

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
23 November 2006 (23.11.2006)

PCT

(10) International Publication Number
WO 2006/124790 A2

(51) International Patent Classification:
H04L 12/66 (2006.01) **H04L 12/56** (2006.01)

(21) International Application Number:
PCT/US2006/018758

(22) International Filing Date: 16 May 2006 (16.05.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/681,329 16 May 2005 (16.05.2005) US

(71) Applicant (for all designated States except US): CAMI-
ANT, INC. [US/US]; 5 Mount Royal Avenue, Marlbor-
ough MA 01752 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): MEDLOCK, James
[GB/US]; 91 Brown Road, Harvard, MA 01451 (US).
ABOU-ASSALI, Tarek [FR/US]; 10 Museum Way, Apt.
1223, Cambridge, MA 02141 (US). RILEY, Yusun,
Kim [US/US]; 630 Hemenway Street, Marlborough, MA
01762 (US).

(74) Agents: DEMSHER, Ronald, R. et al.; Wilmer Cutler
Pickering Hale and Door LLP, 60 State Street, Boston, MA
02109 (US).

(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN,
CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI,
GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE,
KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV,
LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI,
NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG,
SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US,
UZ, VC, VN, YU, ZA, ZM, ZW.

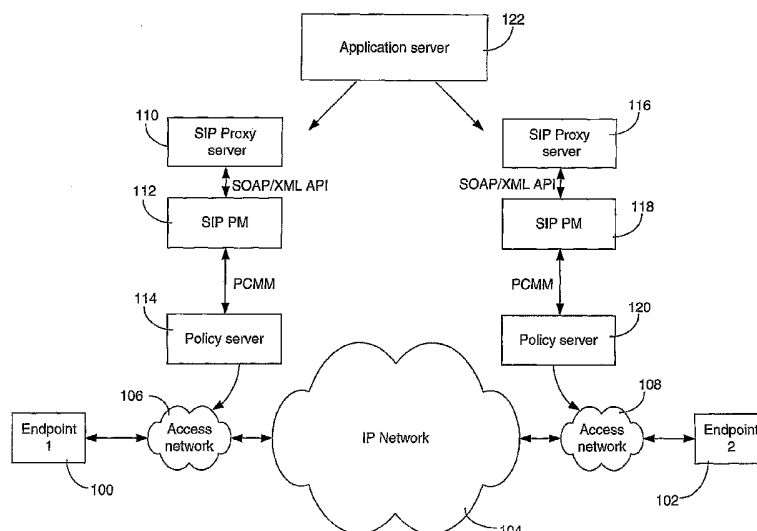
(84) Designated States (unless otherwise indicated, for every
kind of regional protection available): ARIPO (BW, GH,
GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,
ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),
European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,
FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT,
RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA,
GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished
upon receipt of that report

[Continued on next page]

(54) Title: SDP WEB SERVICES INTERFACE



(57) Abstract: A method of allocating network assets for communication between first network endpoint and a second network endpoint includes providing a session initiation invite to a SIP proxy server for initiating a call flow between the network endpoints through the set of network assets. The call flow employs an application residing on an application server. The method also includes providing a resource reservation message from the SIP proxy server to a SIP protocol manager as a result of the session initiation invite. The resource reservation message is provided via a web services interface. The method further includes using the SIP protocol manager to allocate network assets for creating a call flow path connecting the network endpoints. The SIP protocol manager extracts information from the session initiation invite to determine an estimation of required network resources to be allocated.

WO 2006/124790 A2



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SDP WEB SERVICES INTERFACE

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims benefit of the following Patent Applications:
U.S. Provisional Patent Application Serial No. 60/681,329, filed May 16, 2005.

TECHNICAL FIELD

[0002] The present invention relates generally to policy-based network resource management.

BACKGROUND OF THE INVENTION

[0003] In general, a policy server manages network assets according to a set of rules, i.e., a policy. While some policy servers simply carve out network resources (e.g., from router A to router B, an application-based policy server manages network assets with respect to a particular application. For example, in a Voice over Internet Protocol (VoIP) application, two network endpoints may wish to establish a VoIP call with one another via the network. The network endpoints request network assets from the policy server for the VoIP call. The policy server extracts policy-related information from the request, and assigns network assets accordingly.

[0004] FIG. 1 shows one way to implement this VoIP example. A first network endpoint 10 wishes to establish a VoIP call to a second network endpoint 12 via an IP network 14. The first network endpoint 10 connects to the IP network 14 through a first access network 16, and the second network endpoint 12 connects to the IP network 14 through a second access network 18. The first endpoint 10 requests network assets via a request 16 to an application server 20 using, for this example, Session Initiation Protocol (SIP), although other protocols may be used. The application server 20 hosts the VoIP application that the endpoints will use to implement the call. The application server 20 relays the SIP request for a VoIP call to a policy server 22. The application server 20, through the policy server 22, arranges for a particular Quality of Service (QoS) level in the access networks 16 and 18 necessary to implement the VoIP call.

[0005] The arrangement described above requires the application server 18 to maintain the state of the SIP call (i.e., keep track of the actions associated with the call), along with all of the underlying network resources involved in the call. Thus, the application server 18 is required to be “stateful.”

SUMMARY OF THE INVENTION

[0006] In one aspect, a method of allocating network assets for communication between first network endpoint and a second network endpoint includes providing a SIP session initiation invite to a SIP proxy server for initiating a call flow between the network endpoints through the set of network assets. The call flow employs an application residing on an application server. The method further includes providing a resource reservation message from the SIP proxy server to a SIP protocol manager as a result of the session initiation invite. The resource reservation message is provided via a web services interface. The method also includes using the SIP protocol manager to allocate network assets for creating a call flow path connecting the network endpoints.

[0007] One embodiment further includes extracting using SDP information from the session initiation invite to determine an estimation of required network resources to be allocated. The estimation of required network resources includes, for example, QoS information.

[0008] One embodiment further includes providing a request to allocate network assets from the SIP protocol manager to a policy server via a PCMM message. The policy server allocates network resources for the network endpoints. The policy server may include, for example, a first policy server associated with the first network endpoint, and a second policy server associated with the second network endpoint.

[0009] In another embodiment, the SIP proxy server includes a first SIP proxy server associated with the first network endpoint, and a second SIP proxy server associated with the second network endpoint. In another embodiment, the SIP protocol manager includes a first SIP protocol manager associated with the first network endpoint,

and a second SIP protocol manager associated with the second network endpoint.

[0010] One embodiment further includes creating, for each network endpoint, two PCMM gates for each media type supported. One of the PCMM gates is for upstream communication, and one of the PCMM gates is for downstream communication.

[0011] In one embodiment, the SIP protocol manager maintains the mapping of the network resources allocated with the SIP session, thereby abstracting the need for the application server to maintain state of the session to resource mapping.

[0012] Another embodiment further includes allocating network resources for at least one additional network endpoint. The first network endpoint communicates via a call flow path to the second network endpoint and the at least one additional network endpoint.

[0013] In another aspect, a system for allocating network assets for communication between first network endpoint and a second network endpoint includes a SIP proxy server for initiating communication between the network endpoints through the set of network assets. The communication employs an application residing on an application server. The system also includes a SIP protocol manager for receiving a resource reservation message from the SIP proxy server, as a result of the session initiation invite. The resource reservation message is provided via a web services interface. The SIP protocol manager allocates network assets for creating a call flow path connecting the network endpoints.

[0014] In one embodiment, the session initiation invite contains SDP information, and the SIP protocol manager determines an estimation of required network resources to be allocated. The estimation of required network resources includes, for example, QoS information.

[0015] In another embodiment, the SIP protocol manager provides a request to allocate network assets to a policy server via a PCMM message, and wherein the policy server allocates network resources for the network endpoints.

[0016] In one embodiment, the policy server includes a first policy server associated with the first network endpoint, and a second policy server associated with the second network endpoint. In another embodiment, the SIP proxy server includes a first SIP proxy server associated with the first network endpoint, and a second SIP proxy server associated with the second network endpoint.

[0017] In one embodiment, the SIP protocol manager includes a first SIP protocol manager associated with the first network endpoint, and a second SIP protocol manager associated with the second network endpoint. In another embodiment, the SIP protocol manager, for each network endpoint, creates two PCMM gates for each media type supported, wherein one of the PCMM gates is for upstream communication, and one of the PCMM gates is for downstream communication.

[0018] In one embodiment, wherein the SIP protocol manager maintains the mapping of the network resources allocated with the SIP session, thereby abstracting the need for the application server to maintain state of the session to resource mapping.

[0019] Another embodiment further includes at least one additional network endpoint, wherein SIP protocol manager allocates network resources for the at least one additional network endpoint, and the first network endpoint communicates via a call flow path to the second network endpoint and the at least one additional network endpoint.

BRIEF DESCRIPTION OF DRAWINGS

[0020] FIG. 1 shows a prior art architecture for establishing a VoIP link between two network endpoints.

[0021] FIG. 2 shows the described embodiment for allocating network assets for communication between network endpoints.

[0022] FIG. 3 shows the possible error codes returned by the SIP PM in the described embodiment.

[0023] FIGs. 4A and 4B show an exemplary call flow diagram for an on-net successful call.

[0024] FIG. 5 shows an exemplary call flow diagram for an on-net unsuccessful call.

[0025] FIG. 6 shows an exemplary call flow diagram for an off-net call.

[0026] FIG. 7 shows an exemplary call flow diagram for a call hold, after the call has been connected.

[0027] FIG. 8 shows an exemplary call flow diagram for when media is added/removed successfully.

[0028] FIG. 9 shows an exemplary call flow diagram for when media is added/removed unsuccessfully.

[0029] FIG. 10 shows an exemplary call flow diagram for when the additional media requested will not be granted QoS.

[0030] FIG. 11A, 11B, 11C and 11D show an exemplary call flow diagram for forked call.

[0031] FIGs. 12A and 12B describe how the SIP PM handles 3pcc call flow I, as described in RFC 3725.

[0032] FIGs. 13A and 13B describe how the SIP PM handles 3pcc call flow II, as described in RFC 3725.

[0033] FIGs. 14A, 14B and 14C describe how the SIP PM handles 3pcc call flow III, as described in RFC 3725.

DETAILED DESCRIPTION

[0034] The described embodiment is a network architecture that uses a SIP Protocol Module (SIP PM) in conjunction with a QoS-enabled SIP proxy server to establish, maintain and terminate QoS resources in the access networks, for network endpoints that wish to communicate via the network. The SIP PM abstracts the procedures and functionality necessary to implement SIP communication sessions away from the application server, so that the application server does not need to be involved with the intricacies of those sessions.

[0035] This embodiment describes two endpoints implementing a high quality video teleconference session, although these concepts can apply also to other applications, such as video streaming, Voice over Internet Protocol (VoIP) communications, and networked gaming.

[0036] FIG. 2 shows the described embodiment, including a first network endpoint 100, a second network endpoint 102, an IP network 104, a first access network 106 and a second access network 108. In this embodiment, the first network endpoint 100 implements a video teleconference session with the second network endpoint 102 through the access networks 106, 108 and the IP network 104.

[0037] The embodiment in FIG. 2 further includes a first SIP proxy server 110, a first SIP protocol module (SIP PM) 112 and a first policy server 114. These three components operate together to reserve and maintain QoS resources in the first access network 106 for the first network endpoint 100.

[0038] The embodiment in FIG. 2 also includes a second SIP proxy server 116, a second SIP protocol module (SIP PM) 118 and a second policy server 120. These three components operate together to reserve and maintain QoS resources in the second access network 108 for the second network endpoint 102. The operation of these three components is essentially identical to that of the corresponding components associated with the first network endpoint 100, as described below.

[0039] An application server 122 hosts the video teleconference application that the

network endpoints 100, 102 will use to implement the call.

[0040] The first SIP proxy server 116 communicates with the SIP PM via a Simple Object Access Protocol/eXtensible Markup Language (SOAP/XML) Application Programming Interface (API). This interface is also referred to herein as a Web Services Interface (WSI). The SOAP/XML API enables the SIP Proxy Server 116 to request QoS requirements in the access network 106 based on the Session Description Protocol (SDP) parameters contained in the call setup offer and answer, as defined in RFC 3264 ("An Offer/Answer Model with the Session Description Protocol (SDP)," IETF RFC 3264). The SIP PM 112 uses the CableLabs PacketCable Multimedia (PCMM) protocol (PKT-TR-MM-ARCH-V01-030627, V01, June 27, 2003) to communicate those requirements to the policy server 114.

[0041] To establish the video teleconference session, the first network endpoint 100 attempts to signal the second network endpoint 102 to convey its intention to set up a session. The first network endpoint 100 sends an "INVITE" message to its SIP proxy server 110. The first SIP proxy server 110 then invokes the SOAP/XML API to reserve resources via the first SIP PM for the first network endpoint 100, and forwards the "INVITE" to the domain of the second network endpoint 102. A policy server 114 in the domain of the second network endpoint allocates the necessary network resources. The allocated network resources may include QoS.

[0042] Once the "INVITE" reaches the second SIP proxy server 116 associated with the second network endpoint 102, the second SIP proxy server 116 signals to the second SIP PM 118 that resources need to be reserved through a policy server 120 for the second network endpoint 102. This reservation is based initially on an estimation of resources required using the SDP (i.e., the offer) of the first network endpoint 100. The "INVITE" is then forwarded to the second network endpoint 102.

[0043] When the second network endpoint 102 answers the call, a "200 OK" message is sent back to the first network endpoint 100. During this process, the SIP proxy servers 110, 116 in both domains commit the previously reserved QoS, which is modified

to reflect any changes in requirements based on the SDP (i.e., the answer) of the second network endpoint 102.

[0044] Each SIP PM 112, 118 interprets the SDP messages for both UAs, parsing the SDP messages and reading the associated media information, including the media type, media codecs, source and destination IP addresses and ports, and then formulates a PCMM “GateSet” message for the its respective network endpoint.

[0045] In this embodiment, each SIP proxy server (110, 116) is responsible for communicating with its respective SIP PM (112, 118), if at least one of the network endpoints involved in the session is the responsibility of that SIP proxy server. Intermediary proxies along the signaling path, which are not responsible for the network endpoints involved in the session, are not required to communicate with a SIP PM. Each SIP Proxy must signal its respective SIP PM with the offer and answer, as both are needed to completely ascertain the capabilities of the network endpoints.

