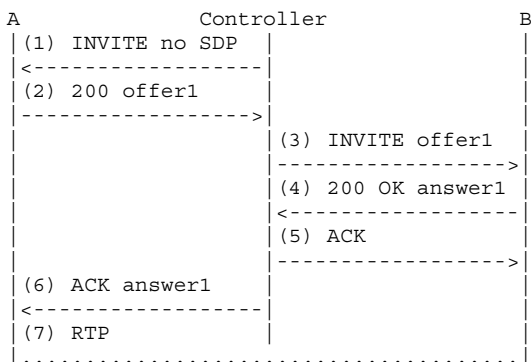
```

I.6 3PCC

In this clause, we will describe how the AM will handle third-party Call Control, specifically those call flow scenarios documented in RFC 3725. Third party call control refers to the ability of one entity to create a call in which communication is actually between other UEs. Third-party call control is possible using the mechanisms specified within the Session Initiation Protocol (SIP). However, there are several possible approaches, each with different benefits and drawbacks. RFC 3725 discusses best current practices for the usage of SIP for third-party call control and presents four call flow scenarios for call establishment. Each of these call flow scenarios is detailed below showing the interaction with the proposed IPAM interface.

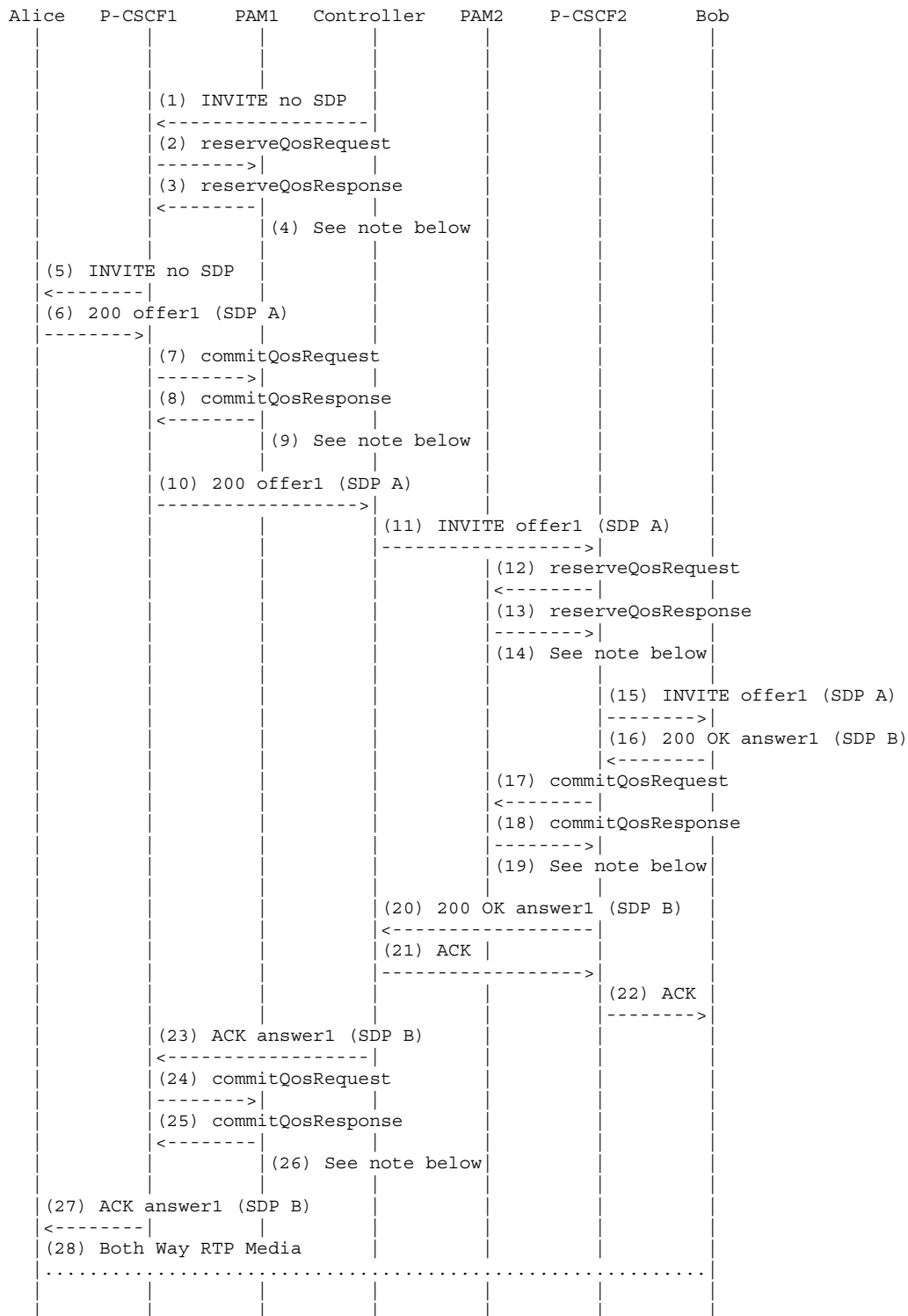
I.6.1 Call flow I

From [RFC 3725] Section 4.1:



The difference between this call flow and the ones described in the clauses above is that the offer is sent in the 200 OK instead of the INVITE.

The following diagram details how this call flow will work with the AM interface.



(2) reserveQosRequest P-CSCF1 -> PAM1

```

<reserveQosRequest>
  <sessionId>1234@mso.net;tag</sessionId>
  <arrayOfPartyInfo>
    <PartyInfo>
      <id>alice@mso.net</id>
      <legId>z9hG4bK74bf9</legId>
      <isLocal>true</isLocal>
      <signalingAddress>Alice IP Address</signalingAddress>
    </PartyInfo>
  </arrayOfPartyInfo>
</reserveQosRequest>
  
```



```

        </arrayOfPartyInfo>
        <emergencyCall>>false</emergencyCall>
</reserveQosRequest>

```

(3) reserveQosResponse PAM1 -> P-CSCF1

```

<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>

```

(4) At this point, the AM does not have any SDP for Alice so it will simply save the supplied information and wait for some SDP before reserving any resources.

The P-CSCF needs to provide Alice's id in this step (as this is when it will look it up). It does not have to provide Alice's IP address given that there was no offer and no reservation is going to happen at this stage. This step can be bypassed as long as the P-CSCF can provide Alice's id at a later stage.

(7) commitQosRequest P-CSCF1 -> PAM1

```

<commitQosRequest>
  <sessionId>1234@mso.net;tag;alicetag</sessionId>
  <arrayOfPartyInfo>
    <legId>z9hG4bK7a3d4</legId>
    <isLocal>>true</isLocal>
    <sdp>SDP A</sdp>
  </arrayOfPartyInfo>
</commitQosRequest>

```

(8) commitQosResponse PAM1 -> P-CSCF1

```

<commitQosResponse>
  <result>0</result>
</commitQosResponse>

```

(9) PAM1 now has the offer from Alice and can go ahead and reserve resources as it did in the previous examples. It will assume that the answer will match the offer in order to compute the flowspec values. The gates will be in the "reserved" state as there is no answer to the offer yet.

(12) reserveQosRequest P-CSCF2 -> PAM2

```

<reserveQosRequest>
  <sessionId>5679@mso.net;tag</sessionId>
  <arrayOfPartyInfo>
    <PartyInfo>
      <isLocal>>false</isLocal>
      <sdp>SDP A</sdp>
    </PartyInfo>
    <PartyInfo>
      <id>bob@mso.net</id>
      <legId>z9hG4bK7a3d4</legId>
      <isLocal>>true</isLocal>
      <signalingAddress>Bob IP Address</signalingAddress>
    </PartyInfo>
  </arrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>

```

(13) reserveQosResponse PAM2 -> P-CSCF2

```

<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>

```

(14) This will look like a normal call where the offer is in the INVITE, so the AM will behave as it did in the basic call. PAM2 will create gates for Bob in the "reserved" state by using SDP A and Bob's IP as the subscriber id.

(17) commitQosRequest P-CSCF2 -> PAM2

```

<commitQosRequest>
  <sessionId>5678@mso.net;tag;bobtag</sessionId>
  <arrayOfPartyInfo>
    <legId>z9hG4bK7a3d4</legId>
    <isLocal>>true</isLocal>
    <sdp>SDP B</sdp>
  </arrayOfPartyInfo>
</commitQosRequest>

```

```
(18) commitQosResponse PAM2 -> P-CSCF2
<commitQosResponse>
  <result>0</result>
</commitQosResponse>
```

(19) PAM2 will commit the gates and update the flowspec and classifiers now that it has the answer.

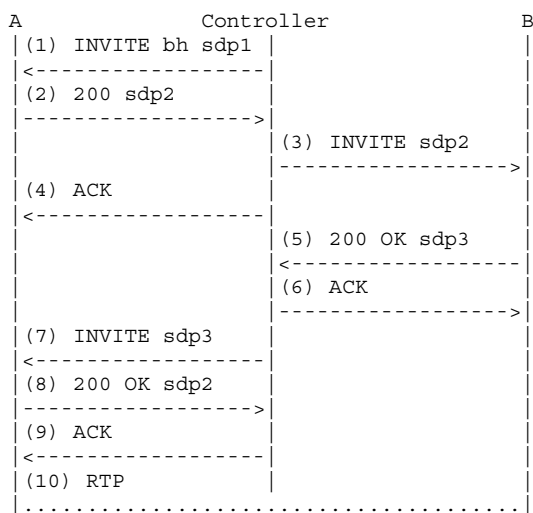
```
(24) commitQosRequest P-CSCF1 -> PAM1
<commitQosRequest>
  <sessionId>1234@mso.net;tag;alicetag</sessionId>
  <arrayOfPartyInfo>
    <isLocal>>false</isLocal>
    <sdp>SDP B</sdp>
  </arrayOfPartyInfo>
</commitQosRequest>
```