[0046] For each network endpoint, the associated SIP PM creates two PCMM gates per media type, one in the upstream and one in the downstream direction. In the described embodiment, the network endpoints 100, 102 are SIP videophones. The SDP for these network endpoints can include two media types: audio and video. For each network endpoint, the SIP PM thus creates a total of four PCMM gates:

- An upstream gate for audio;
- A downstream gate for audio;
- An upstream gate for video; and,
- A downstream gate for video.

[0047] When the session is terminated, the terminating network endpoint sends a “BYE” message to the other network endpoint through their associated SIP proxy servers. As the “BYE” message traverses through each proxy server, that proxy server sends a release QoS message to the SIP PM. The release QoS message sends PCMM GateDeletes for gates that were created during the session setup, clearing the service flows that were set up earlier for the network endpoints.

SOAP/XML API SPECIFICATION

[0048] The API characteristics of the SIP PM for the described embodiment are described below in terms of (1) API return status codes, (2) PartyInfo parameter type, and (3) API functions. The SOAP/XML API provides both a single-phase as well as two-phase reserve/commit functionality. The operator controls the mode of operation, but by default operates in two-phase commit mode where a resource is “reserved” on the cable modem termination system (CMTS), and then “committed” when the called party answers.

API Return Status Codes

[0049] The table shown in FIG. 3 describes the possible error codes returned by the SIP PM. Status Codes are used to indicate the success or failure, and the reason for the failure of the API operation.

[0050] Other embodiments include further detail required for failed operations, for example to identify the particular media line causing a gate operation to fail. This information could be returned as a sub-code to one of the codes shown in FIG. 3, or as an additional code.

PartyInfo Parameter Type

[0051] In PCMM and on a CMTS, QoS information for media flows is conveyed/stored in an object called a Gate. A Gate is associated with a subscriber identifier, which is either the active cable modem or customer provided equipment (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 source and destination IP addresses and ports.

[0052] Within the SIP PM API, a parameter describing the endpoint of a SIP session is used to configure the PCMM Gate as well as to provide information used in the execution of business policies and generation of billing events. This parameter is defined

as the object class, "PartyInfo," and contains the following fields:

```
PartyInfo {  
    string id;  
    string sdp;  
    boolean qosEnabled;  
    string ipAddress;  
}
```

[0053] Each of these parameters is defined as follows:

[0054] id: This is a unique identifier for a subscriber. Its format will be: user@domain (i.e. alice@comcast.net). This id MUST be the same for a particular subscriber, no matter what location he/she is at when making/receiving calls. If this field is empty, this PartyInfo will still be accepted as long as the SDP is present (which will allow the SIP PM to uniquely identify the session description of a party within a dialog as defined in RFC 2327 [Session Description Protocol : IETF RFC 2327]). This identifier only needs to be provided in the first API call for a dialog. This field is used, for example, for nomadic users.

[0055] sdp: This is the SDP contained in the offer/answer. If not available, this field defaults to an empty string.

[0056] qosEnabled: This is a boolean flag (i.e., true/false) that informs the SIP PM whether or not this party is within the responsibility domain of the EP and if it is a QoS enabled party(from the EP's perspective). For instance, for a guest, this flag will be set to false, similarly for a party not within the responsibility domain of the EP, this flag will be false. Once this flag is set to true within a session, it will remain as true for the remainder of the session.

[0057] ipAddress: This is the IP address (dotted decimal notation) of the party initiating/receiving a call (this parameter could be used in the subscriber-id field of a PCMM message only in the case where no SDP is available). If not available or not needed (i.e., enough info is in the SDP), this field should default to an empty string.

[0058] For functions requiring information about a number of session endpoints, an array of PartyInfo objects should be provided.

API Functions

[0059] The description below provides details on the SOAP/XML RPC Interface exposed by the QBUS SIP PM.

[0060] **int reserveQos** – This method is called when the SIP proxy server receives a SIP INVITE message, and is used by the SIP PM to initially reserve resources in the access network, ensuring that those resources are available when the network endpoint being called ultimately answers the call. Using this interface, the SIP proxy server has the opportunity to signal back to the caller if QoS is not available in the case where the operator wishes to block calls that will not have associated QoS resources (i.e., a configurable option).

[0061] The following provides the proper syntax and parameters for this function:

```
int reserveQos(  
    string sessionId,  
    PartyInfo[] parties,  
    boolean e911)
```

Parameters:

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 endpoint is issuing a request), it is the responsibility of the SIP PM to 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
```

parties – an array that contains information regarding the parties in the call as described in the previous section.

e911 – a boolean flag indicating if this call is an emergency call. The SIP PM will make sure that the reservation succeeds for these types of calls (even if it has to delete established QoS on other existing calls). By default, this flag is false. It is not mentioned in the remainder of the document as it is assumed “false” in the described scenarios.

Return Value:

Status Code

[0062] **int commitQos** – When the Callee answers and the terminating proxy server receives the 200 OK, it will send a commitQos() request to the SIP PM including the called party SDP. At this stage, the SIP PM has all the info it needs and will commit resources by changing the state of PCMM gates from reserved to committed as well as adjusting the classifiers and QoS resources. Given that resources were reserved at the call initiation stage, the commitment of the resources should succeed (as long as the committed resources do not exceed the reserved ones).

[0063] The following provides the proper syntax and parameters for this function:

```
int commitQos(  
    string sessionId,  
    PartyInfo[] parties)
```

Parameters:

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 endpoint is issuing a request), it is the responsibility of the SIP PM to 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

parties – array as defined above (in this case it will contain the called party's information). It is defined as an array although in all cases described in the document it is an array of length 1. The reason it is an array is in case this interface is between a back-to-back user agent and a SIP PM and the back-to-back user agent would like to provide information about multiple parties in one API call.

Return Value:

Status Code

[0064] **int releaseQos** – This function releases all QoS resources for the specified session. Typically this function is called when the SIP Proxy receives a BYE message from one of the end points.

[0065] The following provides the proper syntax and parameters for this function:

```
int releaseQos(  
    string sessionId)
```

Parameters:

SessionId – the sessionId used in the previous releaseQos() and commitQos() method calls.

Return Value:

Status Code

[0066] **string getVersion** – This function returns the version string of the QBUS SIP PM. The following provides the proper syntax and parameters for this function:

string getVersion()

Parameters:

None

Return Value:

Version string.

Call Flow Examples

[0067] FIGs. 4 through 14 show SIP call flow for on-net, off-net, call hold, re-invites, forked calls and 3PCC scenarios. Each call flow shows both the SIP message exchange as well as SIP PM API calls.

[0068] FIGs. 4A and 4B show an exemplary call flow diagram for an on-net successful call. In this diagram, the correspondence with the components shown in FIG. 2 is as follows: the “caller” corresponds to the first network endpoint 100, the “callee” corresponds to the second network endpoint 102, “EP 1” corresponds to the first SIP proxy server 110, “EP 2” corresponds to the second SIP proxy server 116, “SIPPM 1” corresponds to the first SIP PM 112, and “SIPPM 2” corresponds to the second SIP PM 118.

[0069] FIG. 5 shows an exemplary call flow diagram for an on-net unsuccessful call. The correspondence with the components shown in FIG. 2 is the same as that described above for FIGs. 4A and 4B.

[0070] FIG. 6 shows an exemplary call flow diagram for an off-net (i.e., public switched telephone network – PSTN) call. This call flow example shows an outgoing PSTN call, where early dialogs are created. The correspondence with the components shown in FIG. 2 is the same as that described above for FIGs. 4A and 4B.

[0071] FIG. 7 shows an exemplary call flow diagram for a call hold, after the call has been connected. The correspondence with the components shown in FIG. 2 is the same as that described above for FIGs. 4A and 4B. In this example, after the call had been connected, the caller puts the call on hold, which will result in the caller sending a re-INVITE with a “a=sendonly” attribute at the session level in SDP 3. Another way to put the call on hold could be a 0.0.0.0 address in the connection field. In the answer, the callee replies with SDP 4 containing “a=recvonly” attribute at the session level. Note that the “a=sendonly” attribute could be at the media level, in which case, it will only affect the media to which it belongs.

[0072] FIGs. 8, 9 and 10 illustrate call flows where re-INVITEs add/remove media, change IP addresses, increase/decrease resource requirements, for example. FIG. 8 shows an exemplary call flow diagram for when media is added/removed successfully. In this case, when the caller re-INVITEs the callee, the call flow diagram covers the case where either a media is added or removed, relative to the initial offer/answer. The correspondence with the components shown in FIG. 2 is the same as that described above for FIGs. 4A and 4B.

[0073] FIG. 9 shows an exemplary call flow diagram for when media is added/removed unsuccessfully. In this case, the callee rejects the new offer by sending a “488” message (Not Acceptable Here). This example shows how the SIP PM handles both cases (i.e., addition and removal of media). The correspondence with the components shown in FIG. 2 is the same as that described above for FIGs. 4A and 4B.

[0074] FIG. 10 shows an exemplary call flow diagram for when the additional media requested will not be granted QoS. The correspondence with the components shown in FIG. 2 is the same as that described above for FIGs. 4A and 4B. If an IP

address change happens for any media, the relevant SIP PM will delete the gates associated with the media and create new ones with the new address (as long as the address is not 0.0.0.0, which will fall under a special case (hold), or private (refer to section on ICE)).

[0075] FIG. 11A, 11B, 11C and 11D show an exemplary call flow diagram for forked call, i.e., when the callee has two contacts (callee1 and callee2). The correspondence with the components shown in FIG. 2 is the same as that described above for FIGs. 4A and 4B, except for the addition of a second callee.

[0076] FIGs. 12A and 12B describe how the SIP PM handles 3pcc call flow I, as described in RFC 3725 [Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP): IETF RFC 3725]. The difference between this 3pcc call flow, and the ones described in the sections above, is that the offer is sent in the 200 OK instead of the INVITE. The correspondence with the components shown in FIG. 2 is the same as that described above for FIGs. 4A and 4B, except for the addition of a controller, required by the RFC 3725 specification.