```
(25) commitQosResponse PAM1 -> P-CSCF1
<commitQosResponse>
  <result>0</result>
</commitQosResponse>
```

(26) PAM1 will modify the created gates and update the flowspec and classifiers now that it has the answer. If a media is rejected, the corresponding gates will be deleted.

I.6.2 Call flow II

From [RFC 3725] Section 4.2:

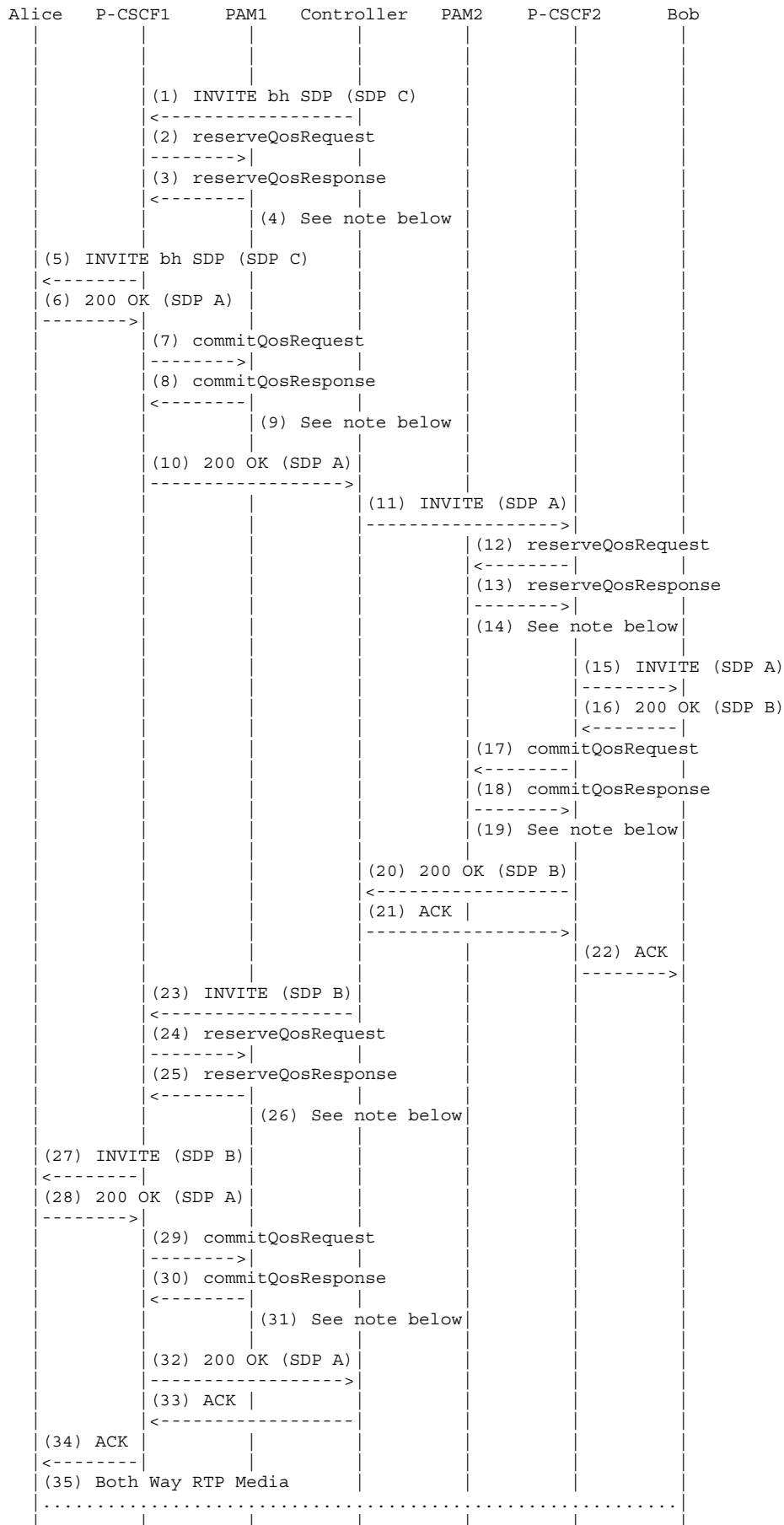


The controller first sends an INVITE to user A (1). This is a standard INVITE, containing an offer (sdp1) with a single audio media line, one codec, a random port number (but not zero), and a connection address of 0.0.0.0. This creates an initial media stream that is "black holed", since no media will flow from A. The INVITE causes A's phone to ring.

When A answers (2), the 200 OK contains an answer, sdp2, with a valid address in the connection line. The controller sends an ACK (4). It then generates a second INVITE (3). This INVITE is addressed to user B, and it contains sdp2 as the offer to B.

This INVITE causes B's phone to ring. When it answers, it generates a 200 OK (5) with an answer, sdp3. The controller then generates an ACK (6). Next, it sends a re-INVITE to A (7) containing sdp3 as the offer.

The following diagram details how this call flow will work with the AM interface:



(2) reserveQosRequest P-CSCF1 -> PAM1

```
<reserveQosRequest>
  <sessionId>1234@mso.net;tag</sessionId>
  <arrayOfPartyInfo>
    <PartyInfo>
      <isLocal>>false</isLocal>
      <sdp>SDP C</sdp>
    </PartyInfo>
    <PartyInfo>
      <id>alice@mso.net</id>
      <legId>z9hG4bK7a3d4</legId>
      <isLocal>>true</isLocal>
      <signalingAddress>Alice IP Address</signalingAddress>
    </PartyInfo>
  </arrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>
```

(3) reserveQosResponse PAM1 -> P-CSCF1

```
<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>
```

(4) Given that SDP C contains a black-hole address, no gates will be reserved at this stage; however, information about the UEs' in the call will be saved.

(7) commitQosRequest P-CSCF1 -> PAM1

```
<commitQosRequest>
  <sessionId>1234@mso.net;tag;alicetag</sessionId>
  <arrayOfPartyInfo>
    <legId>z9hG4bK7a3d4</legId>
    <isLocal>>true</isLocal>
    <sdp>SDP A</sdp>
  </arrayOfPartyInfo>
</commitQosRequest>
```

(8) commitQosResponse PAM1 -> P-CSCF1

```
<commitQosResponse>
  <result>0</result>
</commitQosResponse>
```

(9) At this stage, the AM has the answer to the initial offer. It will set gates only in one direction (Alice's downstream as Alice will not be sending any media). The source IP address in the Gate-Set classifiers will be wildcarded at this stage.

(12) reserveQosRequest P-CSCF2 -> PAM2

```
<reserveQosRequest>
  <sessionId>5679@mso.net;tag</sessionId>
  <arrayOfPartyInfo>
    <PartyInfo>
      <isLocal>>false</isLocal>
      <sdp>SDP A</sdp>
    </PartyInfo>
    <PartyInfo>
      <id>bob@mso.net</id>
      <legId>z9hG4bK7a3d4</legId>
      <isLocal>>true</isLocal>
      <signalingAddress>Bob IP Address</signalingAddress>
    </PartyInfo>
  </arrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>
```

(13) reserveQosResponse PAM2 -> P-CSCF2

```
<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>
```

(14) PAM2 will behave as a basic call handling here. It will reserve resources based on SDP A and Bob's IP.

(17) commitQosRequest P-CSCF2 -> PAM2

```
<commitQosRequest>
  <sessionId>5678@mso.net;tag;bobtag</sessionId>
  <arrayOfPartyInfo>
    <legId>z9hG4bK7a3d4</legId>
```

```

        <isLocal>true</isLocal>
        <sdp>SDP B</sdp>
    </ArrayOfPartyInfo>
</commitQosRequest>

```

(18) **commitQosResponse PAM2 -> P-CSCF2**

```

<commitQosResponse>
    <result>0</result>
</commitQosResponse>

```

(19) At this point, PAM2 will commit all gates and will adjust the flowspec and classifiers based on both UEs' SDP.

(24) **reserveQosRequest P-CSCF1 -> PAM1**

```

<reserveQosRequest>
    <sessionId>1234@mso.net;alicetag;bobtag</sessionId>
    <ArrayOfPartyInfo>
        <PartyInfo>
            <legId>z9hG4bK74bf9</legId>
            <sdp>SDP B</sdp>
        </PartyInfo>
    </ArrayOfPartyInfo>
    <emergencyCall>>false</emergencyCall>
</reserveQosRequest>

```

(25) **reserveQosResponse PAM1 -> P-CSCF1**

```

<reserveQosResponse>
    <result>0</result>
</reserveQosResponse>

```

(26) This is a re-INVITE so there is no need to include Alice's info. PAM1 will behave similarly to any re-invite as defined in the re-invite section above.

(29) **commitQosRequest P-CSCF1 -> PAM1**

```

<commitQosRequest>
    <sessionId>1234@mso.net;tag;alicetag</sessionId>
    <ArrayOfPartyInfo>
        <legId>z9hG4bK7a3d4</legId>
        <isLocal>true</isLocal>
        <sdp>SDP A</sdp>
    </ArrayOfPartyInfo>
</commitQosRequest>

```

(30) **commitQosResponse PAM1 -> P-CSCF1**