[0077] FIGs. 13A and 13B describe how the SIP PM handles 3pcc call flow II, as described in RFC 3725. The correspondence with the components shown in FIG. 2 is the same as that described above for FIGs. 4A and 4B, except for the addition of a controller, required by the RFC 3725 specification. The controller first sends an INVITE to the caller. 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 the caller. The INVITE causes the caller's phone to ring.

[0078] When caller1 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 the callee, and it contains sdp2 as the offer to the callee.

[0079] This INVITE causes the callee'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 caller (7) containing sdp3 as the offer.

[0080] FIGs. 14A, 14B and 14C describe how the SIP PM handles 3pcc call flow III, as described in RFC 3725. The correspondence with the components shown in FIG. 2 is the same as that described above for FIGs. 4A and 4B, except for the addition of a controller, required by the RFC 3725 specification. First, the controller sends an INVITE (1) to the caller without any SDP. The INVITE causes the caller's phone to ring. When the caller answers, the caller generates a 200 OK (2) that contains the caller's 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.

[0081] The controller then sends an INVITE to the callee without SDP (4). This causes the callee's phone to ring. When the callee answers, the callee sends a 200 OK, containing the callee's offer, offer2 (5). This SDP is used to create a re-INVITE back to the caller (6).

[0082] The SDP in the 200 OK (7) from the caller, answer2', may also need to be reorganized or trimmed before sending it in the ACK to the callee (8) as answer2. Finally, an ACK is sent to the caller (9), and then media can flow.

[0083] For 3pcc call flow IV, as described in RFC 3725, the SIP PM handles the flow in a similar manner to call flow III shown in FIGs. 14A, 14B and 14C. Flow IV includes a variation on Flow III that reduces the flow 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 makeup of the session will be established later through a re-INVITE (*see*, Session Description Protocol: IETF RFC 2327). Once the INVITE is received, the caller is alerted. When the caller answers the call, the 200 OK (2) has an answer with no media either. This controller acknowledges the answer (3). The flow from this point onward is identical to Flow III as described in FIGs. 14A, 14B and 14C.

[0084] Since the only difference with flow III is in the first 3 messages, the

interaction with the SIP PM is the same as Flow III beyond message 3. Before message 3, when the EP receives the INVITE (1), the EP will send a reserveQos request to the SIP PM with the sdp copied from the INVITE. Since there is no media in the SDP, the SIP PM will save the information, but will not reserve any resources. When the answer is received (2), the SIP PM will as well only update its info and will not reserve any resources as no media is present either.

ICE interaction

[0085] Given that an update to the current ICE draft [*see*, Interactive Connectivity Establishment (ICE): NAT Traversal for Multimedia Session Establishment Protocols: draft-ietf-mmusic-ice-04] will mandate the endpoints to send a re-INVITE with the chosen candidate and place it in the "c=" line, the SIP PM will not need to understand the new SDP [4] attribute ("candidate") introduced by ICE [6].

[0086] If in the re-INVITE the address is a private address (which would be different than the initial address provided in the initial offer/answer exchange), the SIP PM will delete any created gates as it would for a re-INVITE with an IP address change and given that the new address is private, it will not create any new gates.

[0087] The invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. The present embodiments are therefore to be considered in respects as illustrative and not restrictive, the scope of the invention being indicated by the appended claims rather than by the foregoing description, and all changes which come within the meaning and range of the equivalency of the claims are therefore intended to be embraced therein.

What is claimed is:

- 1 1. A method of allocating network assets for communication between first
2 network endpoint and a second network endpoint, comprising:
3 providing a SIP session initiation invite to a SIP proxy server for initiating a call
4 flow between the network endpoints through the set of network assets, wherein the call
5 flow employs an application residing on an application server;
6 providing a resource reservation message from the SIP proxy server to a SIP
7 protocol manager as a result of the session initiation invite, wherein the resource
8 reservation message is provided via a web services interface;
9 using the SIP protocol manager to allocate network assets for creating a call flow
10 path connecting the network endpoints.
- 1 2. The method of claim 1, further including extracting SDP information from
2 the session initiation invite to determine an estimation of required network resources to be
3 allocated.
- 1 3. The method of claim 2, wherein the estimation of required network
2 resources includes QoS information.
- 1 4. The method of claim 1, further including providing a request to allocate
2 network assets from the SIP protocol manager to a policy server via a PCMM message,
3 wherein the policy server allocates network resources for the network endpoints.
- 1 5. The method of claim 4, wherein the policy server includes a first policy
2 server associated with the first network endpoint, and a second policy server associated
3 with the second network endpoint.
- 1 6. The method of claim 1, wherein the SIP proxy server includes a first SIP
2 proxy server associated with the first network endpoint, and a second SIP proxy server
3 associated with the second network endpoint.

1 7. The method of claim 1, wherein the SIP protocol manager includes a first
2 SIP protocol manager associated with the first network endpoint, and a second SIP
3 protocol manager associated with the second network endpoint.

1 8. The method of claim 1, further including creating, for each network
2 endpoint, two PCMM gates for each media type supported, wherein one of the PCMM
3 gates is for upstream communication, and one of the PCMM gates is for downstream
4 communication.

1 9. The method of claim 1, wherein the SIP protocol manager maintains the
2 mapping of the network resources allocated with the SIP session, thereby abstracting the
3 need for the application server to maintain state of session to resource mapping.

1 10. The method of claim 1, further including allocating network resources for
2 at least one additional network endpoint, wherein the first network endpoint
3 communicates via a call flow path to the second network endpoint and the at least one
4 additional network endpoint.

1 11. A system for allocating network assets for communication between first
2 network endpoint and a second network endpoint, comprising:
3 a SIP proxy server for initiating communication between the network endpoints
4 through the set of network assets, wherein the communication employs an application
5 residing on an application server;
6 a SIP protocol manager for receiving a resource reservation message from the SIP
7 proxy server, as a result of the session initiation invite, wherein the resource reservation
8 message is provided via a web services interface;
9 wherein the SIP protocol manager allocates network assets for creating a call flow
10 path connecting the network endpoints.

1 12. The system of claim 11, wherein the session initiation invite contains SDP
2 information, and the SIP protocol manager determines an estimation of required network
3 resources to be allocated.

1 13. The system of claim 12, wherein the estimation of required network
2 resources includes QoS information.

1 14. The system of claim 11, wherein the SIP protocol manager provides a
2 request to allocate network assets to a policy server via a PCMM message, and wherein
3 the policy server allocates network resources for the network endpoints.

1 15. The method of claim 14, wherein the policy server includes a first policy
2 server associated with the first network endpoint, and a second policy server associated
3 with the second network endpoint.

1 16. The system of claim 11, wherein the SIP proxy server includes a first SIP
2 proxy server associated with the first network endpoint, and a second SIP proxy server
3 associated with the second network endpoint.

1 17. The system of claim 11, wherein the SIP protocol manager includes a first
2 SIP protocol manager associated with the first network endpoint, and a second SIP
3 protocol manager associated with the second network endpoint.

1 18. The system of claim 11, wherein the SIP protocol manager, for each
2 network endpoint, creates two PCMM gates for each media type supported, wherein one
3 of the PCMM gates is for upstream communication, and one of the PCMM gates is for
4 downstream communication.

1 19. The method of claim 11, wherein the SIP protocol manager monitors the
2 network resources allocated and maintains a state of the communication between the
3 network endpoints, thereby abstracting the state of the communication from the
4 application server.

1 20. The system of claim 11, further including at least one additional network
2 endpoint, wherein SIP protocol manager allocates network resources for the at least one
3 additional network endpoint, and the first network endpoint communicates via a call flow
4 path to the second network endpoint and the at least one additional network endpoint.

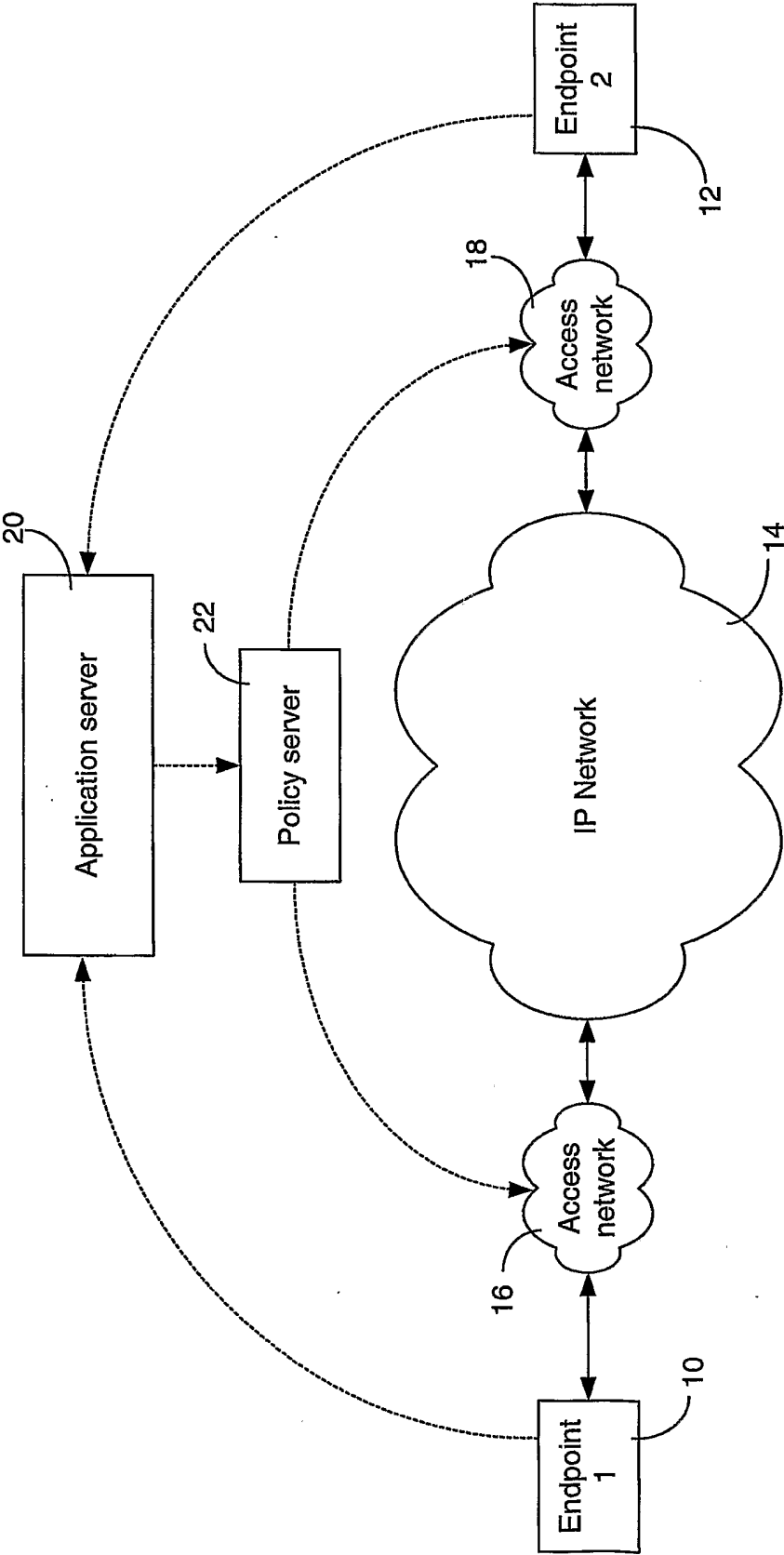


FIG. 1

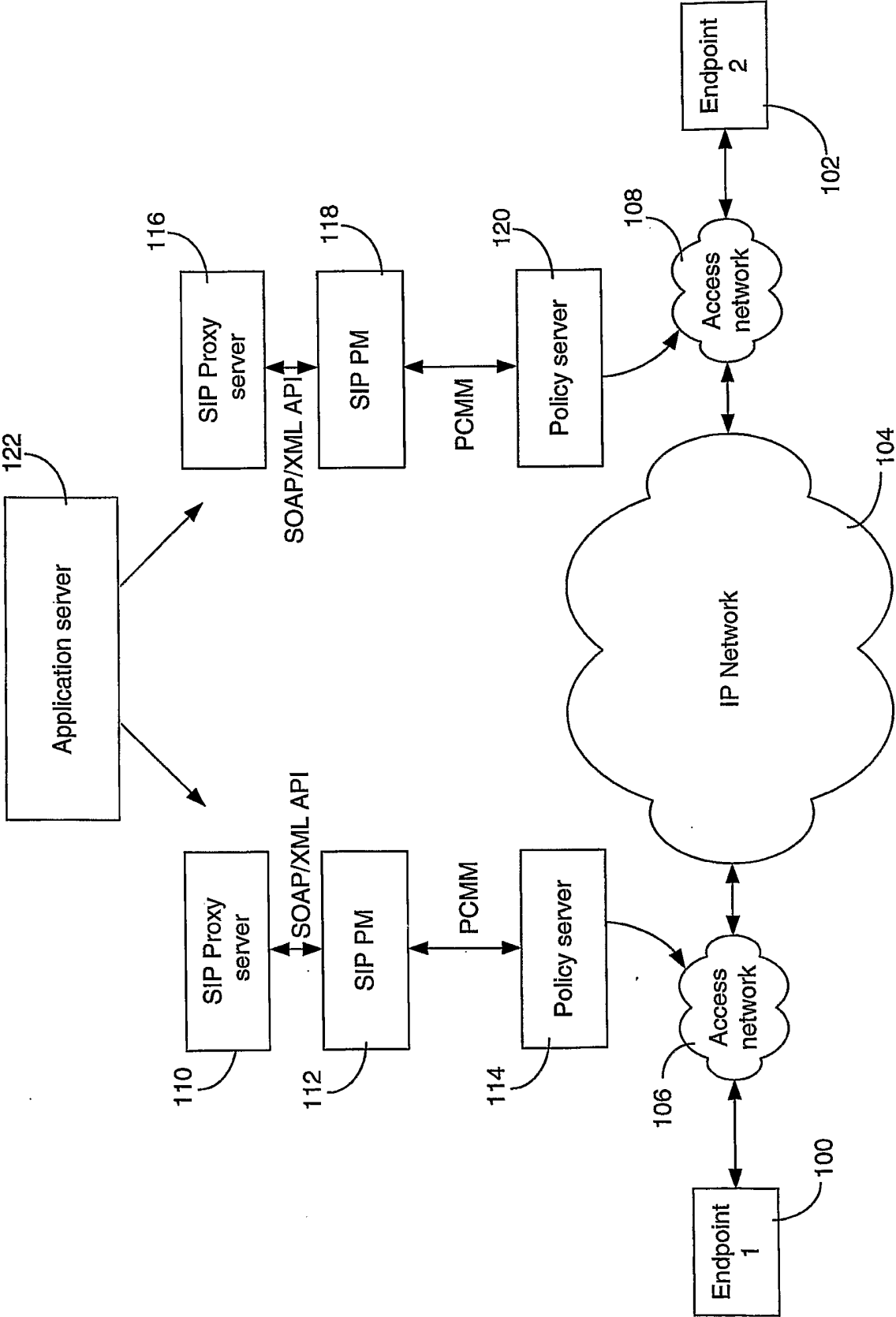
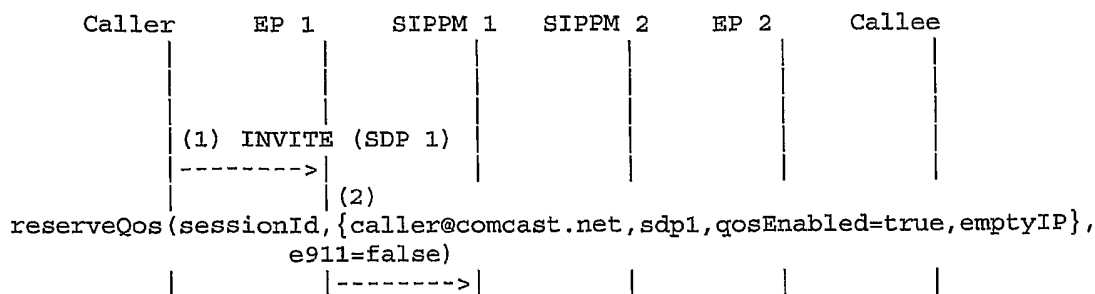


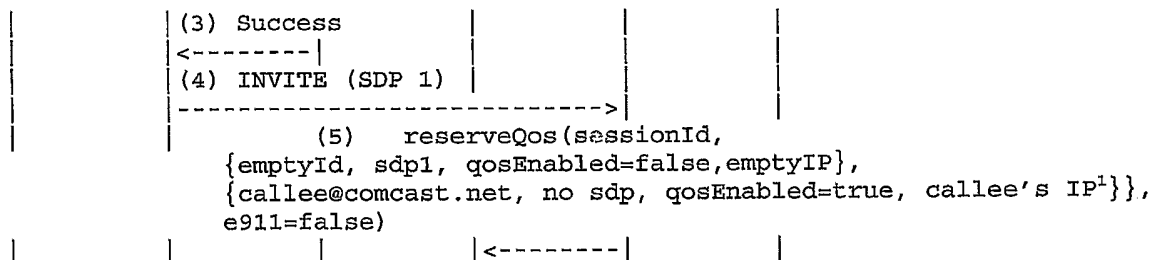
FIG. 2

Status Code	Numeric Value	Description
Success	0	API operation was successful.
Failure	1	General API failure or error.
Fail_Config	2	Error loading/parsing the configuration file.
Fail_Qos_Auth	3	QoS authorization failed.
Fail_Qos_Resource	4	QoS resources not available.
Fail_Qos_Established	5	QoS resources already/still established.
Fail_SessionId	6	Unknown Session ID or Session ID already in use.
Fail_Sdp	7	Error parsing SDP parameters.
Fail_Caller	8	Operation failed due to caller's SDP, or no caller specified.
Fail_Callee	9	Operation failed due to callee's SDP, or no callee specified.
Fail_PS_Comm	10	Failed to reestablished communications to Policy Server.
Fail_IP_Address	11	Invalid IP Address used in reservation.
Fail_Sdp_Type	12	Invalid SDP Protocol Type in SDP.
Fail_GateSet_Response	13	Failed to set a gate.
Fail_GateDelete_Response	14	Failed to delete a gate.
Fail_Too_Many_Sessions	21	Too many concurrent sessions in PS cluster

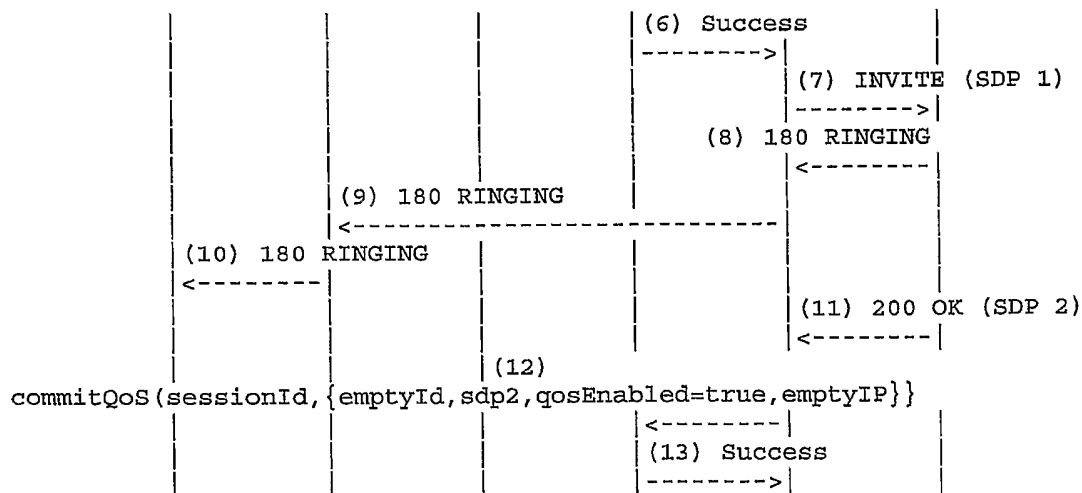
FIG. 3



SIP PM 1 will reserve resources for the caller. It will use the SDP offer for the downstream direction and assume that it is going to be symmetrical for the upstream (this will be adjusted once the answer is received). The address of the callee being unknown at this point, it will be wildcarded in the classifiers.



At this point, the SIP PM 2 will reserve resources for the callee using the same assumptions as SIP PM 1. The only difference is that the subscriber id used in the PCMM Gate-Set messages is the callee's IP. Note that at this stage, the assumption is that the callee's IP address and the media address used by the callee are behind the same cable modem. This information will be adjusted once the answer from the callee is received.



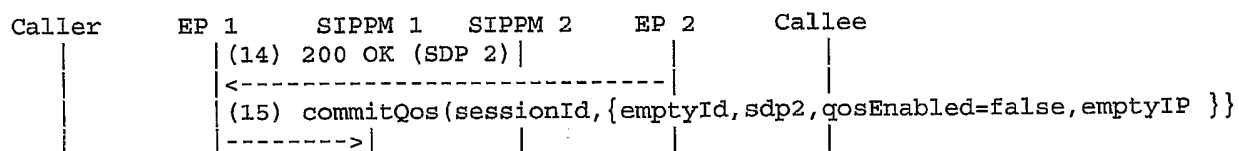
¹ The EP2 can determine the Callee's IP address based on the registration record it has for the callee.

(Continued in FIG. 4B)

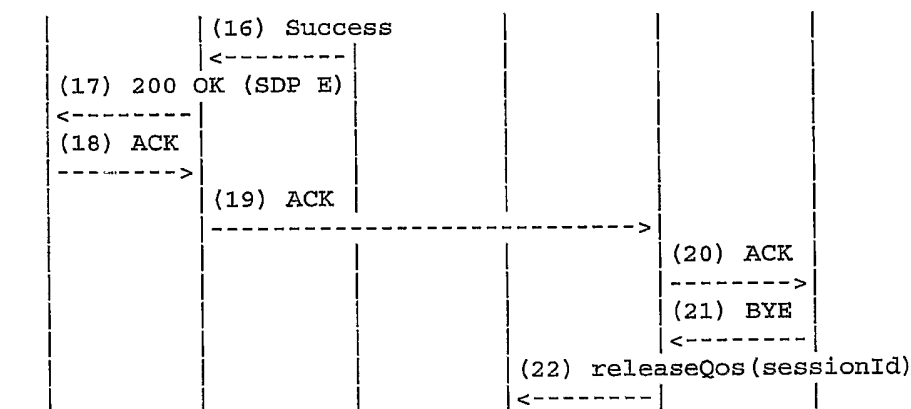
FIG. 4A

(Continued from FIG. 4A)

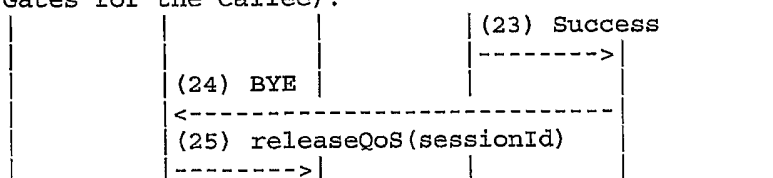
Now that the SIP PM has all the info, it will adjust the gates (flowsecs and classifiers) and move them to the "committed" state. In the case where a media is rejected for instance, gates for the rejected media would be deleted.



SIP PM 1 will adjust the caller's gates as well given that it has now the offer and answer.



At this point, SIP PM 2 will delete all the gates created for this dialog (i.e. Gates for the callee).



At this point, SIP PM 1 will delete all the gates created for this dialog (i.e. Gates for the caller).

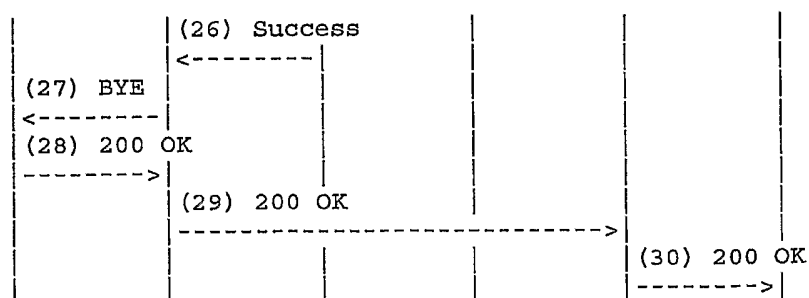
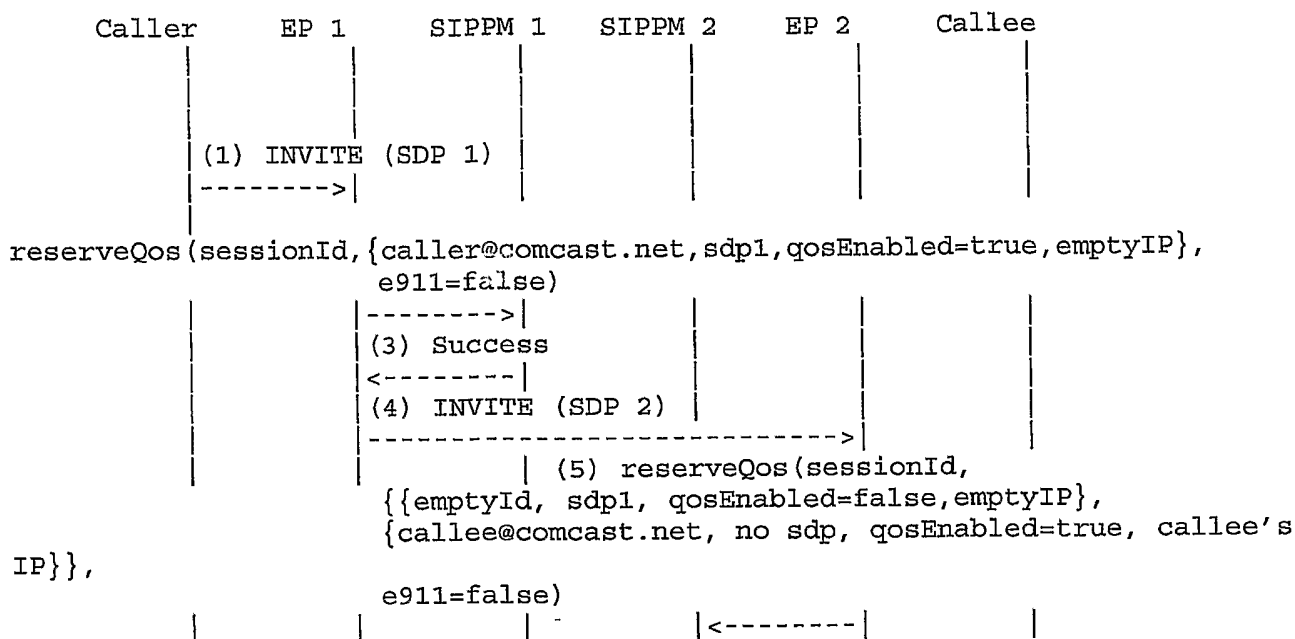


FIG. 4B

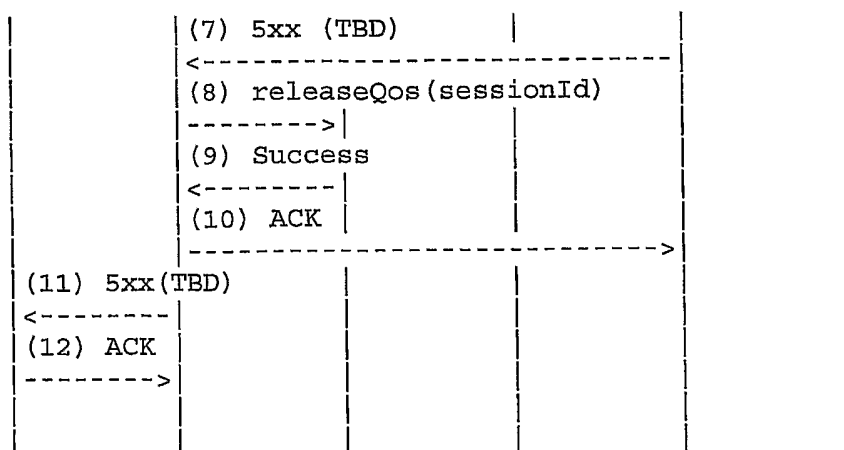


Here the reservation failed, and the error code is passed to the EP.



Given that there was a reservation failure at this point, EP 2 will decide whether to proceed with the call or not (i.e. send a negative final response). This decision should be configurable on the EP. The EP must be operating in synchronous mode (as far as the SOAP interface is concerned), if it is configured to reject calls based on failed qos.

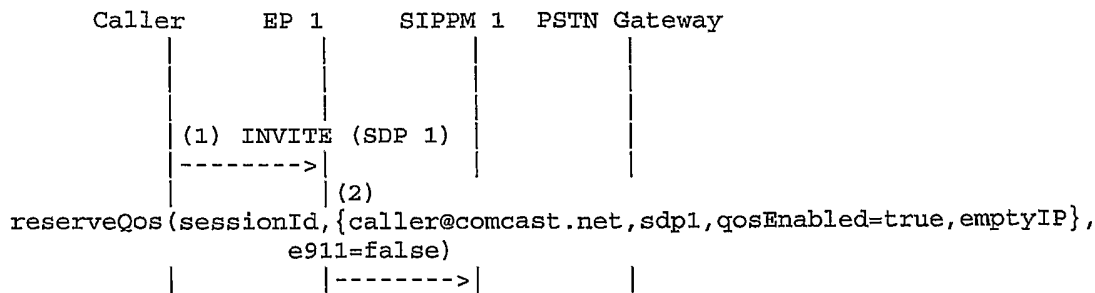
In this case, we are assuming that EP2 is configured to reject calls based on failed qos.



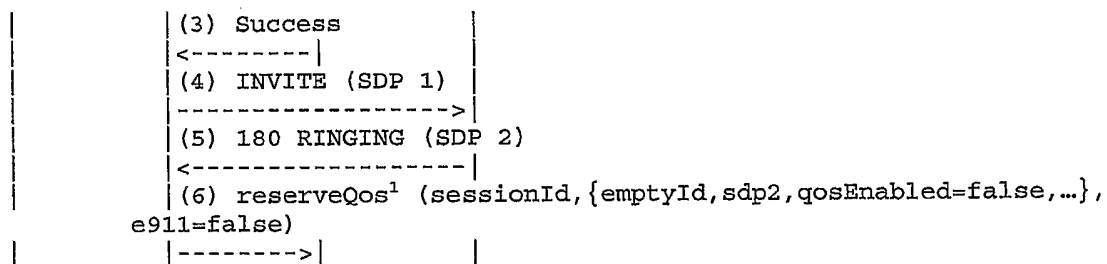
The 5xx response code returned by EP2 needs to be determined.

FIG. 5

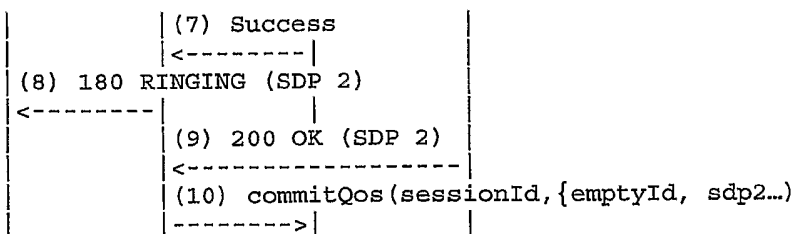
7/22



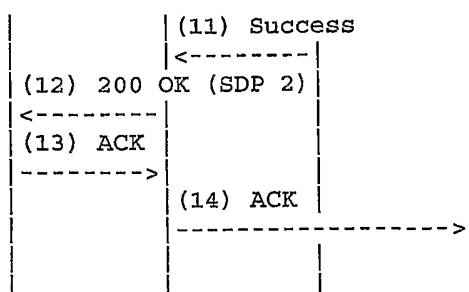
Given that this is an off-net call, the EP doesn't need to provide any information regarding the callee (in this case a PSTN party).



At this stage, the SIP PM will adjust its reservations based on the answer. The gates will still be in the reserved state, unless the EP calls commitQos instead of reserveQos (see footnote).

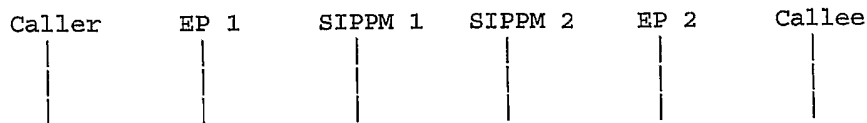


Here resources are committed. If the call in (6) was a commitQos (based on the EP configuration), this step would be a no-op for the SIP PM.

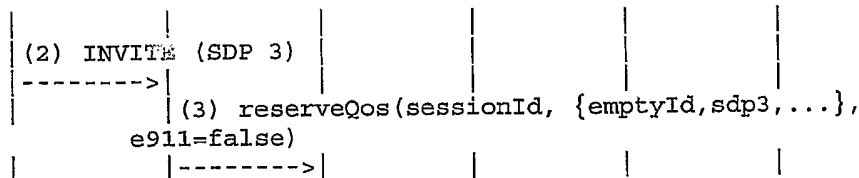


¹ The EP can be configured to either call reserveQos() or commitQos() upon receipt of provisional messages with SDP.

FIG. 6



(1) Call is connected and media is flowing bi-directionally between the caller and the callee



Given that this is a re-INVITE, the EP doesn't include any info regarding the callee(default values as well for the caller's IP as well as the qosEnabled flag).

NOTE: In order to be compliant with ICE [Interactive Connectivity Establishment (ICE): NAT Traversal for Multimedia Session Establishment Protocols: draft-ietf-mmusic-ice-04] and guarantee that NAT bindings don't expire while a call is on hold, the gates will remain in the "committed" state (to guarantee that keep-alive messages sent periodically to maintain NAT bindings get sent/received on time).

For the remainder of the call flow, the SIP PMs take no action, other than saving the new information (new SDP).

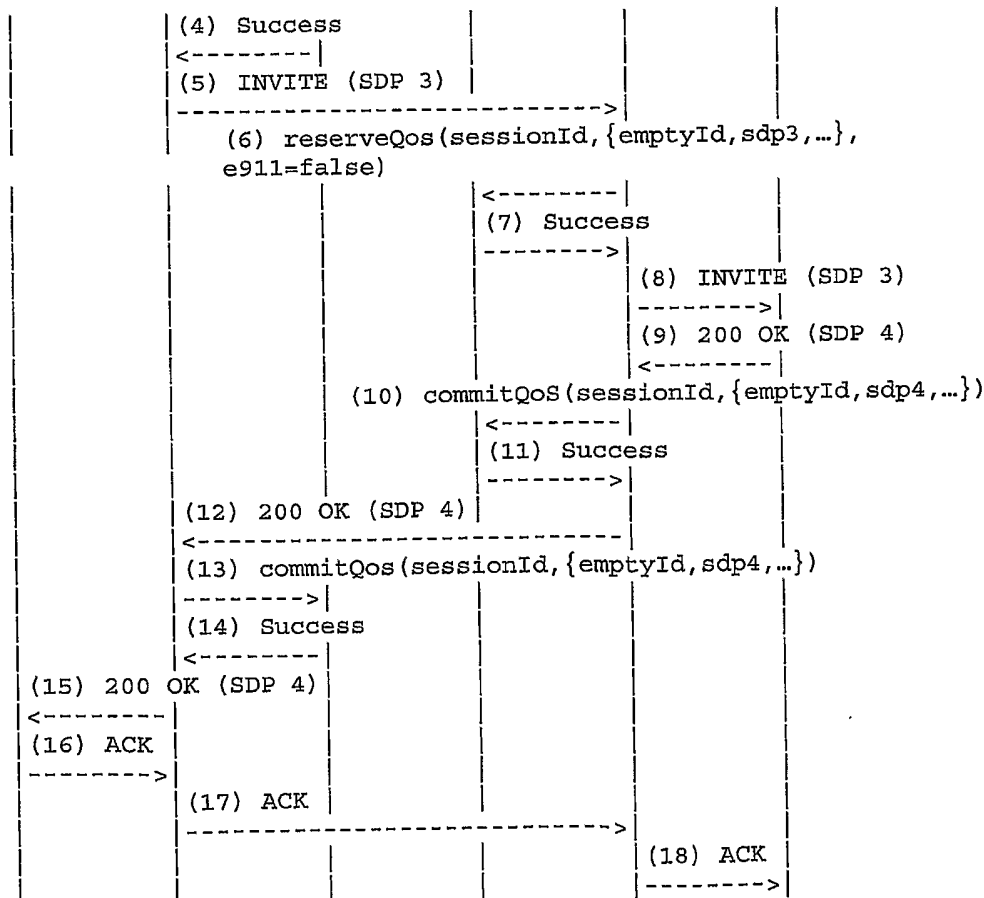
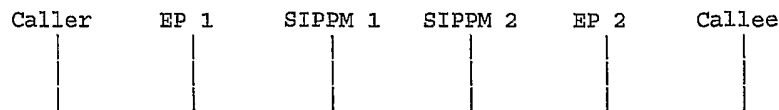
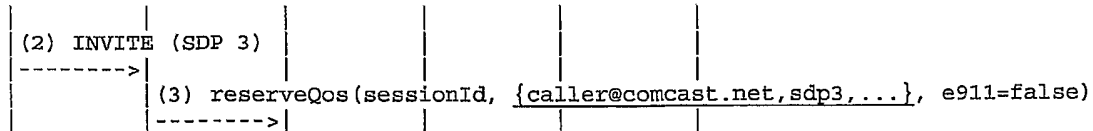


FIG. 7

9/22

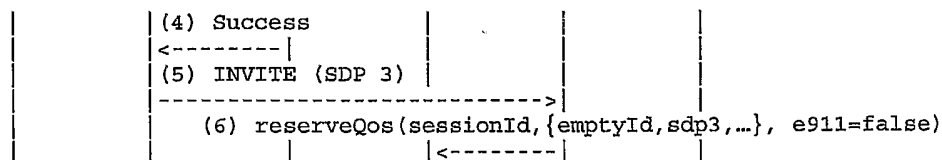


(1) Call is connected and media is flowing bi-directionally between the caller and the callee

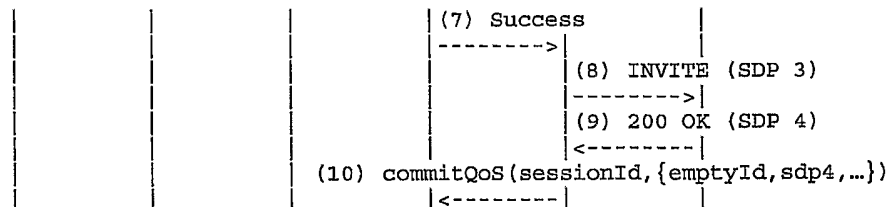


If a media is added, given that SDP 3 differs contains one additional media (relatively to the initial offer), SIP PM 1 will try to reserve this additional resource following the same guidelines as the initial offer. All the existing gates (for media in the initial offer/answer) remain unchanged.

If a media is removed (by setting the port in the "m=" line to zero), SIP PM 1 will save the info as a pending reservation (as in the case where the re-invite fails, it doesn't want to change anything regarding the currently set gates). Once the callee accepts this new offer and replies with a 200 OK, the SIP PM will delete the gates for the corresponding media that was removed.

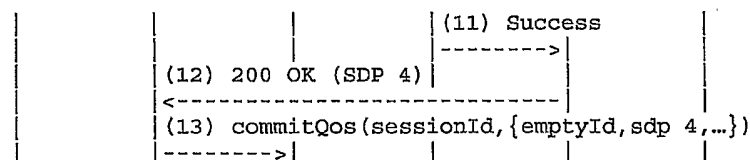


SIP PM 2 will act similarly to SIP PM 1 depending on whether a media is added/removed.



SIP PM 2 will move the gates for the additional media to the "committed" state and adjust the flowspecs/classifiers following the same guidelines as in the initial offer/answer.

If a media was removed, it will delete the corresponding gates.



Similarly to SIP PM 2, SIP PM 1 will adjust the caller's gates.

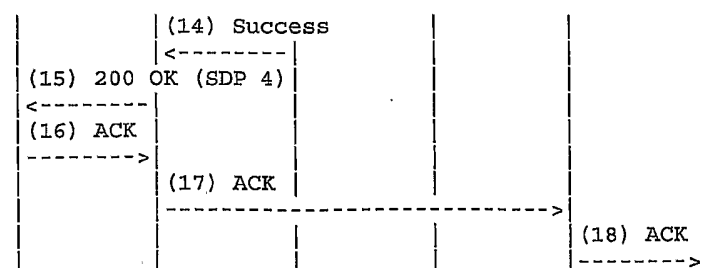
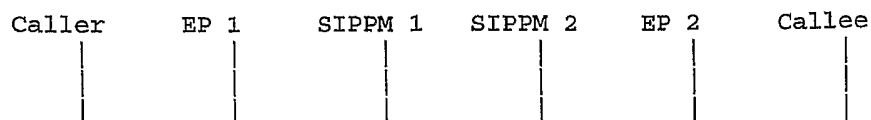
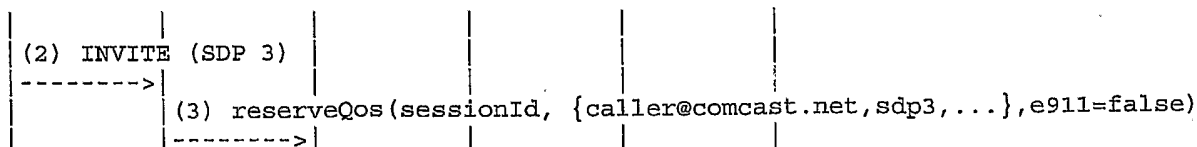


FIG. 8

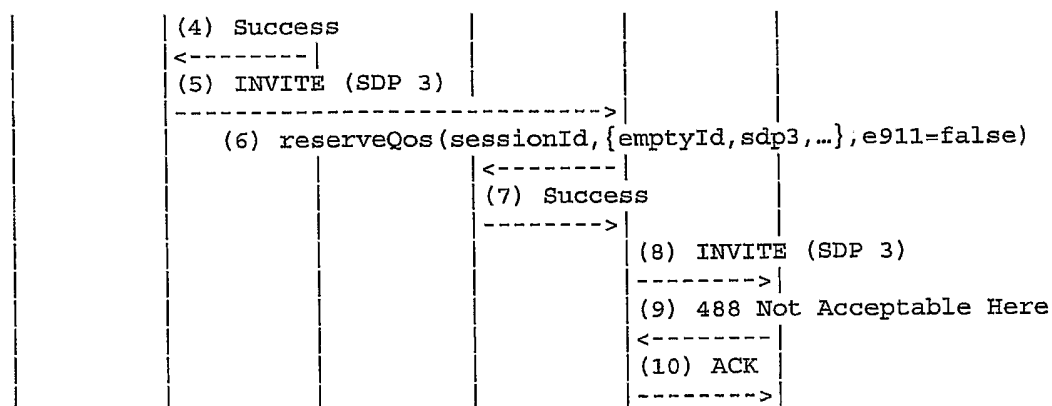
10/22



(1) Call is connected and media is flowing bi-directionally between the caller and the callee

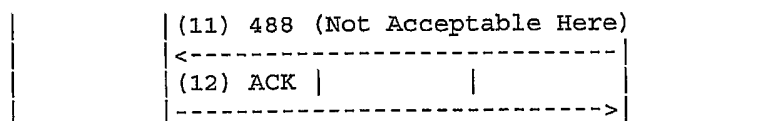


Similar to the previous call flow.



The EPs in this case will not have to inform the SIP PMs of this failure. Reason being, if the caller was trying to remove a media, the SIP PM had not acted on the PCMM side and only has the request pending. The pending request will clear it upon dialog termination or if a new offer/answer is exchanged.

In the case where the SIP PM had reserved resources for the additional media, it will time it out and delete the gate after a certain configured amount of time.



Same here, no action on the EP as far as the SOAP/XML interface is concerned.

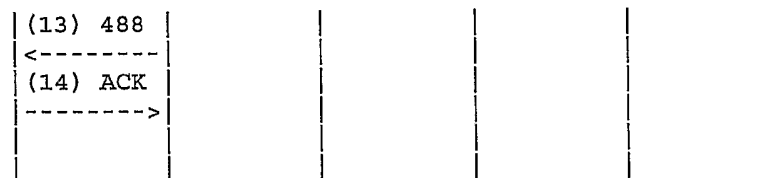


FIG. 9

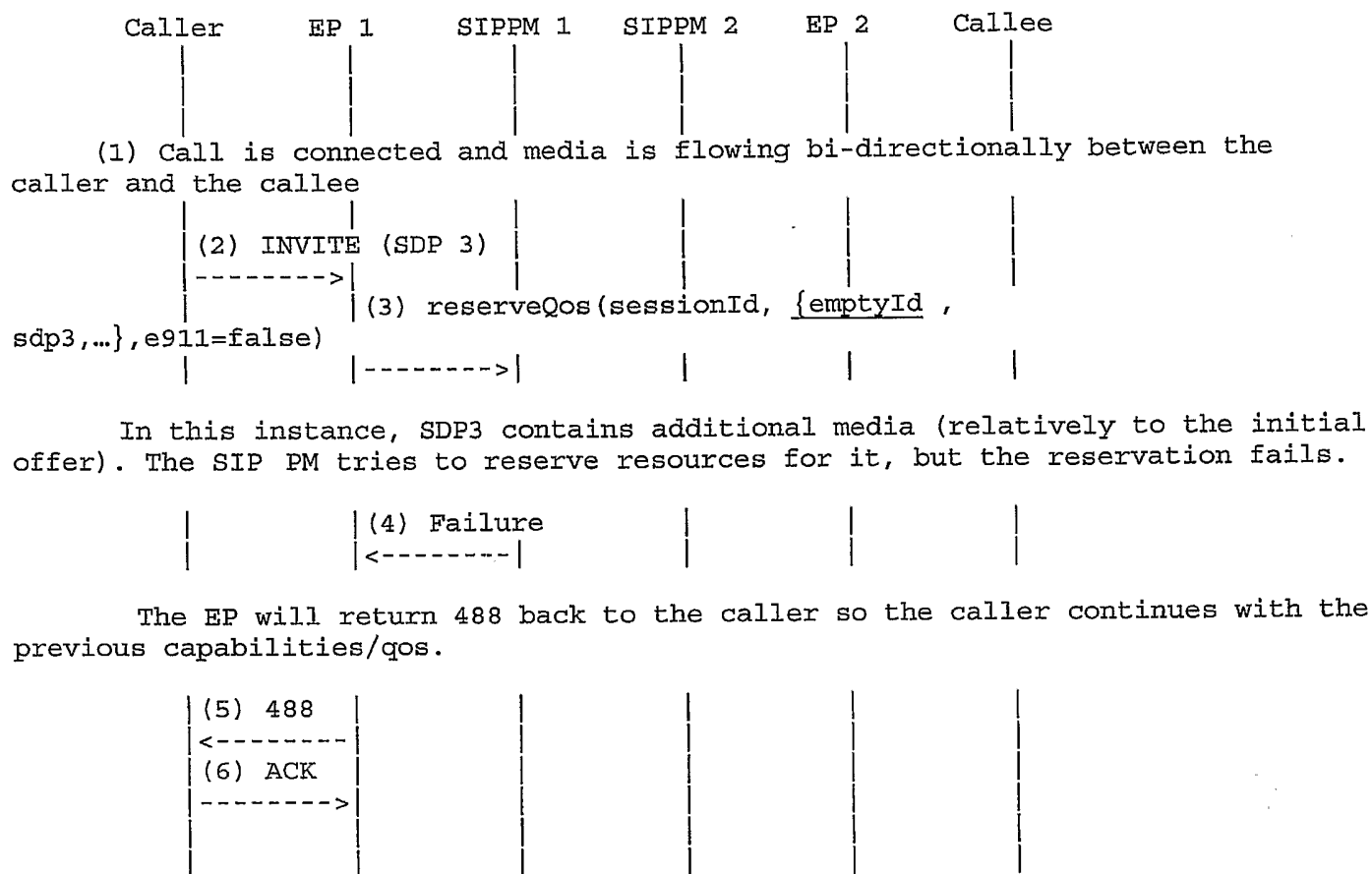
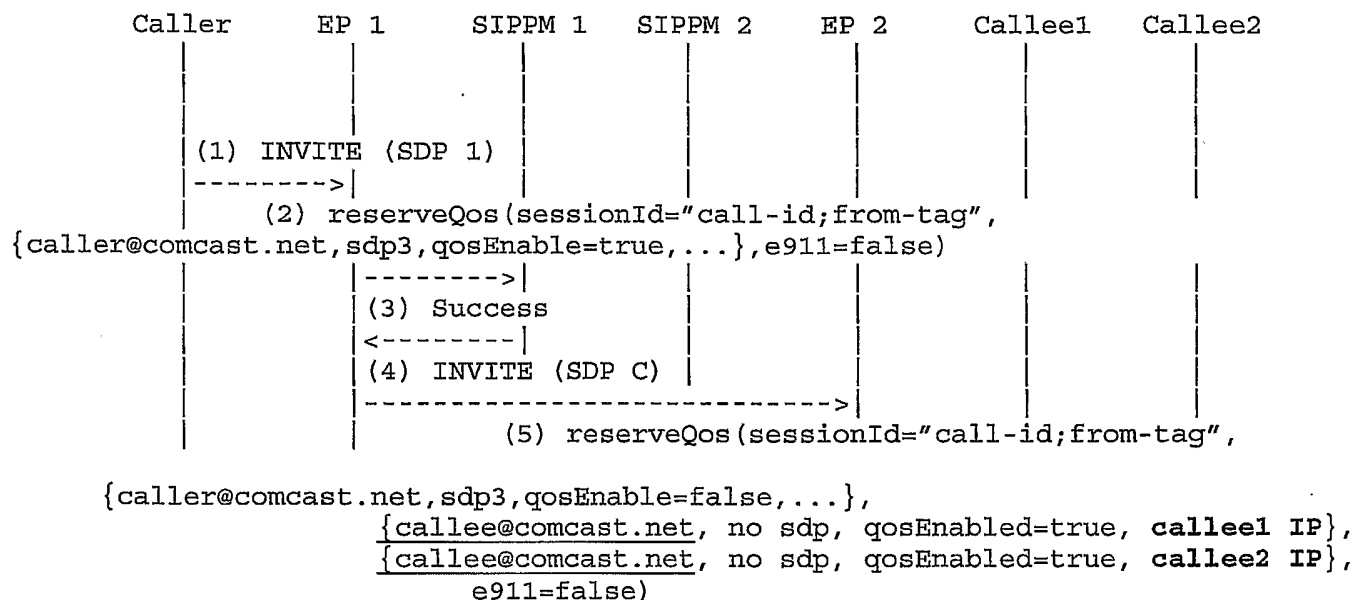
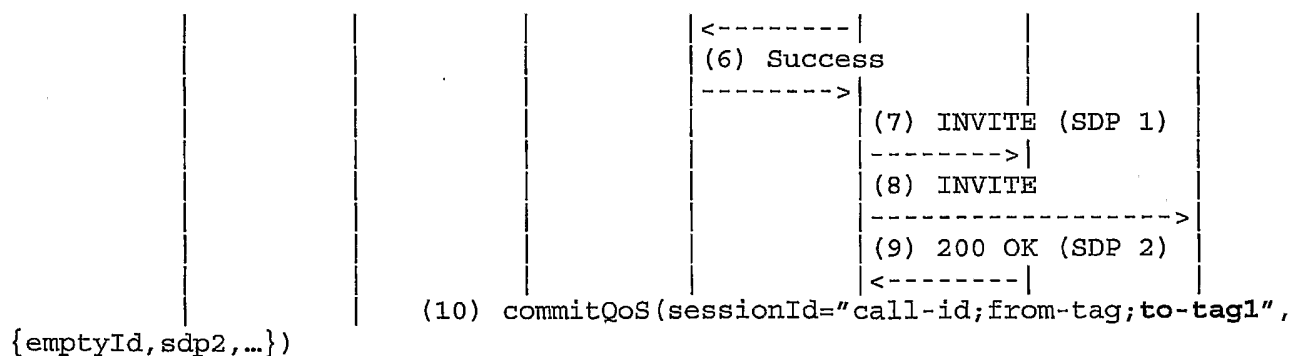


FIG. 10



In this case, given that the callee has multiple contacts (same subscriber), the callee's id is the same in the second and third entry in the partyInfo array, whereas the IP addresses differ.

SIP PM 2 will make reservations as if it had 2 separate dialogs (caller,callee1) and (caller,callee2).



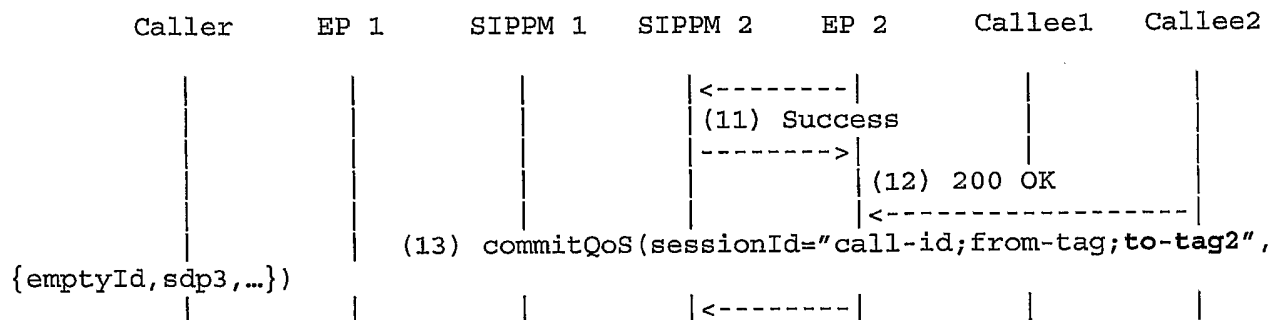
This commitQos is basically confirming the dialog between the caller and the callee1.

SIP PM 2 will adjust the gates for the caller and callee1 and treat this dialog separately from the dialog (that is yet to be created) between the caller and callee2.

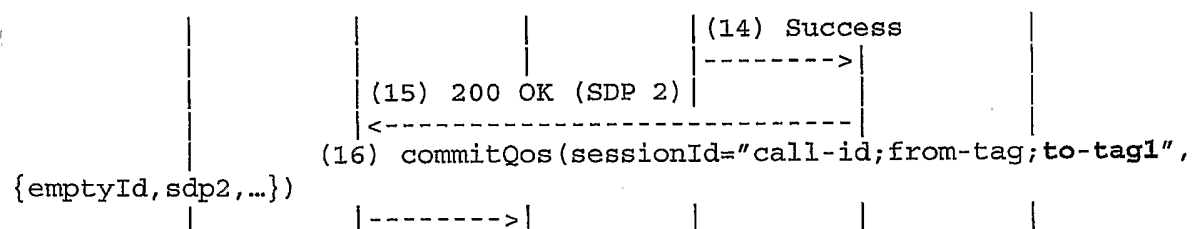
(Continued in FIG. 11B)

FIG. 11A

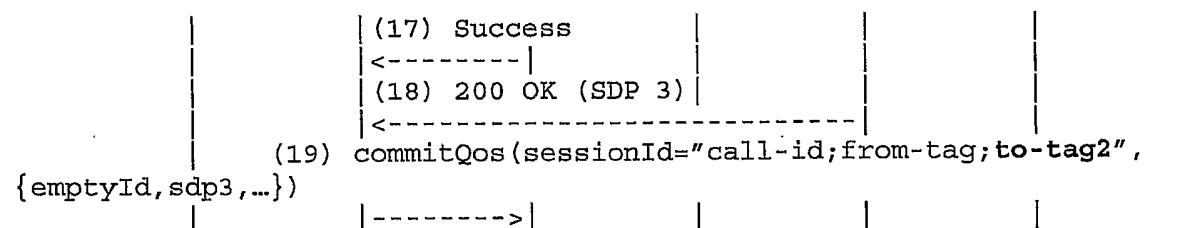
(Continued from FIG. 11A)



Similarly to the previous step, SIP PM 2 will adjust the gates for the dialog between the caller and callee2.



At this stage, SIP PM 1 is unaware of the fact that 2 dialogs will be created. So it will operate as if it did with a non-forked call.



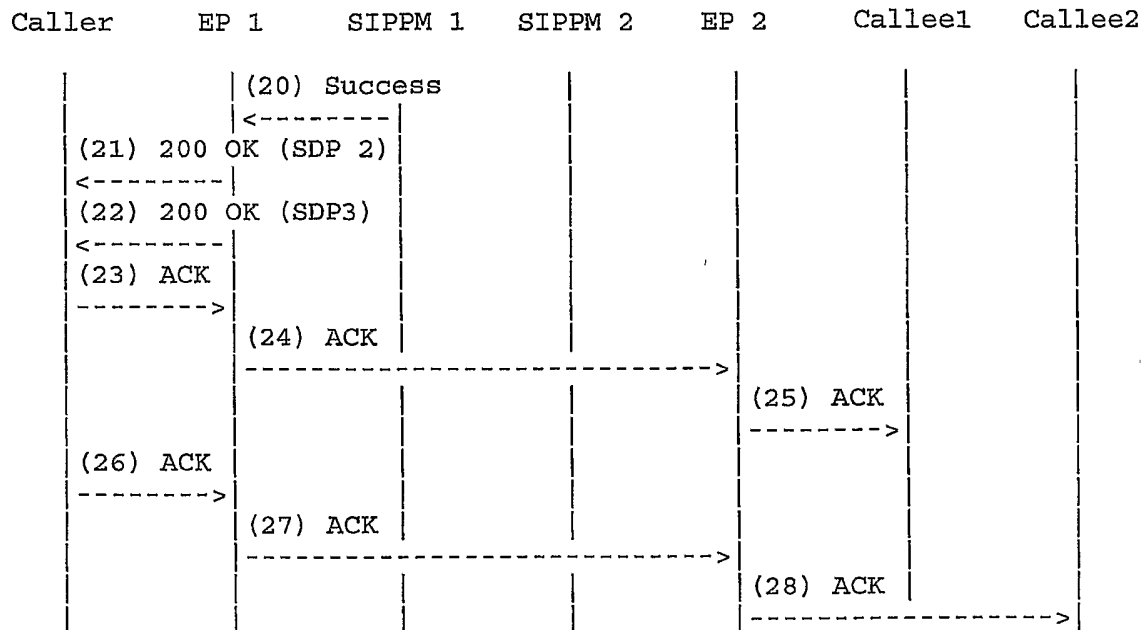
SIP PM 1 now realizes that a new dialog was created (by matching the call-id and from-tag with what it had saved and noticing that the to-tag differs).

(Continued in FIG. 11C)

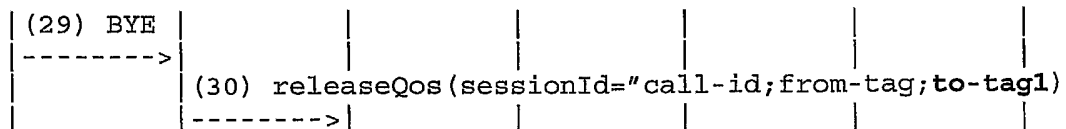
FIG. 11B

(Continued from FIG. 11B)

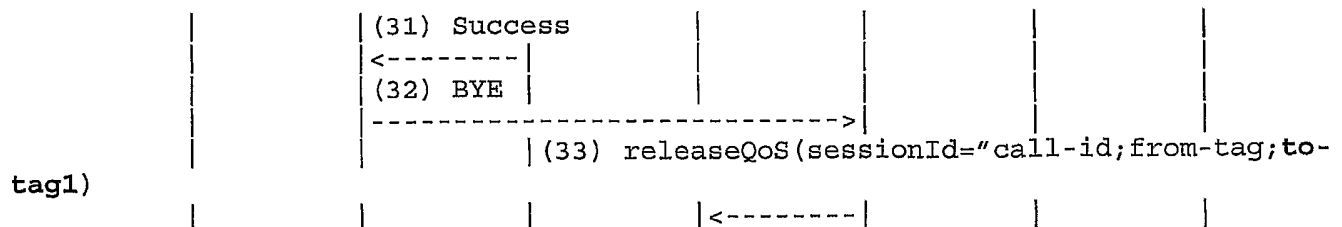
Given that it has the necessary information (caller's and callee2's capabilities), it will create new gates for the caller as if it were a brand new dialog.



The caller decides to hang up with callee1.



SIP PM 1 will delete the gates that were created for the dialog between the caller and callee1 only (the gates created for the dialog between the caller and callee2 will remain unchanged).

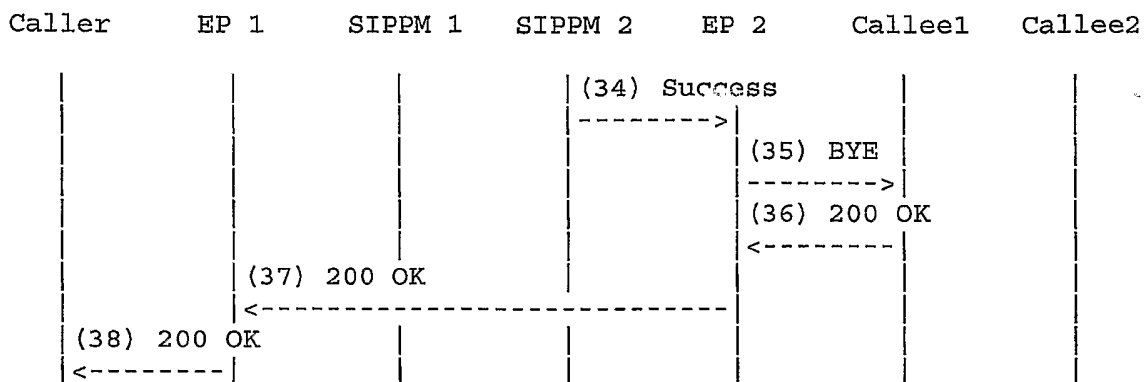


(Continued in FIG. 11D)

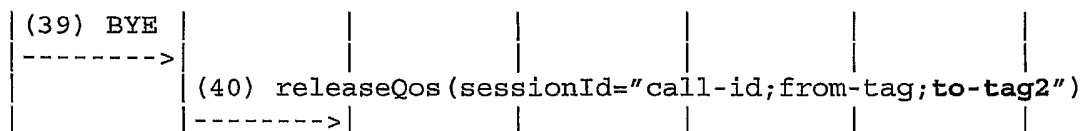
FIG. 11C

(Continued from FIG. 11C)

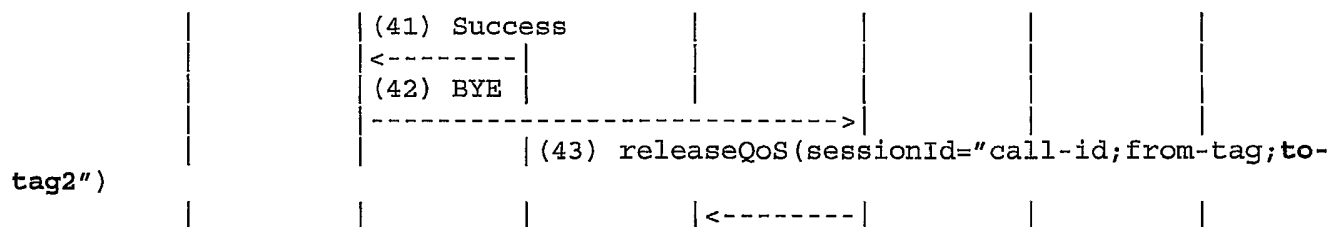
SIP PM 2 will delete the gates for callee1.



The caller hangs up with callee2.



Gates for the dialog between the caller and callee2 will be deleted.
At this point, all gates for the caller have been deleted.



SIP PM 2 will delete the gates for callee2.

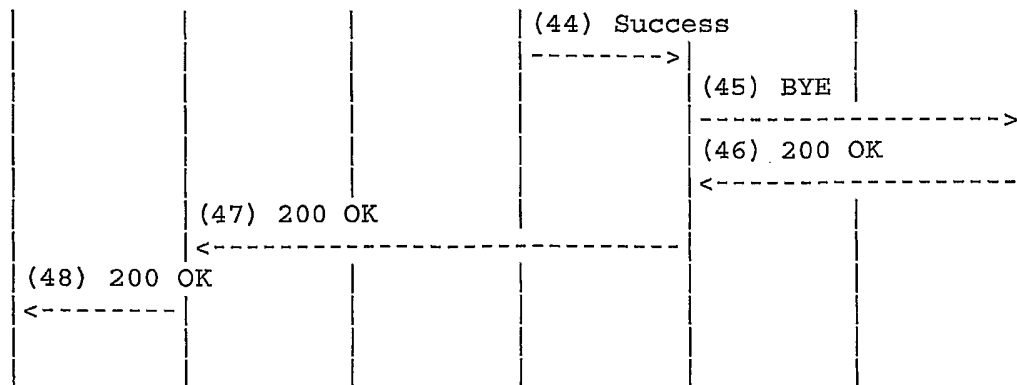
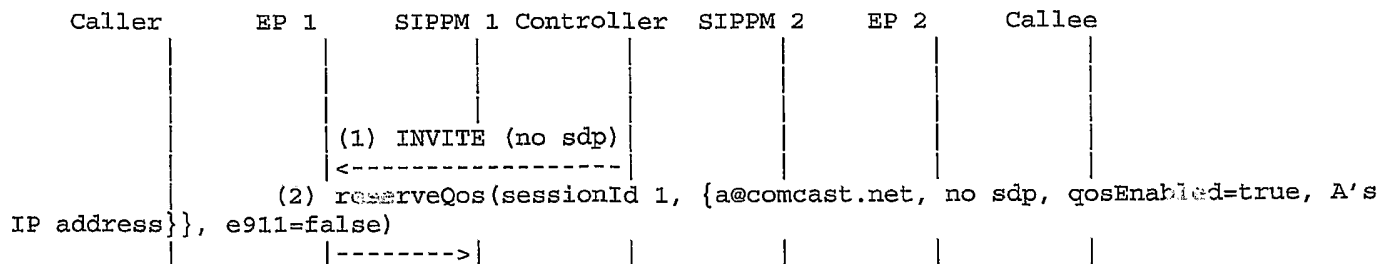
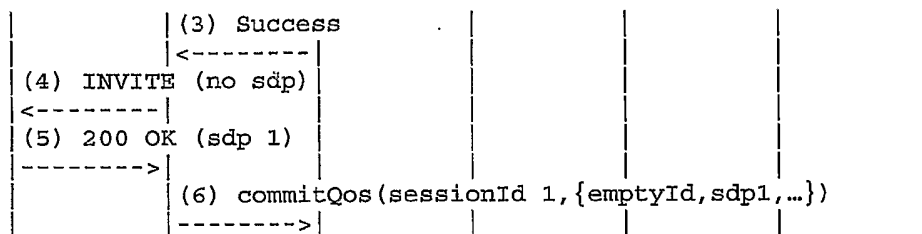


FIG. 11D

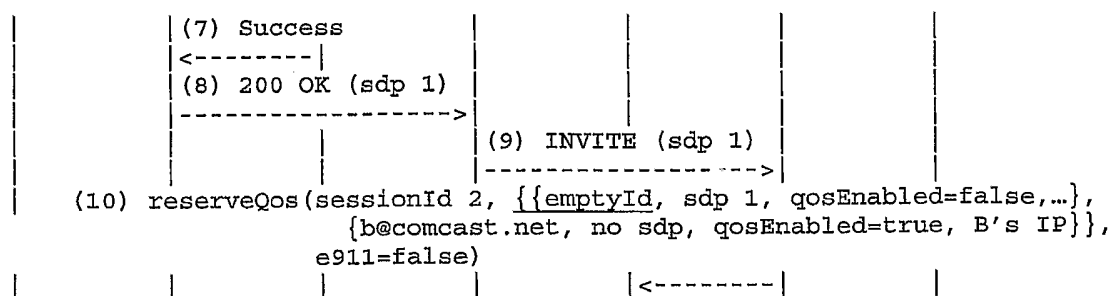


At this point, given that SIP PM 1 doesn't have any capabilities, it will save the information regarding A and wait for more info before reserving any resources.

NOTE: The EP needs to provide in this step A's id (as this is when it will look it up). It doesn't have to provide A's IP address given that there was no offer and no reservation is going to happen at this stage).



SIP PM 1 has now the offer from A. It will go ahead and reserve resources as it did in the previous sections. 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.

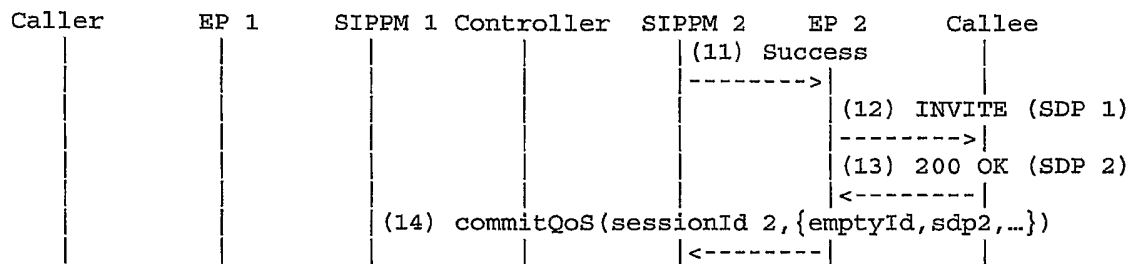


(Continued in FIG. 12B)

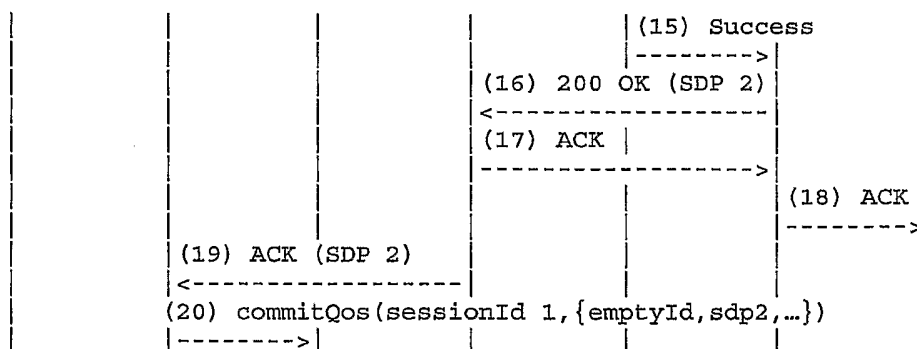
FIG. 12A

(Continued from FIG. 12A)

This will look like a normal call where the offer is in the INVITE, so the SIP PM will behave as it did in the basic call. SIP PM 2 will create gates for B in the "reserved" state by using sdp 1 and B's IP as the subscriber id.



SIP PM 2 will "commit" the gates and update the FlowSpecs/classifiers now that it has the answer as well.



Similarly, SIP PM 1 will modify the created gates and update the FlowSpecs/Classifiers now that it has the answer. If a media is rejected for instance, the corresponding gates will be deleted.

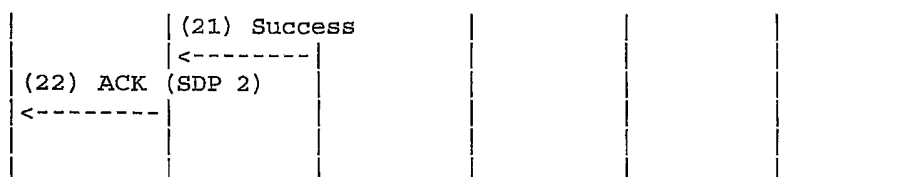
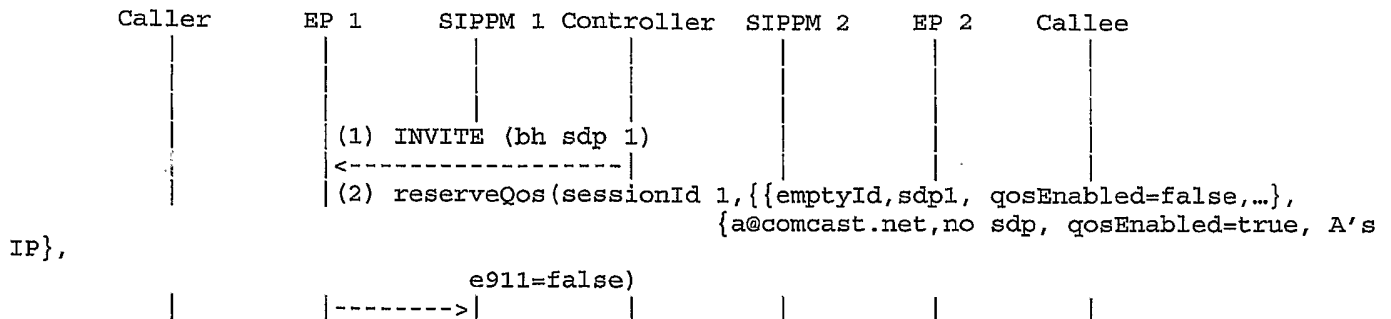
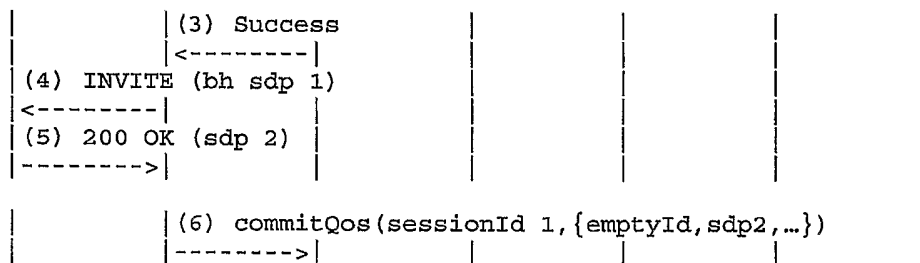


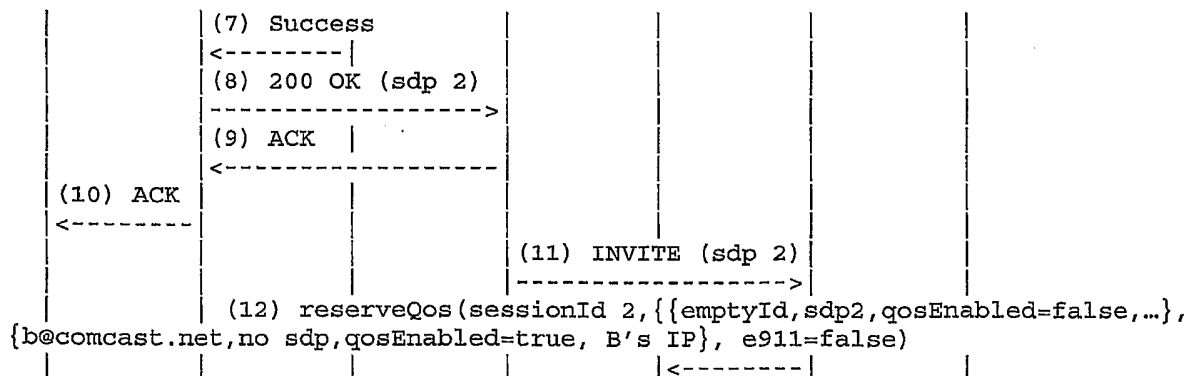
FIG. 12B



Given that sdp 1 contains a black-hole address, no gates will be reserved at this stage, however, information about the parties in the call will be saved.



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

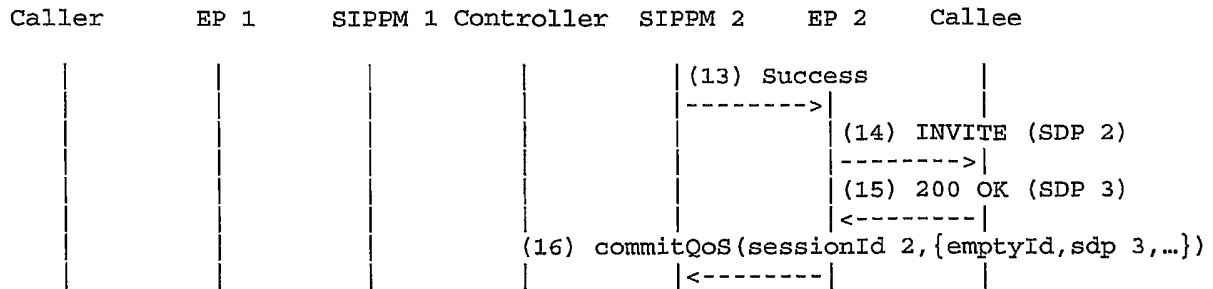


(Continued in FIG. 13B)

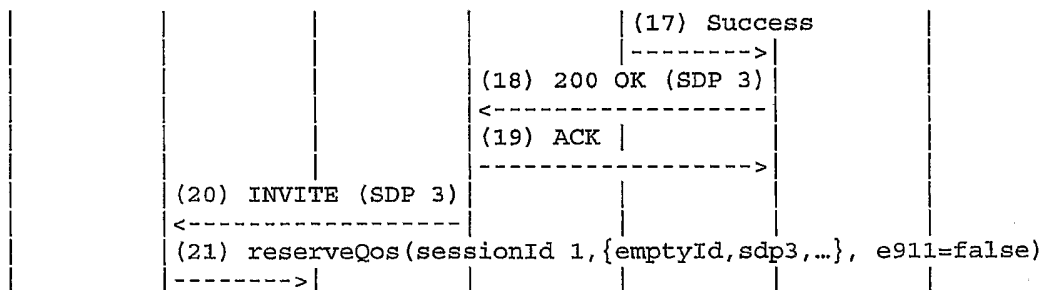
FIG. 13A

(Continued from FIG. 13A)

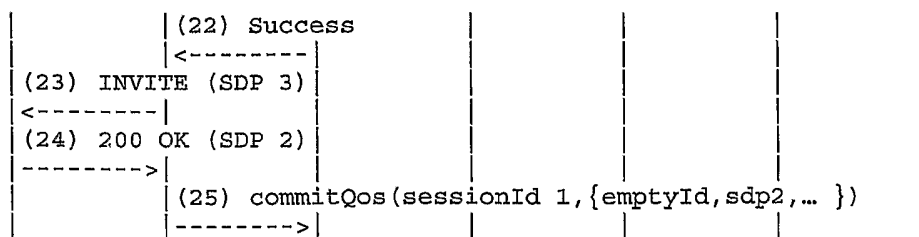
SIP PM 2 will behave as a basic call handling here. It will reserve resources based on sdp 2 and B's IP.



At this point, SIP PM 2 will commit all gates and given that it has all the info it needs, it will adjust the FlowSpecs/Classifiers as well as the state of the gates.



Given that this is a re-INVITE, no need to include A's info. SIP PM 1 will behave similarly to any re-invite as defined in the re-invite section.



At this point, If a change happened in the new offer/answer, the SIP PM will adjust the reservation according similarly to any re-invite case.

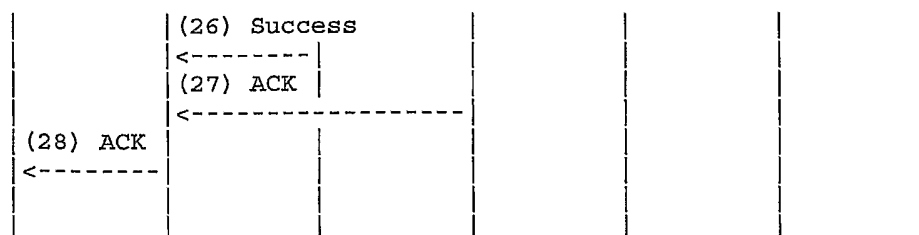
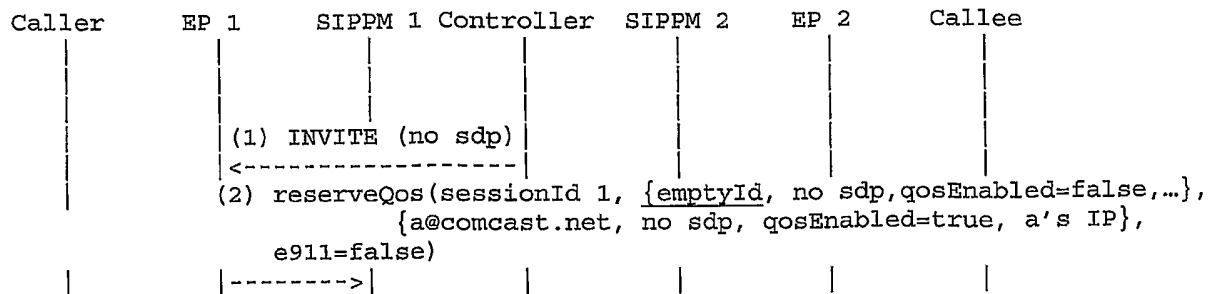
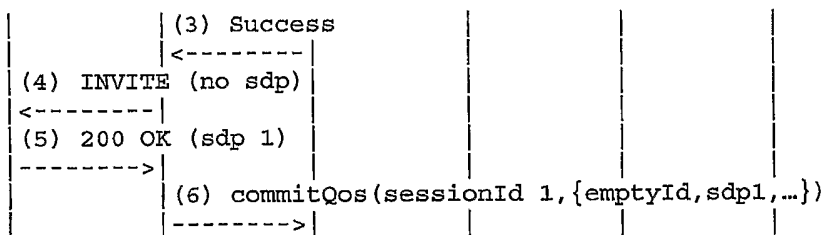


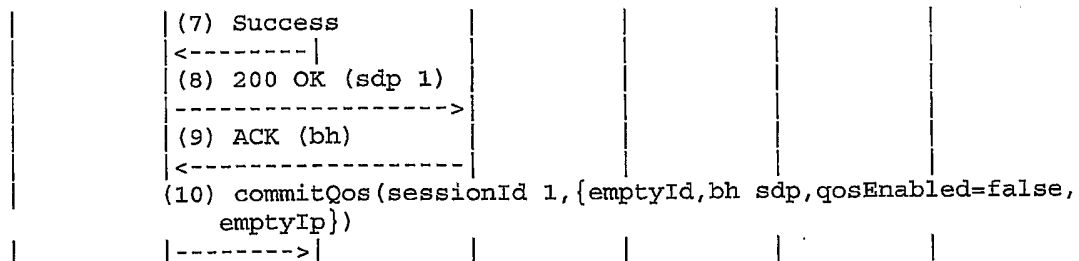
FIG. 13B



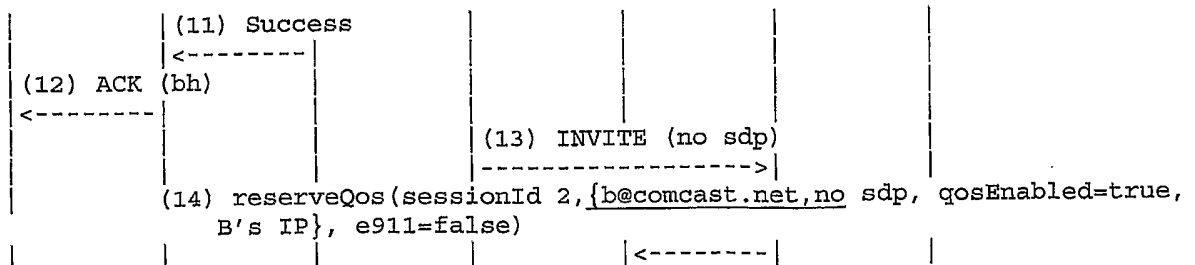
At this point, given that SIP PM 1 doesn't have any capabilities, it will save the info regarding both parties in the call and wait for more info before reserving any resources.



SIP PM 1 has now the offer from A. It will go ahead and reserve resources as it did in the previous sections. 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 the answer hasn't been received yet.



Now that the answer is received, the SIP PM will process it as any answer and move the downstream gate to the "committed" state and keep the upstream gate in the "reserved" state. This is because the SIP PM sees from the offer/answer that the answerer doesn't want to receive media (bh address), but cannot know if the answerer will not send media as the offer did contain a valid address.

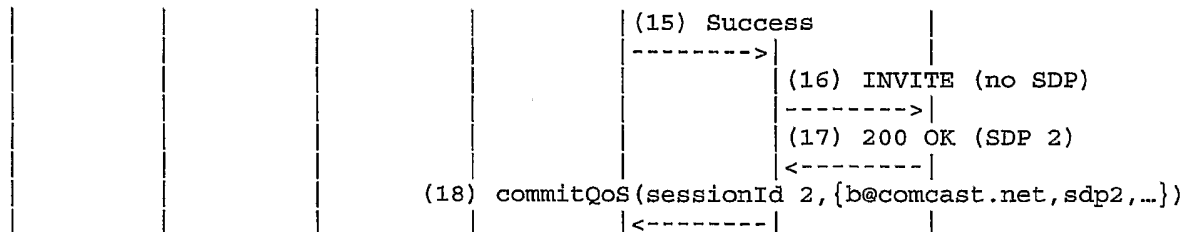


(Continued on FIG. 14B)

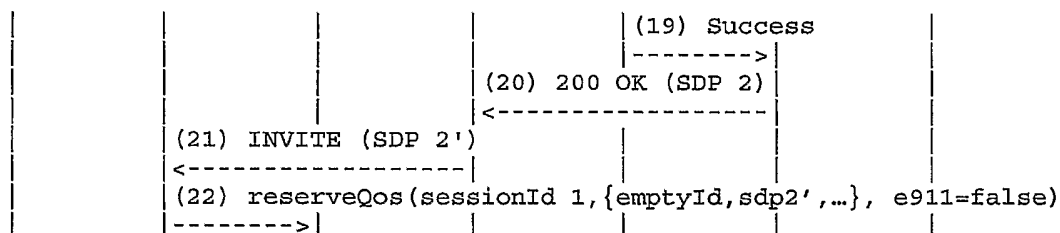
FIG. 14A

(Continued from FIG. 14A)

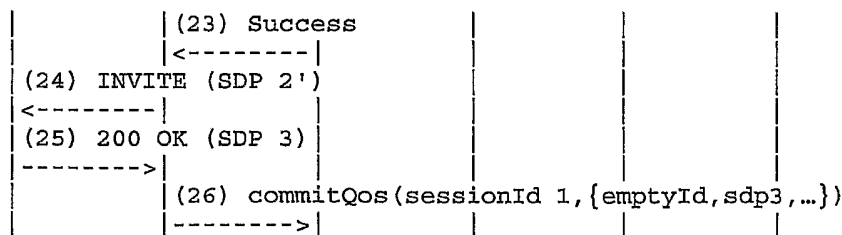
SIP PM 2 will just save the info and not reserve any resources at this stage, given that it doesn't have any capabilities.



SIP PM 2 has now the offer from B. It will go ahead and reserve resources as it did in the previous sections. It will assume that the answer will match the offer in order to compute the FlowSpec values. The gates will be in the "committed" state as the call was answered.



This will be treated as any re-invite (refer to re-invite section).

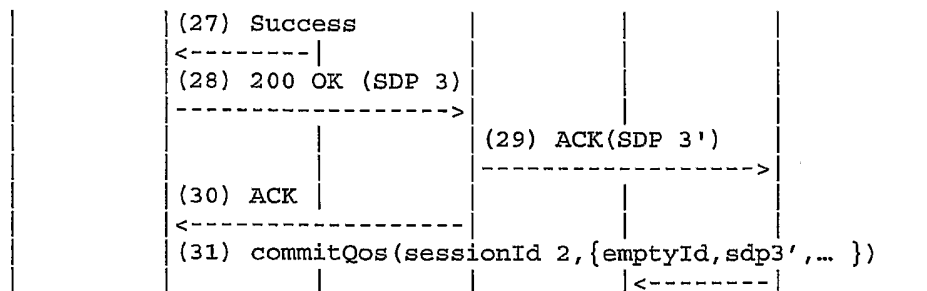


(Continued on FIG. 14C)

FIG. 14B

(Continued from FIG. 14B)

SIP PM 1 will modify adjust the gates based on the new offer/answer.



At this point, SIP PM 2 has both capabilities and will modify the gates appropriately by recomputing the FlowSpecs/classifiers based on the answer.

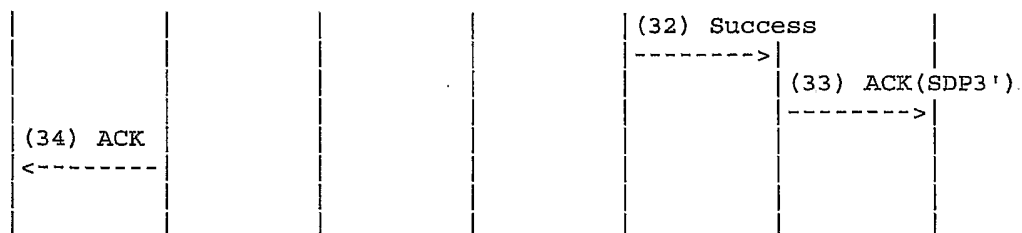


FIG. 14C