```

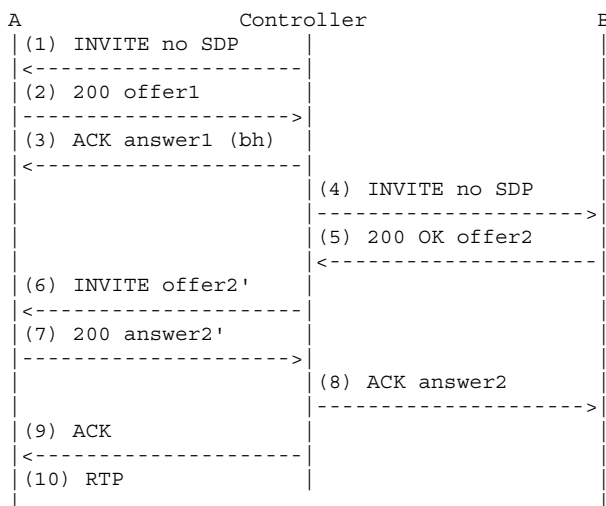
<commitQosResponse>
    <result>0</result>
</commitQosResponse>

```

(31) If a change happened in the new offer/answer, the AM will adjust the reservation according similarly to any re-invite case.

I.6.3 Call flow III

From [RFC 3725] Section 4.3:

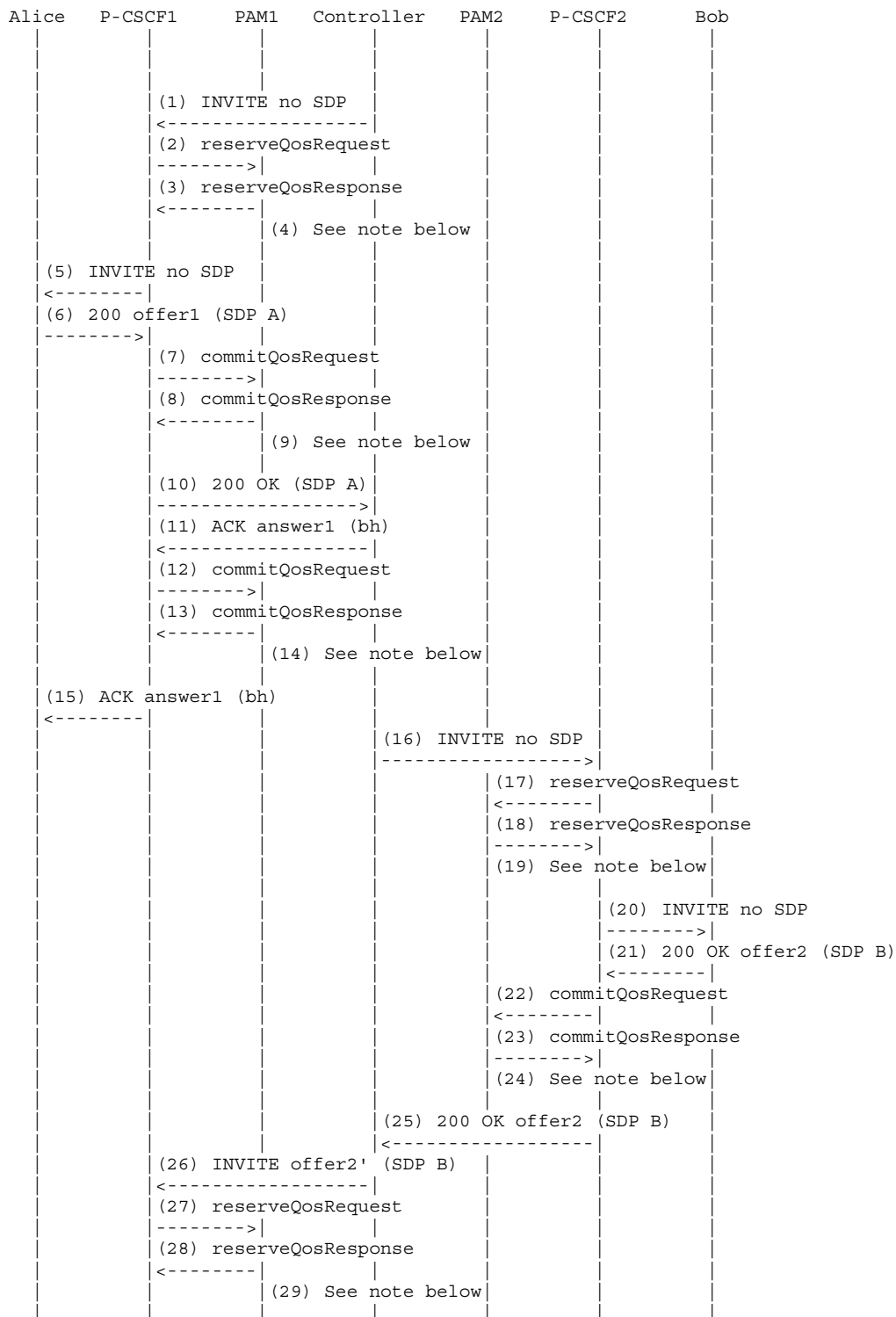


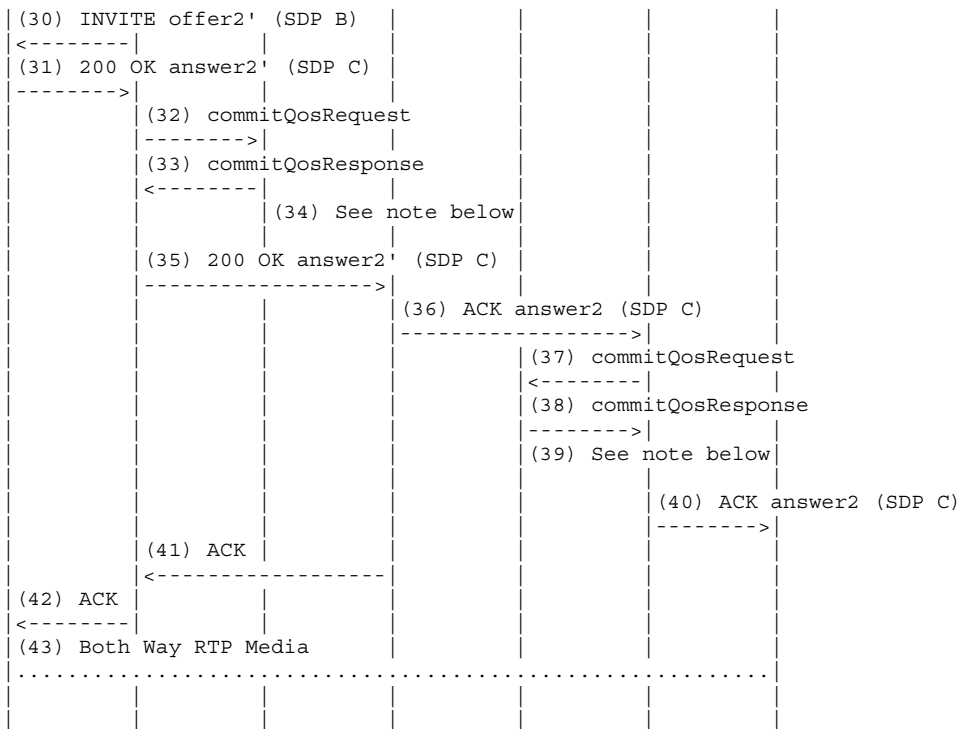
First, the controller sends an INVITE (1) to user A without any SDP. A's phone rings. When A answers, a 200 OK is generated (2) containing its offer, offer1. The controller generates an immediate ACK containing an answer (3). This answer is a "black hole" SDP, with its connection address equal to 0.0.0.0.

The controller then sends an INVITE to B without SDP (4). This causes B's phone to ring. When they answer, a 200 OK is sent, containing their offer, offer2 (5). This SDP is used to create a re-INVITE back to A (6).

The SDP in the 200 OK (7) from A, answer2', may also need to be reorganized or trimmed before sending it in the ACK to B (8) as answer2. Finally, an ACK is sent to A (9), and then media can flow.

The following diagram details how this call flow will work with the AM interface.





(2) reserveQosRequest P-CSCF1 -> PAM1

```

<reserveQosRequest>
  <sessionId>1234@mso.net;tag</sessionId>
  <arrayOfPartyInfo>
    <PartyInfo>
      <id>alice@mso.net</id>
      <legId>z9hG4bK7a3d4</legId>
      <isLocal>true</isLocal>
      <signalingAddress>Alice IP Address</signalingAddress>
    </PartyInfo>
  </arrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</reserveQosRequest>
  
```

(3) reserveQosResponse PAM1 -> P-CSCF1

```

<reserveQosResponse>
  <result>0</result>
</reserveQosResponse>
  
```

(4) At this point, given that PAM1 does not have any SDP, it will simply save the session info and wait for more info before reserving any resources. Note that this step could be bypassed.

(7) commitQosRequest P-CSCF1 -> PAM1

```

<commitQosRequest>
  <sessionId>1234@mso.net;alicetag;tag</sessionId>
  <arrayOfPartyInfo>
    <legId>z9hG4bK7a3d4</legId>
    <isLocal>true</isLocal>
    <sdp>SDP A</sdp>
  </arrayOfPartyInfo>
</commitQosRequest>
  
```

(8) commitQosResponse PAM1 -> P-CSCF1

```

<commitQosResponse>
  <result>0</result>
</commitQosResponse>
  
```

(9) PAM1 now has an SDP from Alice. It will reserve resources as it did in the previous sections, assuming that the answer will match the offer. The gates will be in the reserved state as the answer has not been received yet and it only has a single SDP.

(12) commitQosRequest P-CSCF1 -> PAM1

```

<commitQosRequest>
  <sessionId>1234@mso.net;tag;alicetag</sessionId>
  <arrayOfPartyInfo>
    <legId>z9hG4bK7a3ab</legId>
    <isLocal>>false</isLocal>
  </arrayOfPartyInfo>
</commitQosRequest>
  
```

```

        <sdp>bh</sdp>
    </ArrayOfPartyInfo>
</commitQosRequest>

```

(13) **commitQosResponse PAM1 -> P-CSCF1**

```

<commitQosResponse>
    <result>0</result>
</commitQosResponse>

```

(14) Now that the answer is received, PAM1 will process this as any answer and change the downstream gate to the committed state while keeping the upstream gate in the reserved state. This is because the AM can see from the offer/answer that the answerer does not want to receive media (bh address).

(17) **reserveQosRequest P-CSCF2 -> PAM2**

```

<reserveQosRequest>
    <sessionId>5679@mso.net;tag</sessionId>
    <ArrayOfPartyInfo>
        <PartyInfo>
            <id>bob@mso.net</id>
            <legId>z9hG4bK7a3d4</legId>
            <isLocal>true</isLocal>
            <signalingAddress>Bob IP Address</signalingAddress>
        </PartyInfo>
    </ArrayOfPartyInfo>
    <emergencyCall>>false</emergencyCall>
</reserveQosRequest>

```

(18) **reserveQosResponse PAM2 -> P-CSCF2**

```

<reserveQosResponse>
    <result>0</result>
</reserveQosResponse>

```

(19) PAM2 will simply save the info and not reserve any resources at this stage, given that it does not have any SDP.

(22) **commitQosRequest P-CSCF2 -> PAM2**

```

<commitQosRequest>
    <sessionId>5678@mso.net;tag;bobtag</sessionId>
    <ArrayOfPartyInfo>
        <legId>z9hG4bK7a3d4</legId>
        <isLocal>true</isLocal>
        <sdp>SDP B</sdp>
    </ArrayOfPartyInfo>
</commitQosRequest>

```

(23) **commitQosResponse PAM2 -> P-CSCF2**

```

<commitQosResponse>
    <result>0</result>
</commitQosResponse>

```

(24) PAM2 now has an SDP from Bob. It will reserve resources as it did in the previous sections assuming that the answer will match the offer. The gates will be in the committed state as the call was answered.

(27) **reserveQosRequest P-CSCF1 -> PAM1**

```

<reserveQosRequest>
    <sessionId>1234@mso.net;alicetag;bobtag</sessionId>
    <ArrayOfPartyInfo>
        <PartyInfo>
            <legId>z9hG4bK74bf9</legId>
            <sdp>SDP B</sdp>
        </PartyInfo>
    </ArrayOfPartyInfo>
    <emergencyCall>>false</emergencyCall>
</reserveQosRequest>

```

(28) **reserveQosResponse PAM1 -> P-CSCF1**

```

<reserveQosResponse>
    <result>0</result>
</reserveQosResponse>

```

(29) This will be treated as any re-INVITE (refer to re-INVITE call flows).

(32) **commitQosRequest P-CSCF1 -> PAM1**

```

<commitQosRequest>
    <sessionId>1234@mso.net;tag;alicetag</sessionId>
    <ArrayOfPartyInfo>
        <legId>z9hG4bK7a3d4</legId>

```



```

        <isLocal>true</isLocal>
        <sdp>SDP C</sdp>
    </ArrayOfPartyInfo>
</commitQosRequest>

```

```

(33) commitQosResponse PAM1 -> P-CSCF1
<commitQosResponse>
    <result>0</result>
</commitQosResponse>

```

(34) PAM1 will modify the gates based on the new offer/answer.

```

(37) commitQosRequest P-CSCF2 -> PAM2
<commitQosRequest>
    <sessionId>5678@mso.net;tag;bobtag</sessionId>
    <ArrayOfPartyInfo>
        <legId>z9hG4bK7a3ab</legId>
        <isLocal>true</isLocal>
        <sdp>SDP C</sdp>
    </ArrayOfPartyInfo>
</commitQosRequest>

```

```

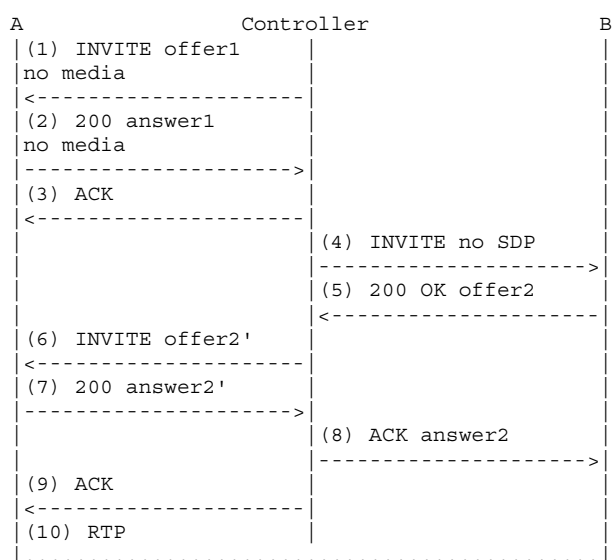
(38) commitQosResponse PAM2 -> P-CSCF2
<commitQosResponse>
    <result>0</result>
</commitQosResponse>

```

(39) At this point, PAM2 has a new SDP and will modify the gates appropriately.

I.6.4 Call flow IV

From [RFC 3725] Section 4.4:



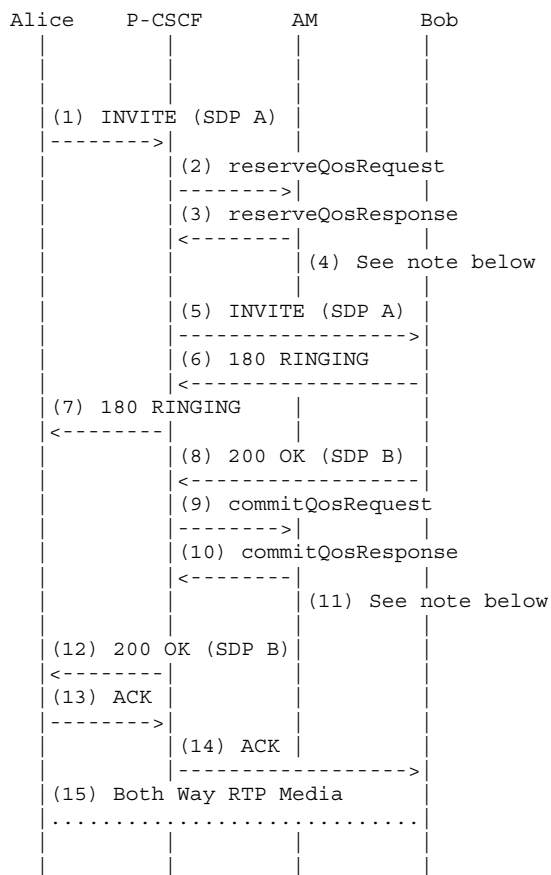
Call flow IV shows a variation on Call flow III that reduces its complexity. The actual message flow is identical, but the SDP placement and construction differs. The initial INVITE (1) contains SDP with no media at all, meaning that there are no m lines. This is valid, and implies that the media make-up of the session will be established later through a re-INVITE. Once the INVITE is received, user A is alerted. When they answer the call, the 200 OK (2) has an answer with no media either. This is acknowledged by the controller (3).

The flow from this point onwards is identical to Call flow III.

Given that the only differences from Call flow III are the first 3 messages, the interaction with the AM will be the same as Call flow III beyond message 3. Before message 3, when the P-CSCF receives the INVITE, it will send a reserveQos request to the AM with the SDP from the INVITE. Given that there is no media in the SDP, the AM will save the session information but will not reserve any resources. When the answer is received (2), the AM will as well only update its session information and will not reserve any resources as no media is present.

I.7 UEs behind same NAT

The following call flow demonstrates how the AM handles the situation where requests are received for a session involving two UEs behind the same NAT, and therefore having the same public IP address. This example also shows that a P-CSCF can optimize the messaging by collapsing messages if it wishes.



(2) reserveQosRequest P-CSCF -> AM

```

<tns:reserveQosRequest>
  <sessionId>1234@mso.net;alicetag</sessionId>
  <arrayOfPartyInfo>
    <tns:PartyInfo>
      <id>alice@mso.net</id>
      <legId>z9hG4bKalice</legId>
      <isLocal>>true</isLocal>
      <sdp>SDP A</sdp>
      <signalingAddress>Alice IP Address</signalingAddress>
    </tns:PartyInfo>
    <tns:PartyInfo>
      <id>bob@mso.net</id>
      <legId>z9hG4bKbob</legId>
      <isLocal>>true</isLocal>
      <signalingAddress>Bob's IP Address</signalingAddress>
    </tns:PartyInfo>
  </arrayOfPartyInfo>
  <emergencyCall>>false</emergencyCall>
</tns:reserveQosRequest>

```

(3) reserveQosResponse AM -> P-CSCF

```

<tns:reserveQosResponse>
  <result>0</result>
</tns:reserveQosResponse>

```

(4) In this case, Alice's and Bob's IP address would be the same; however, the AM can distinguish between the two UEs using two factors: At this stage, Alice has SDP and Bob does not, and Alice's legId is different from Bob's.

```
(9) commitQosRequest P-CSCF -> AM
<tns:commitQosRequest>
  <sessionId>1234@mso.net;alicetag;bobtag</sessionId>
  <arrayOfPartyInfo>
    <legId>z9hG4bKbob</legId>
    <isLocal>true</isLocal>
    <sdp>SDP B</sdp>
  </arrayOfPartyInfo>
</tns:commitQosRequest>
```

```
(10) commitQosResponse AM -> P-CSCF
<tns:commitQosResponse>
  <result>0</result>
</tns:commitQosResponse>
```

(11) The AM will match Bob's legId with the information associated with Bob that it had stored.

Note that going forward, given that the SDP's o= line contains information that uniquely identifies Alice from Bob (and vice versa), the two UEs (even though they share the same signalling addresses) can be clearly disambiguated.

For instance, on a re-INVITE, the AM can know whose SDP is in the new offer and whose SDP is in the answer by matching the SDP's o= line unchanged parameters throughout the session, as defined in [IETF RFC 3264].

If Alice's and Bob's UEs support ICE and are behind the same NAT, they will re-INVITE with their private addresses being in the m/c lines. The AM knows that Alice and Bob share the same signalling address, however, their media addresses are private, thus, the AM could decide to release any resources it had reserved/committed for this dialog as the media will be flowing locally between Alice and Bob and will not require QoS.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems