TransNexus

OSP Toolkit

Implementation Guide

Release 2.5.5

09 February 2002

OSP Toolkit

Implementation Guide

Release 2.5.5

09 February 2002

TransNexus
1140 Hammond Drive, Building E
Suite 5250
Atlanta, GA 30328
USA

Phone: +1 770 671 1888
Fax: +1 770 671 1188

E-mail: support@transnexus.com

Contents

Introduction

This document provides guidelines for the implementation of release 2.5.5 of the Open Settlement Protocol (OSP) Toolkit. That Toolkit, freely available under license from TransNexus, contains an implementation of the standard settlement protocol endorsed by the European Telecommunications Standards Institute (ETSI) and the International Multimedia Teleconferencing Consortium's Voice over IP (VoIP) Forum. The Toolkit also implements, as an option, extensions to the standard that allow access to enhanced services offered by some settlement service providers such as TransNexus.

The OSP Toolkit contains eleven separate documents, including this one. The documents are:

- *Introduction*
- *Implementation Guide*
- *How to Build and Test the OSP Toolkit*
- *Errorcode List*
- *Programming Interface*
- *Cisco Interoperability Example*
- *Device Enrollment*
- *Internal Architecture*
- *Porting Guide*
- *Protocol Extensions*
- *ETSI Technical Specification TS 101 321*

The *OSP Toolkit Introduction* includes a "Document Roadmap" section that summarizes the various documents and their application. This document contains two major sections. Those sections provide details on using the OSP Toolkit with various call control protocols, principally H.323 and SIP. The second section outlines Toolkit support for calling card and end user roaming services.

Notational Convention: This document discusses both the Open Settlement Protocol itself and the TransNexus OSP Software Development Kit. When referring to an element or component of the protocol, the document includes that element in XML angle brackets, `<AuthorisationRequest>` for example. When referring to a function call or parameter within the Toolkit, this document uses the full name of the function or variable, such as `OSPPTransactionRequestAuthorisation`. Note that Toolkit naming conventions use "`OSPP`" as the prefix for all global procedures and "`ospv`" as the prefix for all local variables.

Call Control Protocols

The Open Settlement Protocol is not limited to any particular call control protocol. Rather, it is neutral, and is designed for deployment with devices conforming to H.323 and Session Initiation Protocol (SIP), as well as various proprietary signaling protocols.

Note: Some well-known IP telephony protocols, including the Simple Gateway Control Protocol (SGCP), IP Device Control (IPDC), and Media Gateway Control Protocol (MGCP), do not integrate directly with OSP. Instead, systems relying on those

protocols integrate with OSP at the level of protocols between gateway controllers (e.g. call agents). Today, those protocols are primarily based on H.323 or SIP.

This section illustrates how OSP may be used with various call signaling protocols. Rather than considering each protocol separately, however, it considers three different architectures—peer-to-peer, distributed tightly-coupled, and distributed loosely-coupled—that can be applied to various signaling protocols. The H.323 and SIP protocols are used as examples to describe the interaction with OSP in detail. The principals described in these sections can also be used for other protocols.

Peer-to-Peer Architecture

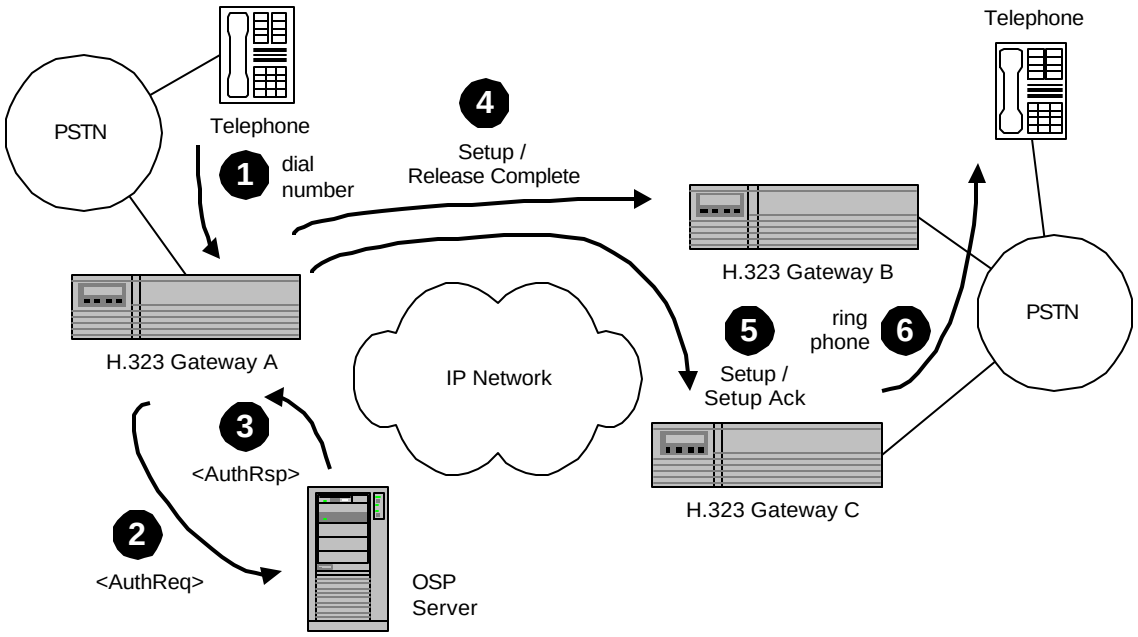
In a peer-to-peer architecture, gateways contact each other directly, without any control or coordination from other devices. This architecture corresponds to simple H.323 deployments without gatekeepers as well as basic SIP-based services. In such an environment, the endpoints of a call implement the open settlement protocol to find and authorize each other, and to directly report usage information to a settlement provider. The following subsections illustrate the use of OSP and the OSP Toolkit in both H.323 and SIP architectures.

H.323 Gateways

When operating with H.323 gateways, OSP provides services at both the start and end of each call. During initial call setup, gateways can use OSP to obtain call routing information and authorization tokens. Once the call has ended, gateways use OSP to report usage details.

Call Routing and Authorization

Figure 1 shows how the open settlement protocol may be used by H.323 gateways to find



• Figure 1 Call Routing and Authorization for Peer-to-Peer H.323 Gateways.

and authorize each other.

The figure highlights the following interactions between the devices.

1) Calling party indicates the desired, destination phone number to Gateway A (by, for example, responding to a DTMF-based IVR, sending an ISDN Q.931 Setup, or transmitting an SS7 Initial Address Message).

2) Gateway A sends an OSP `<AuthorisationRequest>` message to the OSP Server. The significant elements within the `<AuthorisationRequest>` include

`<Timestamp>`	Time of request
`<CallId>`	H.323 Call Identifier to be used for the call
`<SourceInfo type="e164">`	Calling party's E.164 number if available; otherwise a local E.164 number controlled by Gateway A, e.g. 14048724887; this number must be passed to the destination gateway(s) in Setup messages
`<SourceAlternate type="transport">`	DNS name or IP address of Gateway A, for example `gatewayA.carrier.com`
`<DestinationInfo type="e164">`	Called party's E.164 number, e.g. `33492944299`
`<Service/>`	Empty (for basic service)
`<MaximumDestinations>`	The maximum number of destinations, including alternatives, Gateway A will consider

If using the TransNexus OSP Toolkit, Gateway A can generate this message by calling `OSPPTransactionRequestAuthorisation()` with the following significant parameters.

`ospvSource`	DNS name or IP address of Gateway A, for example `"gatewayA.carrier.com"`
`ospvSourceDevice`	not needed in peer-to-peer environments; empty string (`""`) in this example
`ospvCallingNumber`	calling party's E.164 number if available; otherwise a local E.164 number controlled by Gateway A, e.g. `"14048724887"`; this number must be passed to the destination gateway during Setup
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvUser`	optional user identification; empty string (`""`) in this example
`ospvNumberOfCallIds`	the number of H.323 Call Identifiers given to the OSP server; if all Setup attempts for this call will use the same Call Identifier value (as in this example), this parameter value is `1`
`ospvCallIds`	an array (containing, in this example, but a single element) of H.323 Call Identifiers; the array structure includes a size field which indicates the size of the value
`ospvPreferredDestinations`	optional list of preferred destination gateways; empty string (`""`) in this example
`ospvNumberOfDestinations`	the maximum number of potential destinations that Gateway A is prepared to consider for the call; for example 3

3) OSP Server replies with an <AuthorisationResponse> message. The message indicates two candidate destinations: Gateway B and Gateway C, in that order. In particular, the <AuthorisationResponse> contains the following elements.

<Timestamp>	time of response
<Status>	result of response, e.g. <Code>200</Code>
<TransactionId>	transaction identifier assigned by settlement provider
<Destination>	first destination gateway to try for call
<DestinationSignalAddress type="transport">	DNS name or IP address of Gateway B, for example gatewayB.itsp.fr
<Token>	authorization token to be passed to Gateway B
<ValidAfter>	time after which token for Gateway B is valid
<ValidUntil>	time until which token for Gateway B is valid
<UsageDetail>	how much service is authorized with Gateway B
<Service/>	empty (for basic service)
<Amount>	amount of authorized service, e.g. 3600
<Increment>	increment of service measurement, e.g. 1
<Unit>	unit of service measurement, e.g. s for seconds
<CallId>	H.323 Call Identifier to be used for the call to Gateway B
<Destination>	second destination gateway to try for call
<DestinationSignalAddress type="transport">	DNS name or IP address of Gateway C, for example gatewayC.isp.fr
<Token>	authorization token to be passed to Gateway C
<ValidAfter>	time after which token for Gateway C is valid
<ValidUntil>	time until which token for Gateway C is valid
<UsageDetail>	how much service is authorized with Gateway C
<Service/>	empty (for basic service)
<Amount>	amount of authorized service, e.g. 3600
<Increment>	increment of service measurement, e.g. 1
<Unit>	unit of service measurement, e.g. s for seconds
<CallId>	H.323 Call Identifier to be used for the call to Gateway C

If using the TransNexus OSP Toolkit, Gateway A will be informed of the receipt of a valid OSP <AuthorisationResponse> by the return value of the call to the function OSPPTransactionRequestAuthorisation().

4) Gateway A sends an H.225.0 Setup message to Gateway B; however, the Setup is refused with a Release Complete. Note that the H.323 Call Identifier in the Setup message has the same value as in the original <AuthorisationRequest>. The Setup must also include the authorization token(s) provided in the OSP <AuthorisationResponse>. The OSP token, when carried as part of a call signalling message of an ASN.1 based protocol, may be identified by object identifiers defined in Annex D of ETSI TS 101 32 v2.1.1. For interoperability with Cisco gateways, use the osp_token_xml_format object identifier (see also the OSP Toolkit document *Cisco Interoperability Example*).

itu-t(0), identified-organization(4), etsi(0), ts-101-321(1321), token(1), xml-format(2)

If using the TransNexus OSP Toolkit, Gateway A can retrieve the information needed to access Gateway B by calling `OSPPTransactionGetFirstDestination()`. It will return the following information.

`ospvValidAfter`	time after which authorization token for Gateway B is valid
`ospvValidUntil`	time until which authorization token for Gateway B is valid
`ospvTimeLimit`	amount of service authorized with Gateway B, e.g. `3600` (seconds)
`ospvCallId`	H.323 Call Identifier to use in Setup message to Gateway B
`ospvCalledNumber`	number to present to Gateway B as the called party's number; often the same as passed to `OSPPTransactionRequestAuthorisation()`, e.g. `"33492944299"`, but not, for example, if the OSP server performs number translation
`ospvDestination`	DNS name or IP address of the destination gateway, for example `"gatewayB.itsp.fr"`
`ospvDestinationDevice`	not needed in peer-to-peer environments; empty string (`""`) in this example
`ospvToken`	authorization token to present to Gateway B during setup

5) Gateway A then sends a second H.225.0 Setup message, this time to Gateway C, and this time the Setup is accepted with a Setup Acknowledge. This Setup message also uses the H.323 Call Identifier from the `<AuthorisationRequest>`, and the Setup message must also include the authorization token(s) provided in the OSP `<AuthorisationResponse>`. The OSP token, when carried as part of a call signalling message of an ASN.1 based protocol, may be identified by object identifiers defined in Annex D of ETSI TS 101 32 v2.1.1. For interoperability with Cisco gateways, use the osp_token_xml_format object identifier (see also the OSP Toolkit document *Cisco Interoperability Example*).

itu-t(0), identified-organization(4), etsi(0), ts-101-321(1321), token(1), xml-format(2)

If using the TransNexus OSP Toolkit, Gateway A retrieves the information needed to access Gateway C with a call to `OSPPTransactionGetNextDestination()`. It returns the following information.

`ospvValidAfter`	time after which authorization token for Gateway C is valid
`ospvValidUntil`	time until which authorization token for Gateway C is valid
`ospvTimeLimit`	amount of service authorized with Gateway C, e.g. `3600` (seconds)
`ospvCallId`	H.323 Call Identifier to use in Setup message to Gateway C
`ospvCalledNumber`	number to present to Gateway C as the called party's number; often the same as passed to `OSPPTransactionRequestAuthorisation()`, e.g. `"33492944299"`, but not, for example, if the OSP server performs number translation
`ospvDestination`	DNS name or IP address of the destination gateway, for example `"gatewayC.isp.fr"`
`ospvDestinationDevice`	Not needed in peer-to-peer environments; empty string (`""`) in this example
`ospvToken`	authorization token to present to Gateway C during setup

6) Gateway C accepts the Setup and completes the call to the called party; the phone conversation can now take place.

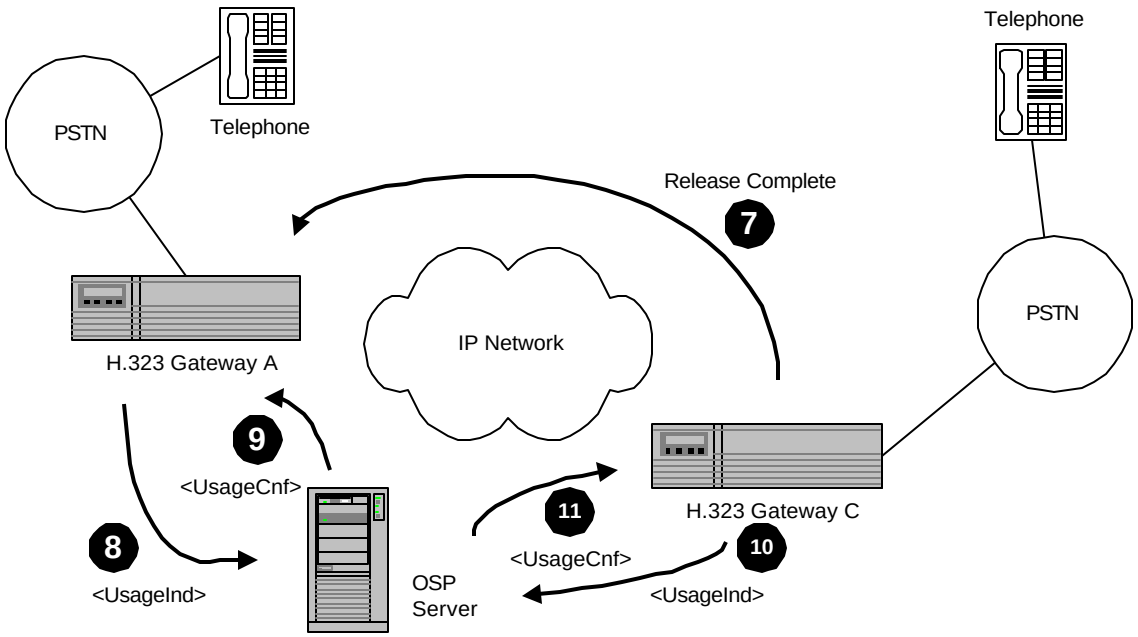
If using the TransNexus OSP Toolkit, Gateway C must call the Toolkit function `OSPPTransactionValidateAuthorisation()` to verify the authorization token in the Setup message. It would call it with the following parameters.

`ospvSource`	DNS name or IP address of gateway A, e.g. `"[172.16.1.1]"`
`ospvDestination`	DNS name or IP address of gateway C, e.g. `"gatewayC.isp.fr"`
`ospvSourceDevice`	not needed in peer-to-peer environments; empty string (`""`) in this example
`ospvDestinationDevice`	not needed in peer-to-peer environments; empty string (`""`) in this example
`ospvCallingNumber`	calling party's E.164 number, e.g. `"14048724887"`; this must be the same as was contained in the original `<AuthorisationRequest>`
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvCallId`	H.323 Call Identifier received in Setup message
`ospvToken`	authorization token presented to Gateway C during setup

The `OSPPTransactionValidateAuthorisation()` function will return an indication of whether or not the call is authorized (`ospvAuthorised`), and, if so, how much service is authorized (`ospvTimeLimit`).

Usage Reports

Once the call has ended, both gateways report usage details to an OSP server. As Figure 2 indicates, those reports are conveyed in OSP `<UsageIndication>` messages.



• Figure 2 Usage Reporting for Peer-to-Peer H.323 Gateways.

The steps shown in the figure are straightforward.

7) Gateways A and C clear the call by exchanging an H.225.0 Release Complete message.

8) Gateway A sends a <UsageIndication> message to the OSP server. If Gateway A is not using any protocol extensions, the message will contain the following elements.

<Timestamp>	time of request
<Role>	for Gateway A, source
<TransactionId>	transaction ID assigned by OSP server in authorization response
<CallId>	H.323 Call Identifier used for the call
<SourceInfo type="e164">	calling party's E.164 number as returned in the authorization response, e.g. 14048724887
<SourceAlternate type="transport">	DNS name or IP address of Gateway A, for example gatewayA.carrier.com
<DestinationInfo type="e164">	called party's E.164 number, e.g. 33492944299
<DestinationAlternate type="transport">	DNS name or IP address of Gateway C, for example, gatewayC.isp.fr
<UsageDetail>	usage information for the call
<Service/>	empty (for basic service)
<Amount>	amount of service used, e.g. 300
<Increment>	increment of service measurement, e.g. 1
<Unit>	unit of service measurement, e.g. s for seconds

If using the TransNexus OSP Toolkit, Gateway A can generate that message by calling OSPPTransactionReportUsage() with the following significant parameters

ospvDuration	length of the call in seconds, e.g. 300
ospvLossPacketsSent	value ignored (because of following parameter value)
ospvLossFractionSent	-1 if not reported
ospvLossPacketsReceived	value ignored (because of following parameter value)
ospvLossFractionReceived	-1 if not reported

9) The OSP server responds with a <UsageConfirmation> message. If it has accepted the usage report, that message will contain a successful <Status> element (e.g. <Code>200</Code>. Gateway A will learn of this response in the return value from the OSPPTransactionReportUsage() function.

10) Gateway C also sends a <UsageIndication> to the OSP server. Assuming that it uses TransNexus extensions for statistics reporting, that message would include the following elements.

<Timestamp>	time of request

`<Role>`	for Gateway C, `destination`
`<TransactionId>`	transaction ID assigned by OSP server and passed to Gateway C in authorization token
`<CallId>`	H.323 Call Identifier used for the call
`<SourceInfo type="e164">`	calling party's E.164 number as presented in the Setup message, e.g. `14048724887`
`<SourceAlternate type="transport">`	DNS name or IP address of Gateway A, for example `[172.16.1.1]`
`<DestinationInfo type="e164">`	called party's E.164 number, e.g. `33492944299`
`<DestinationAlternate type="transport">`	DNS name or IP address of Gateway C, for example, `gatewayC.isp.fr`
`<UsageDetail>`	usage information for the call
`<Service/>`	empty (for basic service)
`<Amount>`	amount of service used, e.g. `300`
`<Increment>`	increment of service measurement, e.g. `1`
`<Unit>`	unit of service measurement, e.g. `s` for seconds
`<transnexus.com:Statistics>`	statistical information for call
`<transnexus.com:LossSent>`	loss information for packets sent by Gateway C
`<transnexus.com:Packets>`	number of packets lost from Gateway C to Gateway A
`<transnexus.com:Fraction>`	fraction (from 0 to 255) of packets lost from C to A
`<transnexus.com:LossReceived>`	loss information for packets sent by Gateway A
`<transnexus.com:Packets>`	number of packets lost from Gateway A to Gateway C
`<transnexus.com:Fraction>`	fraction (from 0 to 255) of packets lost from A to C
`<transnexus.com:OneWayDelay>`	one way delay measured from Gateway A to C
`<transnexus.com:Minimum>`	minimum measured value for delay, in seconds
`<transnexus.com:Mean>`	sample mean of delay measurements, in seconds
`<transnexus.com:Variance>`	sample variance of delay measurements, in squared seconds
`<transnexus.com:Samples>`	number of sample measurements
`<transnexus.com:RoundTripDelay>`	round trip delay between Gateway A and C measured during call
`<transnexus.com:Minimum>`	minimum measured value for delay, in seconds
`<transnexus.com:Mean>`	sample mean of delay measurements, in seconds
`<transnexus.com:Variance>`	sample variance of delay measurements, in squared seconds
`<transnexus.com:Samples>`	number of sample measurements

If using the TransNexus OSP Toolkit, Gateway C may call two Toolkit function while the call is in progress: `OSPPTransactionAccumulateOneWayDelay()` for one-way delay measurements, and `OSPPTransactionAccumulateRoundTripDelay()` for round-trip delay measurements. Note that these functions must be called before `OSPPTransactionReportUsage()`. That call is used to indicate loss statistics as well as the call's duration.

11) The OSP server responds with a `<UsageConfirmation>` message. If it has accepted the usage report, that message will contain a successful `<Status>` element (e.g. `<Code>200</Code>`. Gateway C will learn of this response in the return value from the `OSPPTransactionReportUsage()` function.

Session Initiation Protocol Gateways

In a simple peer-to-peer environment, Session Initiation Protocol (SIP) gateways operate in a manner nearly identical to H.323 gateways. As before, it is convenient to consider interaction using OSP before and after the multimedia call.

Call Routing and Authorization

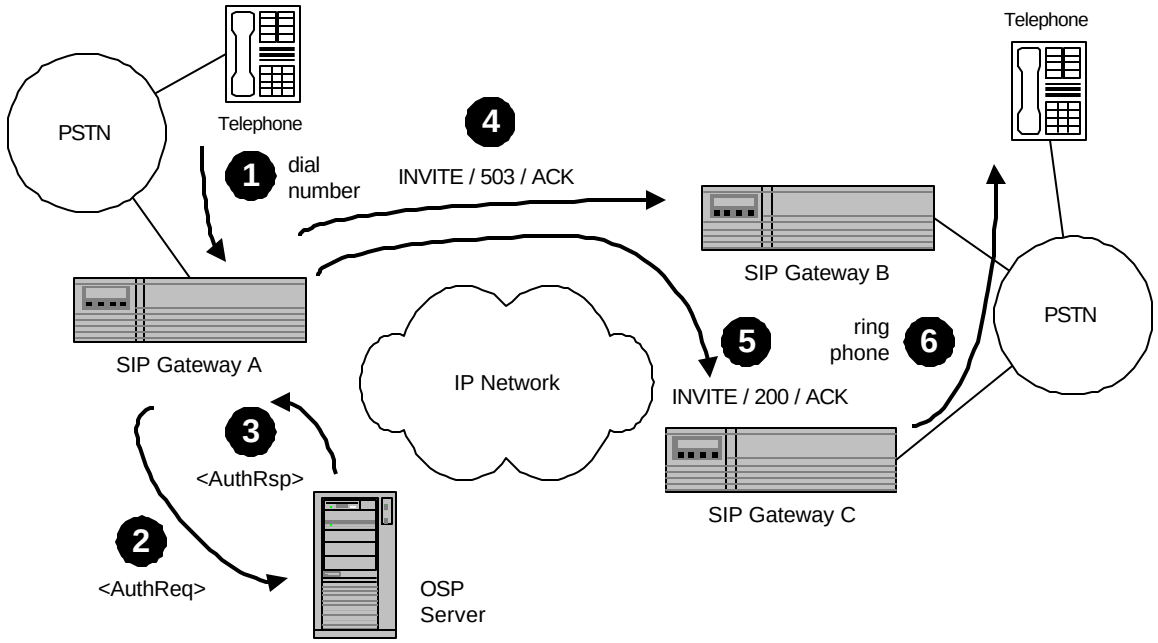
Figure 3 shows how the open settlement protocol may be used by SIP gateways to find and authorize each other. (Note: this function allows a SIP gateway to find a peer gateway for a particular, PSTN-terminated, telephone user. As such, it is not the same as the standard SIP user location function.)

The figure highlights the following interactions between the devices.

1) Calling party indicates the desired, destination phone number to Gateway A (by, for example, responding to a DTMF-based IVR, sending an ISDN Q.931 Setup, or transmitting an SS7 Initial Address Message).

2) Gateway A sends an OSP `<AuthorisationRequest>` message to the OSP Server. The significant elements within the `<AuthorisationRequest>` include

`<Timestamp>`	time of request



- Figure 3 Call Routing and Authorization for Peer-to-Peer SIP Gateways.

<CallId>	SIP Call-ID to be used for the call
<SourceInfo type="e164">	calling party's E.164 number if available; otherwise a local E.164 number controlled by Gateway A, e.g. 14048724887; this number must be passed to the destination gatew ay(s) in the subsequent INVITE method
<SourceAlternate type="transport">	DNS name or IP address of Gateway A, for example gatewayA.carrier.com
<DestinationInfo type="e164">	called party's E.164 number, e.g. 33492944299
<Service/>	empty (for basic service)
<MaximumDestinations>	the maximum number of destinations, including alternatives, Gateway A will consider

If using the TransNexus OSP Toolkit, Gateway A can generate this message by calling OSPPTransactionRequestAuthorisation() with the following significant parameters.

ospvSource	DNS name or IP address of Gateway A, for example "gatewayA.carrier.com"
ospvSourceDevice	not needed in peer-to-peer environments; empty string ("") in this example
ospvCallingNumber	calling party's E.164 number if available; otherwise a local E.164 number controlled by Gateway A, e.g. "14048724887"; this number must be passed to the destination gateway in the INVITE method
ospvCalledNumber	called party's E.164 number, e.g. "33492944299"
ospvUser	optional user identification; empty string ("") in this example
ospvNumberOfCallIds	the number of SIP Call-IDs given to the OSP server; if all INVITE attempts for this call will use the same Call-ID value (as in this example), this parameter value is 1
ospvCallIds	an array (containing, in this example, but a single element) of SIP Call-IDs; the array structure includes a size field which indicates the size of the value
ospvPreferredDestinations	optional list of preferred destination gateways; empty string ("") in this example
ospvNumberOfDestinations	the maximum number of potential destinations that Gateway A is prepared to consider for the call; for example 3

3) OSP Server replies with an <AuthorisationResponse> message. The message indicates two candidate destinations: Gateway B and Gateway C, in that order. In particular, the <AuthorisationResponse> contains the following elements.

<Timestamp>	time of response
<Status>	result of response, e.g. <Code>200</Code>
<TransactionId>	transaction identifier assigned by settlement provider
<Destination>	first destination gateway to try for call
<DestinationSignalAddress type="transport">	DNS name or IP address of Gateway B, for example gatewayB.itsp.fr
<Token>	authorization token to be passed to Gateway B
<ValidAfter>	time after which token for Gateway B is valid

<ValidUntil>	time until which token for Gateway B is valid
<UsageDetail>	how much service is authorized with Gateway B
<Service/>	empty (for basic service)
<Amount>	amount of authorized service, e.g. 3600
<Increment>	increment of service measurement, e.g. 1
<Unit>	unit of service measurement, e.g. s for seconds
<CallId>	SIP Call-ID to be used for the call to Gateway B
<Destination>	second destination gateway to try for call
<DestinationSignalAddress type="transport">	DNS name or IP address of Gateway C, for example gatewayC.isp.fr
<Token>	authorization token to be passed to Gateway C
<ValidAfter>	time after which token for Gateway C is valid
<ValidUntil>	time until which token for Gateway C is valid
<UsageDetail>	how much service is authorized with Gateway C
<Service/>	empty (for basic service)
<Amount>	amount of authorized service, e.g. 3600
<Increment>	increment of service measurement, e.g. 1
<Unit>	unit of service measurement, e.g. s for seconds
<CallId>	SIP Call-ID to be used for the call to Gateway C

If using the TransNexus OSP Toolkit, Gateway A will be informed of the receipt of a valid OSP <AuthorisationResponse> by the return value of the call to the function OSPPTransactionRequestAuthorisation().

4) Gateway A sends an INVITE message to Gateway B; however, Gateway B cannot accept the request. It responds with a "503 Service Unavailable," which Gateway A acknowledges with an ACK. Note that the SIP Call-ID in the INVITE message has the same value as in the original <AuthorisationRequest>. The INVITE must also include authorization token(s) provided in the OSP <AuthorisationResponse>. Each token should be conveyed in the SIP message body using the application/osp-token MIME type, as defined in http://www.ietf.org/internet-drafts/draft-johnston-sip-osp-token-02.txt. (The OSP Toolkit document "Cisco Interoperability Example" provides additional information on OSP token formatting.)

   If using the TransNexus OSP Toolkit, Gateway A can retrieve the information needed to access Gateway B by calling OSPPTransactionGetFirstDestination(). It will return the following information.

ospvValidAfter	time after which authorization token for Gateway B is valid
ospvValidUntil	time until which authorization token for Gateway B is valid
ospvTimeLimit	amount of service authorized with Gateway B, e.g. 3600 (seconds)
ospvCallId	SIP Call-ID to use in INVITE message to Gateway B
ospvCalledNumber	number to present to Gateway B as the called party's number; often the same as passed to OSPPTransactionRequestAuthorisation(), e.g. "33492944299", but not, for example, if the OSP server performs number translation

`ospvDestination`	DNS name or IP address of the destination gateway, for example `"gatewayB.itsp.fr"`
`ospvDestinationDevice`	not needed in peer-to-peer environments; empty string (`""`) in this example
`ospvToken`	authorization token to present to Gateway B during setup

5) Gateway A then sends a second INVITE message, this time to Gateway C, and this time the INVITE is accepted with "200 Success," to which Gateway A replies with an ACK. This INVITE message also uses the SIP Call-ID from the `<AuthorisationRequest>`, and the INVITE must also include the authorization token(s) provided in the OSP `<AuthorisationResponse>`. Each token should be conveyed in the SIP message body using the application/osp-token MIME type, as defined in http://www.ietf.org/internet-drafts/draft-johnston-sip-osp-token-02.txt.

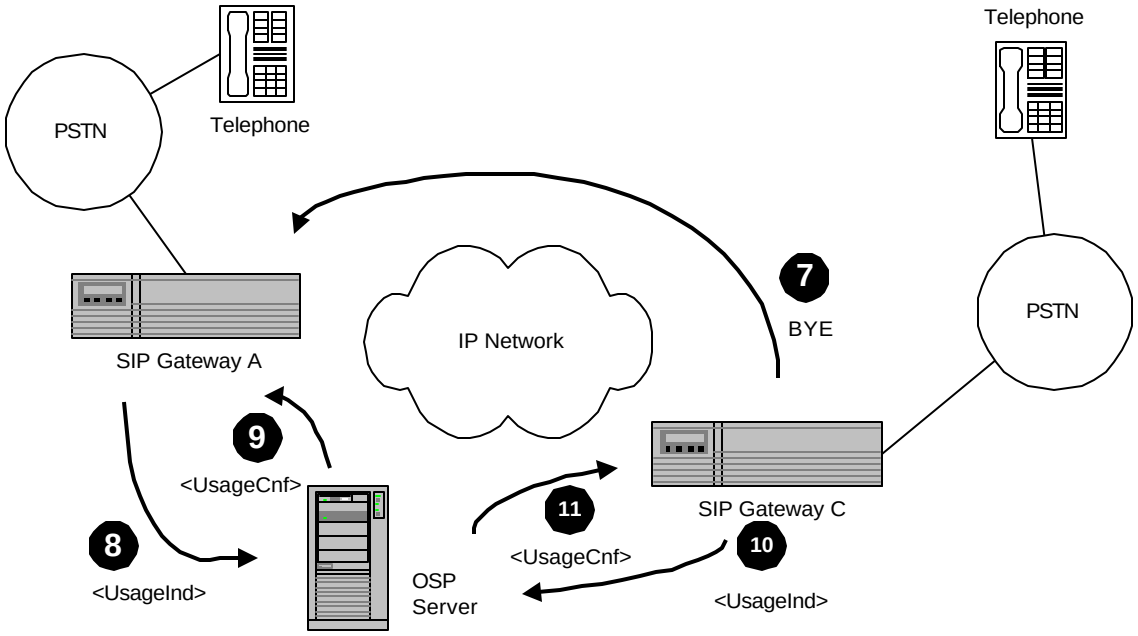
   If using the TransNexus OSP Toolkit, Gateway A retrieves the information needed to access Gateway C with a call to `OSPPTransactionGetNextDestination()`. It returns the following information.

`ospvValidAfter`	time after which authorization token for Gateway C is valid
`ospvValidUntil`	time until which authorization token for Gateway C is valid
`ospvTimeLimit`	amount of service authorized with Gateway C, e.g. `3600` (seconds)
`ospvCallId`	H.323 Call Identifier to use in Setup message to Gateway C
`ospvCalledNumber`	number to present to Gateway C as the called party's number; often the same as passed to `OSPPTransactionRequestAuthorisation()`, e.g. `"33492944299"`, but not, for example, if the OSP server performs number translation
`ospvDestination`	DNS name or IP address of the destination gateway, for example `"gatewayC.isp.fr"`
`ospvDestinationDevice`	not needed in peer-to-peer environments; empty string (`""`) in this example
`ospvToken`	authorization token to present to Gateway C during setup

6) Gateway C accepts the INVITE and completes the call to the called party; the phone conversation can now take place.

   If using the TransNexus OSP Toolkit, Gateway C must call the Toolkit function `OSPPTransactionValidateAuthorisation()` to verify the authorization token in the INVITE message. It would call it with the following parameters.

`ospvSource`	DNS name or IP address of gateway A, e.g. `"[172.16.1.1]"`
`ospvDestination`	DNS name or IP address of gateway C, e.g. `"gatewayC.isp.fr"`
`ospvSourceDevice`	not needed in peer-to-peer environments; empty string (`""`) in this example
`ospvDestinationDevice`	not needed in peer-to-peer environments; empty string (`""`) in this example
`ospvCallingNumber`	calling party's E.164 number, e.g. `"14048724887"`; this must be the same as was contained in the original `<AuthorisationRequest>`
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`



• Figure 4 Usage Reporting for Peer-to-Peer SIP Gateways.

`ospvCallId`	SIP Call-ID received in Setup message
`ospvToken`	authorization token presented to Gateway C during setup

The `OSPPTransactionValidateAuthorisation()` function will return an indication of whether or not the call is authorized (`ospvAuthorised`), and, if so, how much service is authorized (`ospvTimeLimit`).

Usage Reports

Once the call has ended, both gateways report usage details to an OSP server. As Figure indicates, those reports are conveyed in OSP `<UsageIndication>` messages.

The steps shown in figure 4 are straightforward.

7) Gateways A and C clear the call with a SIP BYE message.

8) Gateway A sends a `<UsageIndication>` message to the OSP server. If Gateway A is not using any protocol extensions, the message will contain the following elements.

`<Timestamp>`	time of request
`<Role>`	for Gateway A, `source`
`<TransactionId>`	transaction ID assigned by OSP server in authorization response
`<CallId>`	SIP Call-ID used for the call
`<SourceInfo type="e164">`	calling party's E.164 number as returned in the authorization

	response, e.g. `14048724887`
`<SourceAlternate type="transport">`	DNS name or IP address of Gateway A, for example `gatewayA.carrier.com`
`<DestinationInfo type="e164">`	called party's E.164 number, e.g. `33492944299`
`<DestinationAlternate type="transport">`	DNS name or IP address of Gateway C, for example, `gatewayC.isp.fr`
`<UsageDetail>`	usage information for the call
`<Service/>`	empty (for basic service)
`<Amount>`	amount of service used, e.g. `300`
`<Increment>`	increment of service measurement, e.g. `1`
`<Unit>`	unit of service measurement, e.g. `s` for seconds

If using the TransNexus OSP Toolkit, Gateway A can generate that message by calling `OSPPTransactionReportUsage()` with the following significant parameters

`ospvDuration`	length of the call in seconds, e.g. `300`
`ospvLossPacketsSent`	ignored (because of following parameter value)
`ospvLossFractionSent`	`-1` if not reported
`ospvLossPacketsReceived`	ignored (because of following parameter value)
`ospvLossFractionReceived`	`-1` if not reported

9) The OSP server responds with a `<UsageConfirmation>` message. If it has accepted the usage report, that message will contain a successful `<Status>` element (e.g. `<Code>200</Code>`). Gateway A will learn of this response in the return value from the `OSPPTransactionReportUsage()` function.

10) Gateway C also sends a `<UsageIndication>` to the OSP server. Assuming that it uses TransNexus extensions for statistics reporting, that message would include the following elements.

`<Timestamp>`	time of request
`<Role>`	for Gateway C, `destination`
`<TransactionId>`	transaction ID assigned by OSP server and passed to Gateway C in authorization token
`<CallId>`	SIP Call-ID used for the call
`<SourceInfo type="e164">`	calling party's E.164 number as presented in the INVITE message, e.g. `14048724887`
`<SourceAlternate type="transport">`	DNS name or IP address of Gateway A, for example `[172.16.1.1]`
`<DestinationInfo type="e164">`	called party's E.164 number, e.g. `33492944299`
`<DestinationAlternate type="transport">`	DNS name or IP address of Gateway C, for example, `gatewayC.isp.fr`
`<UsageDetail>`	usage information for the call
`<Service/>`	empty (for basic service)
`<Amount>`	amount of service used, e.g. `300`

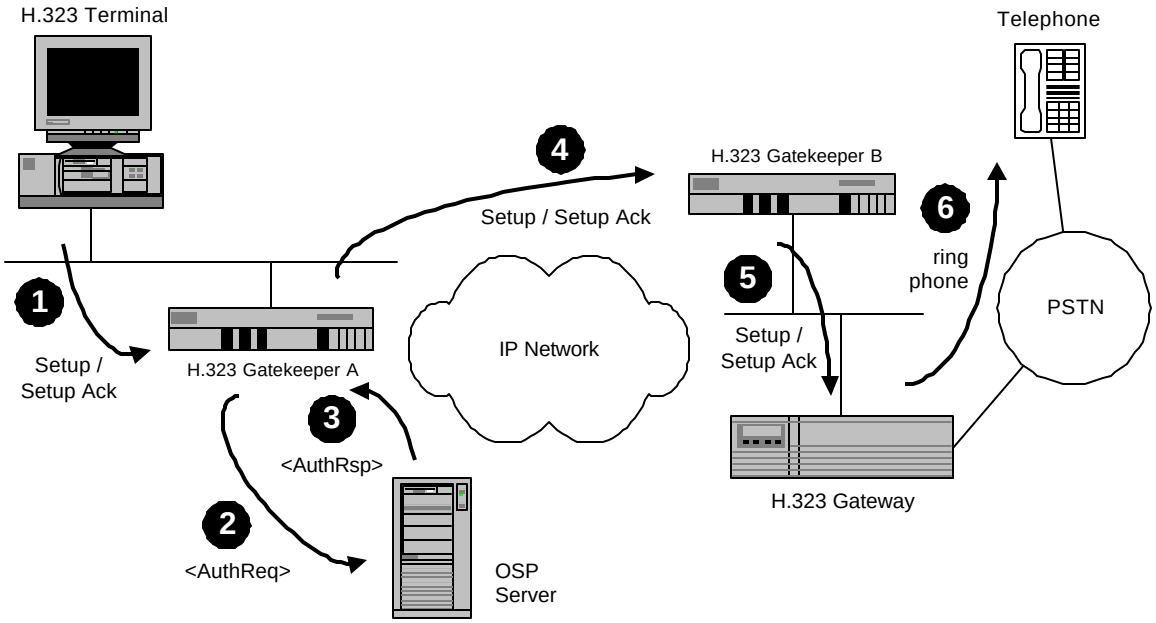
<Increment>	increment of service measurement, e.g. 1
<Unit>	unit of service measurement, e.g. s for seconds
<transnexus.com:Statistics>	statistical information for call
<transnexus.com:LossSent>	loss information for packets sent by Gateway C
<transnexus.com:Packets>	number of packets lost from Gateway C to Gateway A
<transnexus.com:Fraction>	fraction (from 0 to 255) of packets lost from C to A
<transnexus.com:LossReceived>	loss information for packets sent by Gateway A
<transnexus.com:Packets>	number of packets lost from Gateway A to Gateway C
<transnexus.com:Fraction>	fraction (from 0 to 255) of packets lost from A to C
<transnexus.com:OneWayDelay>	one way delay measured from Gateway A to C
<transnexus.com:Minimum>	minimum measured value for delay, in seconds
<transnexus.com:Mean>	sample mean of delay measurements, in seconds
<transnexus.com:Variance>	sample variance of delay measurements, in squared seconds
<transnexus.com:Samples>	number of sample measurements
<transnexus.com:RoundTripDelay>	round trip delay between Gateway A and C measured during call
<transnexus.com:Minimum>	minimum measured value for delay, in seconds
<transnexus.com:Mean>	sample mean of delay measurements, in seconds
<transnexus.com:Variance>	sample variance of delay measurements, in squared seconds
<transnexus.com:Samples>	number of sample measurements

If using the TransNexus OSP Toolkit, Gateway C may call two Toolkit function while the call is in progress: `OSPPTransactionAccumulateOneWayDelay()` for one-way delay measurements, and `OSPPTransactionAccumulateRoundTripDelay()` for round-trip delay measurements. Note that these functions must be called before `OSPPTransactionReportUsage()`. That call is used to indicate loss statistics as well as the call's duration.

11) The OSP server responds with a `<UsageConfirmation>` message. If it has accepted the usage report, that message will contain a successful `<Status>` element (e.g. `<Code>200</Code>`. Gateway C will learn of this response in the return value from the `OSPPTransactionReportUsage()` function.

Tightly Controlled Distributed Architecture

The Open Settlement Protocol supports operation in a tightly controlled, distributed architecture. That architecture includes some H.323-based deployments with active gatekeepers and SIP proxy servers (but not SIP redirect servers). The following subsections illustrate the use of OSP, and the TransNexus OSP Toolkit, in those environments.



- Figure 5 Call Routing and Authorization with H.323 Gatekeeper Routed Calls.

H.323 Gatekeeper Routed Calls

In H.323 deployments with gatekeepers, the gatekeeper may play a very active roll in call signaling. In such an environment, gatekeepers not only assume responsibility for call routing and authorization on behalf of their endpoints. They also act as the signaling endpoint for calls into their zone. As Figure 5 shows, gatekeepers may support multimedia terminals as well as gateways. As that figure and the following figure also highlight, only gatekeepers are required to support the Open Settlement Protocol in this environment.

Note that all destination gateways (or endpoints) controlled by the gatekeepers must be equivalent, as far as the OSP server is concerned. Indeed, the OSP server is not even aware of the existence of any H.323 terminals or gateways. Such an architecture, therefore, places implicit restrictions on the services offered by the operators. An operator, for example, will be unable to price termination services differently for different gateways in the same zone.

Call Routing and Authorization

1) H.323 Terminal begins a call by sending an H.225.0 Setup message its gatekeeper, Gatekeeper A. (Note: This scenario assumes the use of pre-granted admission requests.) The Setup indicates that the called party is identified by an E.164 phone number such as +33 4 92 94 42 99.

2) Gatekeeper A sends an OSP `<AuthorisationRequest>` message to the OSP Server. The significant elements within the `<AuthorisationRequest>` include

`<Timestamp>`	time of request

`<CallId>`	H.323 Call Identifier to be used for the call
`<SourceInfo type="e164">`	a representation of the H.323 Terminal using an E.164 number; in the absence of other information, this number may be derived using the IP address to E.164 number mapping of ETSI TIPHON; this number must be passed to the destination gateway(s) in Setup messages
`<SourceAlternate type="transport">`	DNS name or IP address of Gatekeeper A, for example `gatekeeperA.carrier.com`
`<DestinationInfo type="e164">`	called party's E.164 number, e.g. `33492944299`
`<Service/>`	empty (for basic service)
`<MaximumDestinations>`	the maximum number of destinations, including alternatives, Gatekeeper A will consider

If using the TransNexus OSP Toolkit, Gatekeeper A can generate this message by calling the `OSPPTransactionRequestAuthorisation()` function with the following significant parameters.

`ospvSource`	DNS name or IP address of Gatekeeper A, for example `"gatekeeperA.carrier.com"`
`ospvSourceDevice`	not needed in tightly-coupled environments in which the endpoint (e.g. the H.323 terminal) is not known to the OSP server; empty string (`""`) in this example
`ospvCallingNumber`	an E.164-format representation of the H.323 Terminal, possibly derived using the IP address to E.164 number mapping of ETSI TIPHON; this number must be passed to the destination gateway during Setup
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvUser`	optional user identification; empty string (`""`) in this example
`ospvNumberOfCallIds`	the number of H.323 Call Identifiers given to the OSP server; if all Setup attempts for this call will use the same Call Identifier value (as in this example), this parameter value is `1`
`ospvCallIds`	an array (containing, in this example, but a single element) of H.323 Call Identifiers; the array structure includes a size field which indicates the size of the value
`ospvPreferredDestinations`	optional list of preferred destination gatekeepers; empty string (`""`) in this example
`ospvNumberOfDestinations`	the maximum number of potential destinations that Gatekeeper A is prepared to consider for the call; for example 3

3) OSP Server replies with an `<AuthorisationResponse>` message. The message identifies Gatekeeper B. In particular, the `<AuthorisationResponse>` contains the following elements.

`<Timestamp>`	time of response
`<Status>`	result of response, e.g. `<Code>200</Code>`
`<TransactionId>`	transaction identifier assigned by settlement provider
`<Destination>`	first destination gateway to try for call
`<DestinationSignalAddress type="transport">`	DNS name or IP address of Gatekeeper B, for example `gatekeeperB.itsp.fr`
`<Token>`	authorization token to be passed to Gatekeeper B

<ValidAfter>	time after which token for Gatekeeper B is valid
<ValidUntil>	time until which token for Gatekeeper B is valid
<UsageDetail>	how much service is authorized with Gatekeeper B
<Service/>	empty (for basic service)
<Amount>	amount of authorized service, e.g. 3600
<Increment>	increment of service measurement, e.g. 1
<Unit>	unit of service measurement, e.g. s for seconds
<CallId>	H.323 Call Identifier to be used for the call to Gatekeeper B

If using the TransNexus OSP Toolkit, Gatekeeper A will be informed of the receipt of a valid OSP <AuthorisationResponse> by the return value of the call to the function OSPPTransactionRequestAuthorisation().

4) Gatekeeper A extends the call with its own Setup message to Gatekeeper B.

If using the TransNexus OSP Toolkit, Gatekeeper A can retrieve the information needed to access Gatekeeper B by calling OSPPTransactionGetFirstDestination(). It will return the following information.

ospvValidAfter	time after which authorization token for Gatekeeper B is valid
ospvValidUntil	time until which authorization token for Gatekeeper B is valid
ospvTimeLimit	amount of service authorized with Gatekeeper B, e.g. 3600 (seconds)
ospvCallId	H.323 Call Identifier to use in Setup message to Gatekeeper B
ospvCalledNumber	number to present to Gatekeeper B as the called party's number; often the same as passed to OSPPTransactionRequestAuthorisation(), e.g. "33492944299", but not, for example, if the OSP server performs number translation
ospvDestination	DNS name or IP address of the destination system, for example "gatekeeperB.itsp.fr"
ospvDestinationDevice	not needed in tightly-coupled environments in which the endpoint (e.g. the H.323 Gateway) is not known to the OSP server; empty string ("") in this example
ospvToken	authorization token to present to Gatekeeper B during setup

5) Gatekeeper B accepts the Setup and extends the call to the destination gateway with another Setup / Setup Acknowledge exchange.

If using the TransNexus OSP Toolkit, Gatekeeper B must call the Toolkit function OSPPTransactionValidateAuthorisation() to verify the authorization token in the Setup message. It would call it with the following parameters.

ospvSource	DNS name or IP address of Gatekeeper A, e.g. "[172.16.1.1]"
ospvDestination	DNS name or IP address of Gatekeeper B, for example, "gatekeeperB.itsp.fr"
ospvSourceDevice	not needed in tightly-coupled environments in which the endpoint (e.g. the H.323 Terminal) is not known to the OSP server; empty string ("") in this example

`ospvDestinationDevice`	not needed in tightly-coupled environments in which the endpoint (e.g. the H.323 Gateway) is not known to the OSP server; empty string (`""`) in this example
`ospvCallingNumber`	calling party's E.164 number; this must be the same as was contained in the original `<AuthorisationRequest>`
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvCallId`	H.323 Call Identifier received in Setup message
`ospvToken`	authorization token presented to Gatekeeper B during setup

The `OSPPTransactionValidateAuthorisation()` function will return an indication of whether or not the call is authorized (`ospvAuthorised`), and, if so, how much service is authorized (`ospvTimeLimit`).

6) The Gateway adds the final leg of the call by dialing the called party; the phone conversation can now take place.

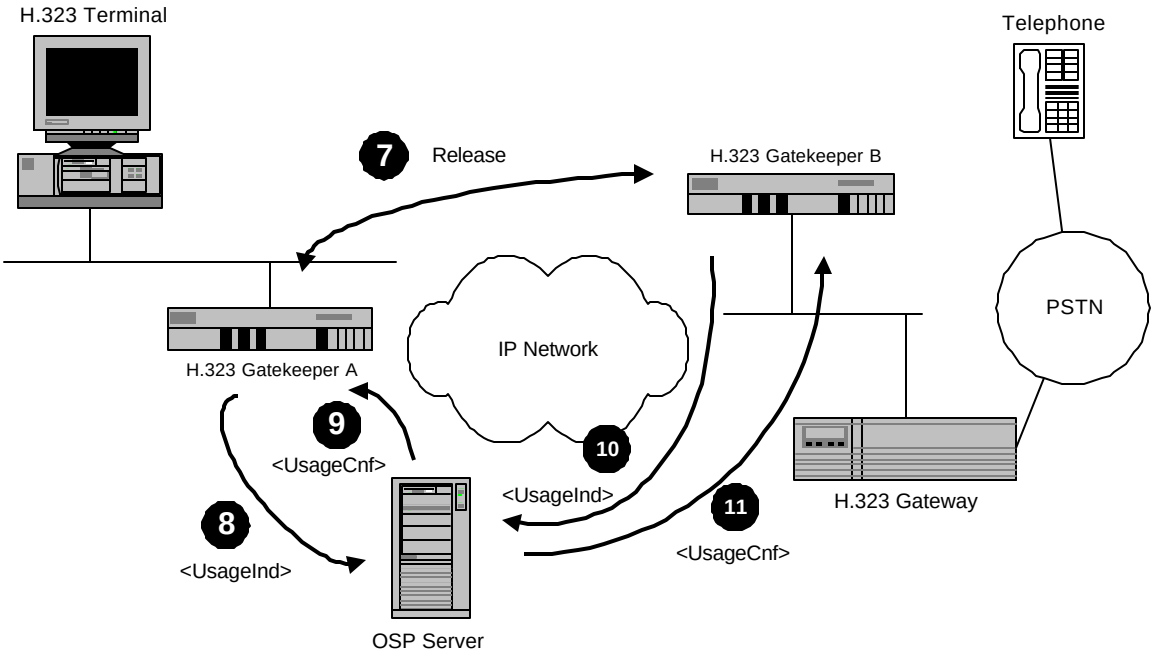
Usage Reports

At the conclusion of the phone call, both gatekeepers must report usage information to the OSP server. Figure 6 identifies the principle steps of a typical call completion.

The steps shown in figure 6 are straightforward.

7) The gatekeepers release the call with an exchange of H.225.0 Release and Release Complete messages.

8) Gatekeeper A then sends a `<UsageIndication>` message to the OSP server. In this example Gatekeeper A may not be able to support protocol extensions such as



• Figure 6 Call Completion with H.323 Gatekeepers.

statistics. Its message, therefore contains just the following elements.

`<Timestamp>`	time of request
`<Role>`	for Gatekeeper A, `source`
`<TransactionId>`	transaction ID assigned by OSP server in authorization response
`<CallId>`	H.323 Call Identifier used for the call
`<SourceInfo type="e164">`	calling party's E.164 number as returned in the authorization response, e.g. `14048724887`
`<SourceAlternate type="transport">`	DNS name or IP address of Gatekeeper A, for example `gatekeeperA.carrier.com`
`<DestinationInfo type="e164">`	called party's E.164 number, e.g. `33492944299`
`<DestinationAlternate type="transport">`	DNS name or IP address of Gatekeeper B, for example, `gatekeeperB.itsp.fr`
`<UsageDetail>`	usage information for the call
`<Service/>`	empty (for basic service)
`<Amount>`	amount of service used, e.g. `300`
`<Increment>`	increment of service measurement, e.g. `1`
`<Unit>`	unit of service measurement, e.g. `s` for seconds

If using the TransNexus OSP Toolkit, Gatekeeper A can generate that message by calling `OSPPTransactionReportUsage()` with the following significant parameters

`ospvDuration`	length of the call in seconds, e.g. `300`
`ospvLossPacketsSent`	value ignored (because of following parameter value)
`ospvLossFractionSent`	`-1` if not reported
`ospvLossPacketsReceived`	value ignored (because of following parameter value)
`ospvLossFractionReceived`	`-1` if not reported

9) The OSP server responds with a `<UsageConfirmation>` message. If it has accepted the usage report, that message will contain a successful `<Status>` element (e.g. `<Code>200</Code>`. Gatekeeper A will learn of this response in the return value from the `OSPPTransactionReportUsage()` function.

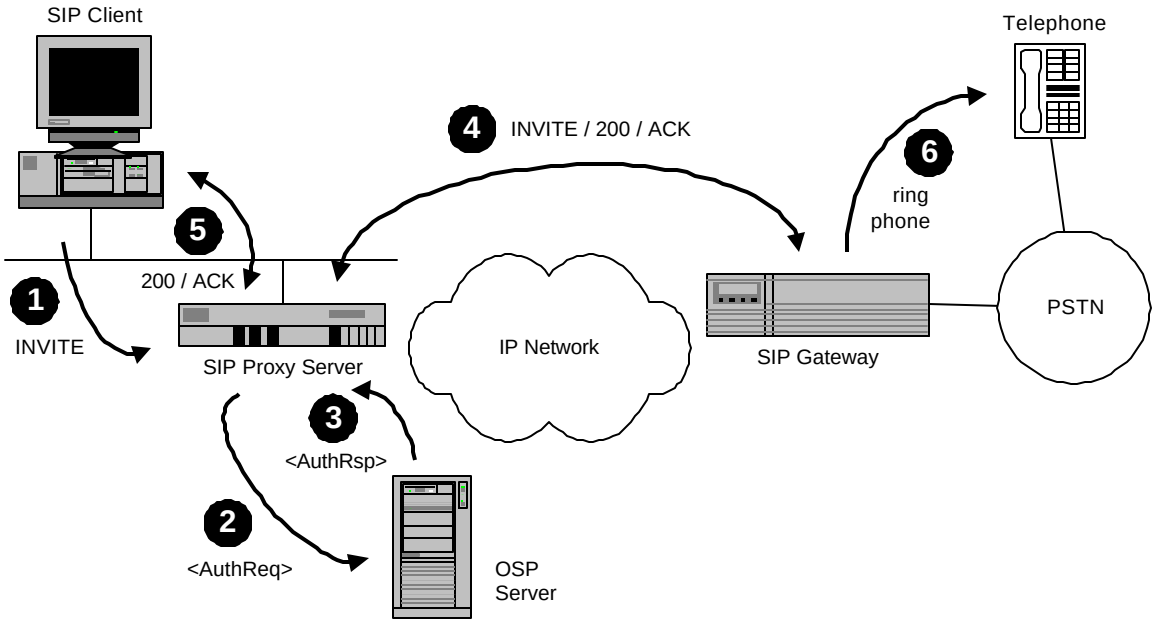
10) Gatekeeper B also sends a `<UsageIndication>` to the OSP server. Assuming that it uses TransNexus extensions for statistics reporting, that message would include the following elements.

`<Timestamp>`	time of request
`<Role>`	for Gatekeeper B, `destination`
`<TransactionId>`	transaction ID assigned by OSP server and passed to Gatekeeper B in authorization token
`<CallId>`	H.323 Call Identifier used for the call
`<SourceInfo type="e164">`	calling party's E.164 number as presented in the Setup message, e.g. `14048724887`
`<SourceAlternate`	DNS name or IP address of H.323 Terminal, for

type="transport">	example [172.16.100.1]
<DestinationInfo type="e164">	called party's E.164 number, e.g. 33492944299
<DestinationAlternate type="transport">	DNS name or IP address of Gatekeeper B, for example, gatekeeperB.itsp.fr
<UsageDetail>	usage information for the call
<Service/>	empty (for basic service)
<Amount>	amount of service used, e.g. 300
<Increment>	Increment of service measurement, e.g. 1
<Unit>	unit of service measurement, e.g. s for seconds
<transnexus.com:Statistics>	statistical information for call
<transnexus.com:LossSent>	loss information for packets sent by Gatekeeper B
<transnexus.com:Packets>	number of packets lost from Gatekeeper B to H.323 Terminal
<transnexus.com:Fraction>	fraction (from 0 to 255) of packets lost from B to H.323 Terminal
<transnexus.com:LossReceived>	loss information for packets sent by H.323 Terminal
<transnexus.com:Packets>	number of packets lost from H.323 Terminal to Gatekeeper B
<transnexus.com:Fraction>	fraction (from 0 to 255) of packets lost from Terminal to B
<transnexus.com:OneWayDelay>	one way delay measured from Terminal to B
<transnexus.com:Minimum>	minimum measured value for delay, in seconds
<transnexus.com:Mean>	sample mean of delay measurements, in seconds
<transnexus.com:Variance>	sample variance of delay measurements, in squared seconds
<transnexus.com:Samples>	number of sample measurements
<transnexus.com:RoundTripDelay>	round trip delay between Terminal and B measured during call
<transnexus.com:Minimum>	minimum measured value for delay, in seconds
<transnexus.com:Mean>	sample mean of delay measurements, in seconds
<transnexus.com:Variance>	sample variance of delay measurements, in squared seconds
<transnexus.com:Samples>	number of sample measurements

If using the TransNexus OSP Toolkit, Gatekeeper B may call two Toolkit functions while the call is in progress: `OSPPTransactionAccumulateOneWayDelay()` for one-way delay, and `OSPPTransactionAccumulateRoundTripDelay()` for round-trip delay measurements. Note that these functions must be called before `OSPPTransactionReportUsage()`. That call is used to indicate loss statistics as well as the call's duration.

11) The OSP server responds with a `<UsageConfirmation>` message. If it has accepted the usage report, that message will contain a successful `<Status>` element (e.g. `<Code>200</Code>`. Gatekeeper B will learn of this response in the return value from the `OSPPTransactionReportUsage()` function.



- Figure 7 Call Routing and Authorization with SIP Proxy Servers.

Session Initiation Protocol Proxy Servers

Session Initiation Protocol (SIP) proxy servers may function in a manner similar to H.323 gatekeepers. In such an environment, the proxy servers may support the Open Settlement Protocol, while the systems on whose behalf they act need not.

Note that all destination gateways (or endpoints) served by the SIP proxies must be equivalent, as far as the OSP server is concerned. Indeed, the OSP server is not even aware of the existence of any endpoints or gateways. Such an architecture, therefore, places implicit restrictions on the services offered by the operators. An operator, for example, will be unable to price termination services differently for different gateways served by the same proxy.

Call Routing and Authorization

Figure 7 shows how OSP plays a role in environments that use SIP Proxy Servers. In the figure, The SIP Proxy Server, acting on behalf of the SIP client, completes a call to a SIP Gateway, and ultimately to the PSTN.

1) The SIP Client begins a call by sending an INVITE request to its proxy server. The INVITE indicates that the called party is identified by an E.164 phone number such as +33 4 92 94 42 99.

2) The SIP Proxy Server sends an OSP `<AuthorisationRequest>` message to the OSP Server. The significant elements within the `<AuthorisationRequest>` include the following.

`<Timestamp>`	time of request
`<CallId>`	SIP Call-ID to be used for the call

`<SourceInfo type="e164">`	a representation of the SIP client using an E.164 number; in the absence of other information, this number may be derived using the IP address to E.164 number mapping of ETSI TIPHON.
`<SourceAlternate type="transport">`	DNS name or IP address of the Proxy Server, for example `proxy.carrier.com`
`<DestinationInfo type="e164">`	called party's E.164 number, e.g. `33492944299`
`<Service/>`	empty (for basic service)
`<MaximumDestinations>`	the maximum number of destinations, including alternatives, the Proxy Server will consider

If using the TransNexus OSP Toolkit, the Proxy Server can generate this message by calling the `OSPPTransactionRequestAuthorisation()` function with the following significant parameters.

`ospvSource`	DNS name or IP address of the Proxy Server, for example `"proxy.carrier.com"`
`ospvSourceDevice`	not needed in tightly-coupled environments in which the endpoint (e.g. the SIP Client) is not known to the OSP server; empty string (`""`) in this example
`ospvCallingNumber`	an E.164-format representation of the SIP Client, possibly derived using the IP address to E.164 number mapping of ETSI TIPHON
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvUser`	optional user identification; empty string (`""`) in this example
`ospvNumberOfCallIds`	the number of SIP Call-IDs given to the OSP server; if all Setup attempts for this call will use the same Call-ID value (as in this example), this parameter value is `1`
`ospvCallIds`	an array (containing, in this example, but a single element) of SIP Call-IDs; the array structure includes a size field which indicates the size of the value
`ospvPreferredDestinations`	optional list of preferred destination gateways; empty string (`""`) in this example
`ospvNumberOfDestinations`	the maximum number of potential destinations that the Proxy Server is prepared to consider for the call; for example 3

3) OSP Server replies with an `<AuthorisationResponse>` message. The message identifies the SIP gateway. In particular, the `<AuthorisationResponse>` contains the following elements.

`<Timestamp>`	time of response
`<Status>`	result of response, e.g. `<Code>200</Code>`
`<TransactionId>`	transaction identifier assigned by settlement provider
`<Destination>`	first destination gateway to try for call
`<DestinationSignalAddress type="transport">`	DNS name or IP address of destination gateway, e.g. `gateway.itsp.fr`
`<Token>`	authorization token to be passed to Gateway
`<ValidAfter>`	time after which token for Gateway is valid
`<ValidUntil>`	time until which token for Gateway is valid
`<UsageDetail>`	how much service is authorized with Gateway

`<Service/>`	empty (for basic service)
`<Amount>`	amount of authorized service, e.g. `3600`
`<Increment>`	increment of service measurement, e.g. `1`
`<Unit>`	unit of service measurement, e.g. `s` for seconds
`<CallId>`	SIP Call-ID to be used for the call to the Gateway

If using the TransNexus OSP Toolkit, the Proxy Server will be informed of the receipt of a valid OSP `<AuthorisationResponse>` by the return value of the call to the function `OSPPTransactionRequestAuthorisation()`.

4) The Proxy Server, acting on behalf of its client, sends an SIP INVITE to the Gateway. In this example, the Gateway accepts the connection with an 200, and the proxy responds with an ACK. This INVITE message also uses the SIP Call-ID from the `<AuthorisationRequest>`, and the INVITE must also include the authorization token(s) provided in the OSP `<AuthorisationResponse>`. Each token should be conveyed in the SIP message body using the application/osp-token MIME type, as defined in http://www.ietf.org/internet-drafts/draft-johnston-sip-osp-token-02.txt.

   If using the TransNexus OSP Toolkit, the Proxy Server retrieves the information needed to access the gateway by calling `OSPPTransactionGetFirstDestination()`. That function will return the following information.

`ospvValidAfter`	time after which authorization token for the Gateway is valid
`ospvValidUntil`	time until which authorization token for the Gateway is valid
`ospvTimeLimit`	amount of service authorized with the Gateway, e.g. `3600` (seconds)
`ospvCallId`	SIP Call-ID to use in INVITE message to the gateway
`ospvCalledNumber`	number to present to the Gateway as the called party's number; often the same as passed to `OSPPTransactionRequestAuthorisation()`, e.g. `"33492944299"`, but not, for example, if the OSP server performs number translation
`ospvDestination`	DNS name or IP address of the Gateway, for example `"gateway.itsp.fr"`
`ospvDestinationDevice`	not needed when the destination device is directly reachable from the source device
`ospvToken`	authorization token to present to the Gateway during setup

5) The Proxy Server, on establishing the connection with the Gateway, completes the connection with its client as well with a SIP 200 / ACK exchange.

6) The Gateway, meanwhile, completes the call to the destination phone number.

   If using the TransNexus OSP Toolkit, the Gateway must call the Toolkit function `OSPPTransactionValidateAuthorisation()` to verify the authorization token in the INVITE message. It would call it with the following parameters.

`ospvSource`	DNS name or IP address of the Proxy Server, e.g. `"[172.16.1.1]"`
`ospvDestination`	DNS name or IP address of the Gateway for example, `"gateway.itsp.fr"`

	`"gateway.itsp.fr"`
`ospvSourceDevice`	not needed in tightly-coupled environments in which the endpoint (e.g. the SIP Client) is not known to the OSP server; empty string (`""`) in this example
`ospvDestinationDevice`	not needed in tightly-coupled environments in which the endpoint (e.g. the Gateway) is directly reachable from the source; empty string (`""`) in this example
`ospvCallingNumber`	calling party's E.164 number; this must be the same as was contained in the original `<AuthorisationRequest>`
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvCallId`	SIP Call-ID received in Setup message
`ospvToken`	authorization token presented to the Gateway during setup

The `OSPPTransactionValidateAuthorisation()` function will return an indication of whether or not the call is authorized (`ospvAuthorised`), and, if so, how much service is authorized (`ospvTimeLimit`).

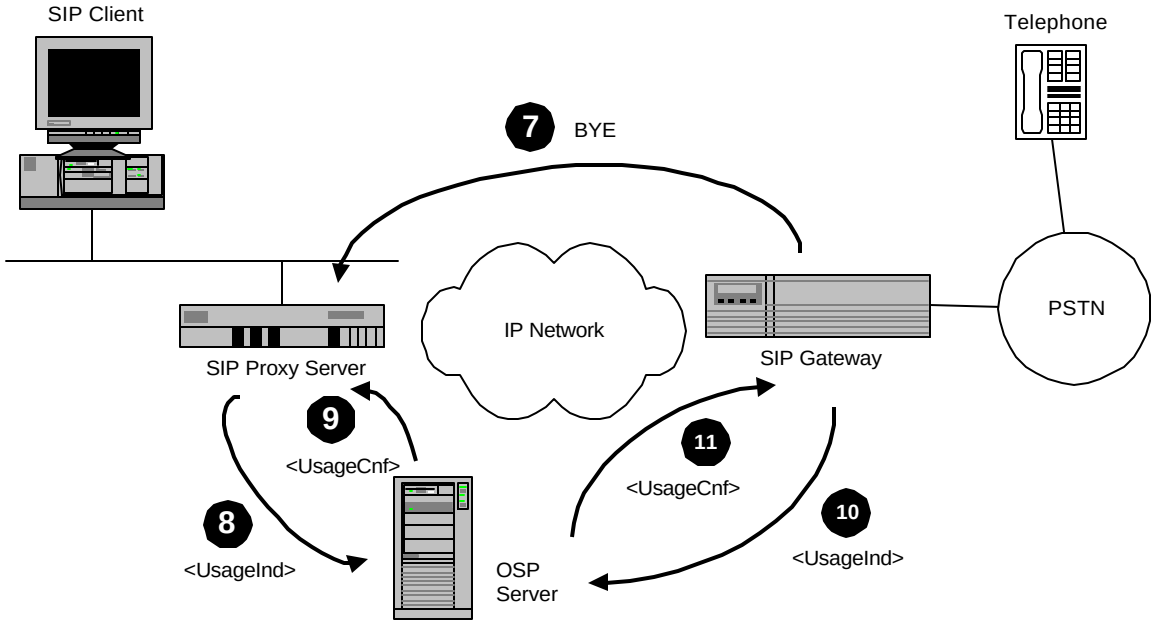
Usage Reports

At the conclusion of the phone call, both the Proxy Server and Gateway report usage information to the OSP server. Figure 8 identifies the principle steps of a typical call completion.

The steps shown in figure 8 are straightforward.

7) The Proxy Server and Gateway release the call by transferring a SIP BYE message.

8) The Proxy Server then sends a `<UsageIndication>` message to the OSP server. In this example, the Proxy Server does not support protocol extensions such as



• Figure 8 Call Completion with SIP proxy servers.

statistics. Its message, therefore contains just the following elements.

`<Timestamp>`	time of request
`<Role>`	for Proxy Server, `source`
`<TransactionId>`	transaction ID assigned by OSP server in authorization response
`<CallId>`	SIP Call-ID used for the call
`<SourceInfo type="e164">`	calling party's E.164 number as returned in the authorization response, e.g. `14048724887`
`<SourceAlternate type="transport">`	DNS name or IP address of the Proxy Server, for example `proxy.carrier.com`
`<DestinationInfo type="e164">`	called party's E.164 number, e.g. `33492944299`
`<DestinationAlternate type="transport">`	DNS name or IP address of the Gateway, for example, `gateway.itsp.fr`
`<UsageDetail>`	usage information for the call
`<Service/>`	empty (for basic service)
`<Amount>`	amount of service used, e.g. `300`
`<Increment>`	increment of service measurement, e.g. `1`
`<Unit>`	unit of service measurement, e.g. `s` for seconds

If using the TransNexus OSP Toolkit, the proxy server can generate that message by calling `OSPPTransactionReportUsage()` with the following significant parameters

`ospvDuration`	length of the call in seconds, e.g. `300`
`ospvLossPacketsSent`	value ignored (because of following parameter value)
`ospvLossFractionSent`	`-1` if not reported
`ospvLossPacketsReceived`	value ignored (because of following parameter value)
`ospvLossFractionReceived`	`-1` if not reported

9) The OSP server responds with a `<UsageConfirmation>` message. If it has accepted the usage report, that message will contain a successful `<Status>` element (e.g. `<Code>200</Code>`. The Proxy Server will learn of this response in the return value from the `OSPPTransactionReportUsage()` function.

10) The Gateway also sends a `<UsageIndication>` to the OSP server. Assuming that it uses TransNexus extensions for statistics reporting, that message would include the following elements.

`<Timestamp>`	time of request
`<Role>`	for the Gateway, `destination`
`<TransactionId>`	transaction ID assigned by OSP server and passed to the gateway in authorization token
`<CallId>`	SIP Call-ID used for the call
`<SourceInfo type="e164">`	calling party's E.164 number as presented in the INVITE message, e.g. `14048724887`
`<SourceAlternate`	DNS name or IP address of the Proxy Server, for

type="transport">	example [172.16.100.1]
<DestinationInfo type="e164">	called party's E.164 number, e.g. 33492944299
<DestinationAlternate type="transport">	DNS name or IP address of the Gateway, for example, gateway.itsp.fr
<UsageDetail>	usage information for the call
<Service/>	empty (for basic service)
<Amount>	amount of service used, e.g. 300
<Increment>	Increment of service measurement, e.g. 1
<Unit>	unit of service measurement, e.g. s for seconds
<transnexus.com:Statistics>	statistical information for call
<transnexus.com:LossSent>	loss information for packets sent by the Gateway
<transnexus.com:Packets>	number of packets lost from the Gateway to the Proxy Server
<transnexus.com:Fraction>	fraction (from 0 to 255) of packets lost from the Gateway to the Proxy Server
<transnexus.com:LossReceived>	loss information for packets sent by the Proxy Server
<transnexus.com:Packets>	number of packets lost from Proxy Server to the Gateway
<transnexus.com:Fraction>	fraction (from 0 to 255) of packets lost from Proxy Server to Gateway
<transnexus.com:OneWayDelay>	one way delay measured from Proxy Server to Gateway
<transnexus.com:Minimum>	minimum measured value for delay, in seconds
<transnexus.com:Mean>	sample mean of delay measurements, in seconds
<transnexus.com:Variance>	sample variance of delay measurements, in squared seconds
<transnexus.com:Samples>	number of sample measurements
<transnexus.com:RoundTripDelay>	round trip delay between Proxy Server and Gateway measured during call
<transnexus.com:Minimum>	minimum measured value for delay, in seconds
<transnexus.com:Mean>	sample mean of delay measurements, in seconds
<transnexus.com:Variance>	sample variance of delay measurements, in squared seconds
<transnexus.com:Samples>	number of sample measurements

If using the TransNexus OSP Toolkit, Gateway B may call two Toolkit functions while the call is in progress: `OSPPTransactionAccumulateOneWayDelay()` for one-way delay, and `OSPPTransactionAccumulateRoundTripDelay()` for round-trip delay measurements. Note that these functions must be called before `OSPPTransactionReportUsage()`. That call is used to indicate loss statistics as well as the call's duration.

11) The OSP server responds with a `<UsageConfirmation>` message. If it has accepted the usage report, that message will contain a successful `<Status>` element (e.g. `<Code>200</Code>`. The Gateway will learn of this response in the return value from the `OSPPTransactionReportUsage()` function.

Loosely Controlled Distributed Architecture

The Open Settlement Protocol can also be used in environments based on loosely-coupled, distributed architectures. One such environment is an H.323-based architecture with direct call signaling. Another example is the use Session Initiation Protocol (SIP) redirect servers. This subsection describes the use of OSP and the OSP Toolkit in each of these architectures.

H.323 Direct Routed Calls (with Gatekeepers)

When H.323 gateways and gatekeepers both implement the Open Settlement Protocol, it is possible to use OSP in an architecture that relies on H.323 Direct Call Signaling (as opposed to Gatekeeper Call Signaling). Such an approach is a hybrid of the peer-to-peer gateway architecture and the tightly controlled gatekeeper model discussed in the previous two subsections. In this approach, the gatekeepers acts as agents for call routing and authorization, but the gateways themselves are responsible for establishing and disconnecting calls directly with each other. In addition to gateways, H.323 proxy devices may also be used to support this model on behalf of H.323 terminals.

Call Routing and Authorization

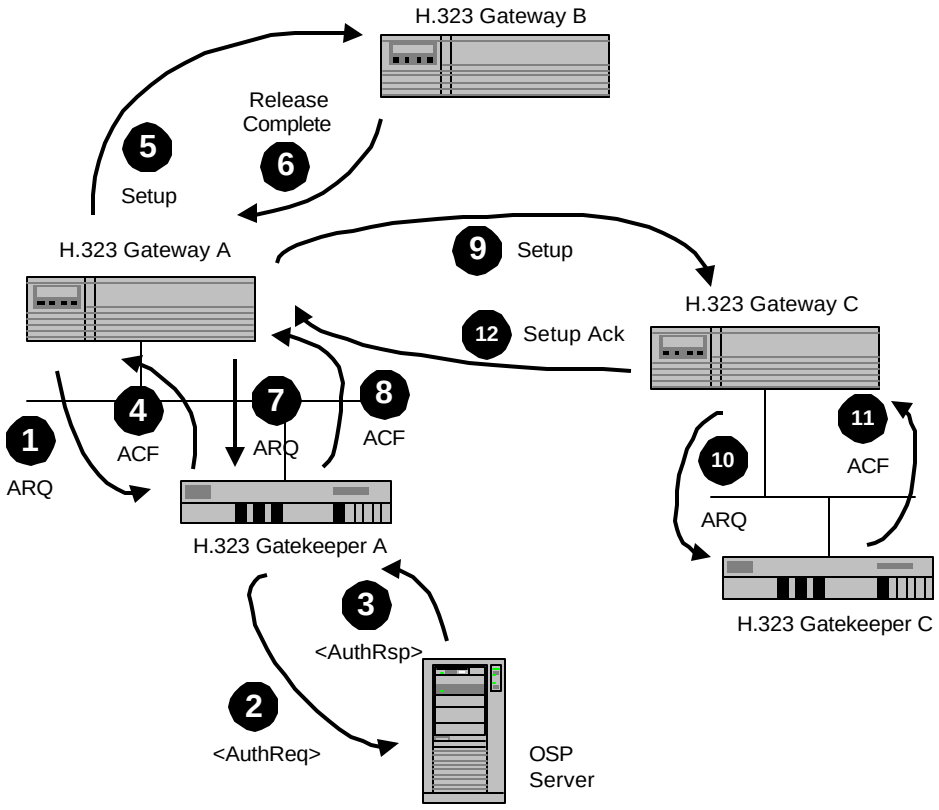
Figure 9 shows a sample routing and authorization scenario for this model. Note that the figure shows a finer level of detail than previous examples in order to clarify several subtle points.

1) Gateway A begins a call by sending an H.225.0 Admission Request (ARQ) to its gatekeeper, Gatekeeper A. The ARQ indicates that the called party is identified by an E.164 phone number such as +33 4 92 94 42 99.

2) Gatekeeper A sends an OSP `<AuthorisationRequest>` message to the OSP Server. The significant elements within the `<AuthorisationRequest>` include

`<Timestamp>`	time of request
`<CallId>`	H.323 Call Identifier to be used for the call
`<SourceInfo type="e164">`	the called party's E.164 number, or, if that is not available, a local E.164 number belonging to Gateway A; this number must be passed to the destination gateway(s) in Setup messages
`<SourceAlternate type="transport">`	DNS name or IP address of Gatekeeper A, for example `gatekeeperA.carrier.com`
`<SourceAlternate type="h323">`	H.323 identifier of Gateway A, e.g. `12345678`
`<DestinationInfo type="e164">`	called party's E.164 number, e.g. `33492944299`
`<Service/>`	empty (for basic service)
`<MaximumDestinations>`	the maximum number of destinations, including alternatives, Gatekeeper A will consider

If using the TransNexus OSP Toolkit, Gatekeeper A can generate this message by calling the `OSPPTransactionRequestAuthorisation()` function with the following significant parameters.



• Figure 9 Call Routing and Authorization for H.323 Direct Signaled Calls.

`ospvSource`	DNS name or IP address of Gatekeeper A, for example `"gatekeeperA.carrier.com"`
`ospvSourceDevice`	the H.323 identifier of Gateway A as a character string, for example `"12345678"`
`ospvCallingNumber`	the called party's E.164 number, or, if that is not available, a local E.164 number belonging to Gateway A; this number must be passed to the destination gateway(s) in Setup messages
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvUser`	optional user identification; empty string (`""`) in this example
`ospvNumberOfCallIds`	the number of H.323 Call Identifiers given to the OSP server; if all Setup attempts for this call will use the same Call Identifier value (as in this example), this parameter value is `1`
`ospvCallIds`	an array (containing, in this example, but a single element) of H.323 Call Identifiers; the array structure includes a size field which indicates the size of the value
`ospvPreferredDestinations`	optional list of preferred destination systems; empty string (`""`) in this example
`ospvNumberOfDestinations`	the maximum number of potential destinations that Gatekeeper A is prepared to consider for the call; for example 3

3) OSP Server replies with an `<AuthorisationResponse>` message. The message identifies Gateway B and Gateway C as candidate destinations. In particular, the `<AuthorisationResponse>` contains the following elements.

<Timestamp>	time of response
<Status>	result of response, e.g. <Code>200</Code>
<TransactionId>	transaction identifier assigned by settlement provider
<Destination>	first destination gateway to try for call
<DestinationSignalAddress type="transport">	DNS name or IP address of Gateway B, for example gatewayB.itsp.fr
<Token>	authorization token to be passed to Gateway B
<ValidAfter>	time after which token for Gateway B is valid
<ValidUntil>	time until which token for Gateway B is valid
<UsageDetail>	how much service is authorized with Gateway B
<Service/>	empty (for basic service)
<Amount>	amount of authorized service, e.g. 3600
<Increment>	increment of service measurement, e.g. 1
<Unit>	unit of service measurement, e.g. s for seconds
<CallId>	H.323 Call Identifier to be used for the call to Gateway B
<Destination>	second destination gateway to try for call
<DestinationSignalAddress type="transport">	DNS name or IP address of destination Gateway C, e.g. gatewayC.isp.fr
<Token>	authorization token to be passed to Gateway C
<ValidAfter>	time after which token for Gateway C is valid
<ValidUntil>	time until which token for Gateway C is valid
<UsageDetail>	how much service is authorized with Gateway C
<Service/>	empty (for basic service)
<Amount>	amount of authorized service, e.g. 3600
<Increment>	increment of service measurement, e.g. 1
<Unit>	unit of service measurement, e.g. s for seconds
<CallId>	H.323 Call Identifier to be used for the call to Gateway C

If using the TransNexus OSP Toolkit, Gatekeeper A will be informed of the receipt of a valid OSP <AuthorisationResponse> by the return value of the call to the function OSPPTransactionRequestAuthorisation().

4) Gatekeeper A sends an H.225.0 Admission Confirm (ACF) message to its gateway. That ACF identifies the destination as Gateway B and it must include the authorization token for gateway B.

If using the TransNexus OSP Toolkit, Gatekeeper A can retrieve the information needed to identify and gain access to Gateway B by calling the library function OSPPTransactionGetFirstDestination(). That function will return the following information.

ospvValidAfter	time after which authorization token for Gateway B is valid
ospvValidUntil	time until which authorization token for Gateway B is valid
ospvTimeLimit	amount of service authorized with Gateway B, e.g. 3600 (seconds)

`ospvCallId`	H.323 Call Identifier to use in Setup message to Gateway B
`ospvCalledNumber`	number to present to Gateway B as the called party's number; often the same as passed to `OSPPTransactionRequestAuthorisation()`, e.g. `"33492944299"`, but not, for example, if the OSP server performs number translation
`ospvDestination`	DNS name or IP address of the destination gateway, for example `"gatewayB.itsp.fr"`
`ospvDestinationDevice`	not needed when the source system will contact the destination device directly; empty string (`""`) in this example
`ospvToken`	authorization token to present to Gateway B during setup

5) Gateway A sends an H.225.0 Setup message to Gateway B. This Setup message uses the H.323 Call Identifier from the `<AuthorisationRequest>`, and the message must include the authorization token(s) provided in the OSP `<AuthorisationResponse>`.

   If using the OSP Toolkit, Gateway A must also create and initialize a transaction to track its interaction with Gateway B. It does this with a call to the library function `OSPPTransactionInitializeAtDevice()`. The significant parameters to that function call include the following.

`ospvIsSource`	indicates that the local system is a source for the phone call; `1` in the example
`ospvSource`	DNS name or IP address of Gateway A, for example `"gatewayA.carrier.com"`
`ospvDestination`	DNS name or IP address of Gateway B, for example, `"gatewayB.itsp.fr"`
`ospvSourceDevice`	the H.323 identifier of Gateway A as a character string, for example `"12345678"`
`ospvDestinationDevice`	not needed when the source system will contact the destination device directly; empty string (`""`) in this example
`ospvCallingNumber`	calling party's E.164 number; this must be the same as was contained in the original `<AuthorisationRequest>`
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvCallId`	H.323 Call Identifier to be sent in Setup message
`ospvToken`	authorization token to be presented to Gateway B during setup

6) Gateway B refuses the setup attempt with a Release Complete message, perhaps, for example, because no outbound PSTN ports are available.

7) Gateway A sends another Admission Request (ARQ) message to its gatekeeper to request an alternate destination for the phone call. Note that this ARQ must use the same H.323 Call Identifier as the original ARQ. For clarity, the example omits other details of the exchange between Gateway A and Gatekeeper A, such as, for example, a Disengage Request (DRQ) and Disengage Confirm (DCF).

8) Gatekeeper A replies with an Admission Confirm (ACF) message that identifies Gateway C as a potential destination. Note that the Gatekeeper need not query the OSP server to get that information. Rather, the information is available in the original `<AuthorisationResponse>` noted in step 3.

Gatekeeper A may retrieve the information needed to identify Gateway C by calling the library function `OSPPTransactionGetNextDestination`. It must pass to that function a reason for the failure of the first destination. In the example that might be the value `OSPC_FAIL_REMOTE_EXT`. The function will return the following.

`ospvValidAfter`	time after which authorization token for Gateway C is valid
`ospvValidUntil`	time until which authorization token for Gateway C is valid
`ospvTimeLimit`	amount of service authorized with Gateway C, e.g. `3600` (seconds)
`ospvCallId`	H.323 Call Identifier to use in Setup message to Gateway C
`ospvCalledNumber`	number to present to Gateway C as the called party's number; often the same as passed to `OSPPTransactionRequestAuthorisation()`, e.g. `"33492944299"`, but not, for example, if the OSP server performs number translation
`ospvDestination`	DNS name or IP address of the destination gateway, for example `"gatewayC.isp.fr"`
`ospvDestinationDevice`	not needed when the source system will contact the destination device directly; empty string (`""`) in this example
`ospvToken`	authorization token to present to Gateway C during setup

9) Gateway A tries a second setup attempt, this time by sending a Setup message to Gateway C.

If using the OSP Toolkit, Gateway A must also re-initialize the transaction to indicate that the peer is now expected to be Gateway C. A call to the library function `OSPPTransactionReinitializeAtDevice()` accomplishes this.

`ospvFailureReason`	an identification of the reason for the failure of the previous setup attempt, for example `OSPC_FAIL_REMOTE_EXT`
`ospvIsSource`	indicates that the local system is a source for the phone call; `1` in the example
`ospvSource`	DNS name or IP address of Gateway A, for example `"gatewayA.carrier.com"`
`ospvDestination`	DNS name or IP address of Gateway C, for example, `"gatewayC.isp.fr"`
`ospvSourceDevice`	the H.323 identifier of Gateway A as a character string, for example `"12345678"`
`ospvDestinationDevice`	not needed when the source system will contact the destination device directly; empty string (`""`) in this example
`ospvCallingNumber`	calling party's E.164 number; this must be the same as was contained in the original `<AuthorisationRequest>`
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvCallId`	H.323 Call Identifier to be sent in Setup message
`ospvToken`	authorization token to be presented to Gateway C during setup

10) Gateway C receives the Setup message and asks its gatekeeper for permission to accept the call. It does so by sending an Admission Request (ARQ) to Gatekeeper C. Note that this message must include the authorization token(s) received in the Setup.

11) Gatekeeper C authorizes the call and returns an Admission Confirm (ACF) message to Gateway C.

If using the TransNexus OSP Toolkit, Gatekeeper C must call the Toolkit function `OSPPTransactionValidateAuthorisation()` to verify the authorization token in the ARQ message. It would call it with the following parameters.

`ospvSource`	DNS name or IP address of Gateway A, e.g. `"[172.16.1.1]"`
`ospvDestination`	DNS name or IP address of Gatekeeper C, for example, `"gatekeeperC.isp.fr"`
`ospvSourceDevice`	not needed when the setup message is received directly from a source device; empty string (`""`) in this example
`ospvDestinationDevice`	H.323 Identifier of Gateway C, for example `"987654321"`
`ospvCallingNumber`	calling party's E.164 number; this must be the same as was contained in the original `<AuthorisationRequest>`
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvCallId`	H.323 Call Identifier received in Setup message
`ospvToken`	authorization token presented to Gateway C during setup

The `OSPPTransactionValidateAuthorisation()` function will return an indication of whether or not the call is authorized (`ospvAuthorised`), and, if so, how much service is authorized (`ospvTimeLimit`).

12) Gateway C receives the ACF and accepts the call by returning a Setup Acknowledge message to Gateway A.

If Gateway is using the OSP Toolkit, it must also create a transaction object to track the call. It does so via the function `OSPPTransactionInitializeAtDevice()`.

`ospvIsSource`	indicates that the local system is not a source for the phone call; `0` in the example
`ospvSource`	DNS name or IP address of Gateway A, for example `"[172.16.1.1]"`
`ospvDestination`	DNS name or IP address of Gateway C, for example, `"gatewayC.isp.fr"`
`ospvSourceDevice`	not needed when the setup message is received directly from a source device; ; empty string (`""`) in this example
`ospvDestinationDevice`	not needed the destination device is the reporting system; empty string (`""`) in this example
`ospvCallingNumber`	calling party's E.164 number; this must be the same as was contained in the original `<AuthorisationRequest>`
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvCallId`	H.323 Call Identifier from the Setup message
`ospvToken`	authorization token presented to Gateway C during setup

Usage Reports

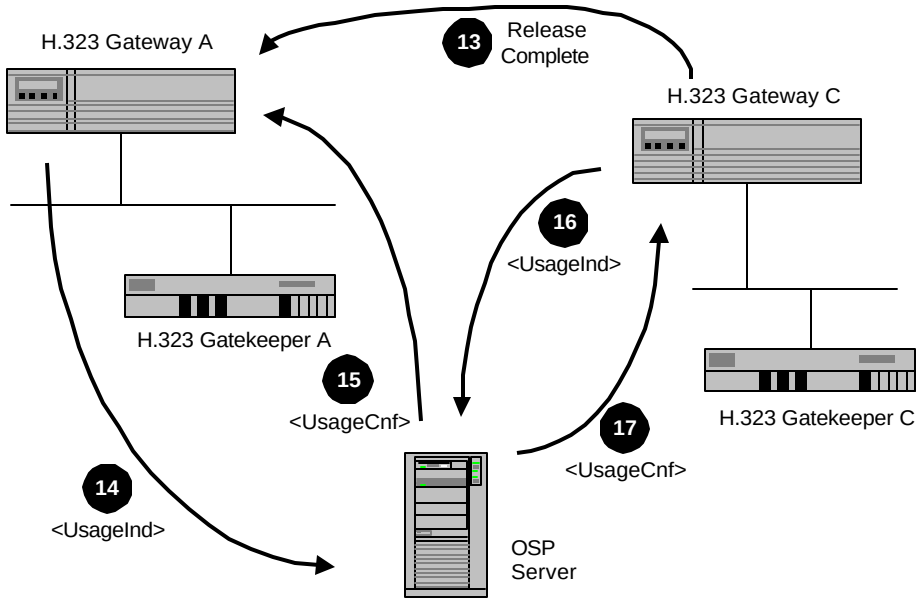
At the conclusion of the phone call, both gateways must report usage information to the OSP server. Figure 10 identifies the principle steps of a typical call completion.

The steps shown in figure 10 are straightforward.

13) The gateways close the connection by exchanging H.323 Release and Release Complete messages

14) Gateway A then sends a <UsageIndication> message to the OSP server. In this example Gateway A's message will include two complete <UsageIndication> components, one for the failed attempt and one for the successful call. The sub-elements for each will include the following.

<UsageIndication>	usage information for the failed setup attempt
<Timestamp>	time of request
<Role>	for Gateway A, source
<TransactionId>	transaction ID assigned by OSP server in authorization response
<CallId>	H.323 Call Identifier used for the call
<SourceInfo type="e164">	calling party's E.164 number, e.g. 14048724887
<SourceAlternate type="transport">	DNS name or IP address of Gateway A, for example gatewayA.carrier.com
<DestinationInfo type="e164">	called party's E.164 number as returned in the authorization response, e.g. 33492944299
<DestinationAlternate type="transport">	DNS name or IP address of Gateway B, for example, gatewayB.itsp.fr
<transnexus.com:FailureReason>	reason for failure of attempted setup, e.g. 422
<UsageIndication>	usage information for the successful setup attempt
<Timestamp>	time of request
<Role>	for Gateway A, source



• Figure 10 Call Completion with H.323 Gateways and Gatekeepers.

`<TransactionId>`	transaction ID assigned by OSP server in authorization response
`<CallId>`	H.323 Call Identifier used for the call
`<SourceInfo type="e164">`	calling party's E.164 number, e.g. `14048724887`
`<SourceAlternate type="transport">`	DNS name or IP address of Gateway A, for example `gatekeeperA.carrier.com`
`<DestinationInfo type="e164">`	called party's E.164 number as returned in the authorization response, e.g. `33492944299`
`<DestinationAlternate type="transport">`	DNS name or IP address of Gateway C, for example, `gatewayC.isp.fr`
`<UsageDetail>`	usage information for the call
`<Service/>`	empty (for basic service)
`<Amount>`	amount of service used, e.g. `300`
`<Increment>`	increment of service measurement, e.g. `1`
`<Unit>`	unit of service measurement, e.g. `s` for seconds

If using the TransNexus OSP Toolkit, Gateway A can generate that message by calling `OSPPTransactionReportUsage()` with the following significant parameters

`ospvDuration`	length of the call in seconds, e.g. `300`
`ospvLossPacketsSent`	value ignored (because of following parameter value)
`ospvLossFractionSent`	`-1` if not reported
`ospvLossPacketsReceived`	value ignored (because of following parameter value)
`ospvLossFractionReceived`	`-1` if not reported

15) The OSP server responds with a `<UsageConfirmation>` message. If it has accepted the usage report, that message will contain a successful `<Status>` element (e.g. `<Code>200</Code>`. Gateway A will learn of this response in the return value from the `OSPPTransactionReportUsage()` function.

16) Gateway C also sends a `<UsageIndication>` to the OSP server. Assuming that it uses TransNexus extensions for statistics reporting, that message would include the following elements.

`<Timestamp>`	time of request
`<Role>`	for Gateway C, `destination`
`<TransactionId>`	Transaction ID assigned by OSP server and passed to Gateway C in authorization token
`<CallId>`	H.323 Call Identifier used for the call
`<SourceInfo type="e164">`	calling party's E.164 number as presented in the Setup message, e.g. `14048724887`
`<SourceAlternate type="transport">`	DNS name or IP address of Gateway A, for example `[172.16.100.1]`
`<DestinationInfo type="e164">`	called party's E.164 number, e.g. `33492944299`
`<DestinationAlternate type="transport">`	DNS name or IP address of Gateway C, for example, `gatewayC.itsp.fr`

`<UsageDetail>`	usage information for the call
`<Service/>`	empty (for basic service)
`<Amount>`	amount of service used, e.g. `300`
`<Increment>`	Increment of service measurement, e.g. `1`
`<Unit>`	unit of service measurement, e.g. `s` for seconds
`<transnexus.com:Statistics>`	statistical information for call
`<transnexus.com:LossSent>`	loss information for packets sent by Gateway C
`<transnexus.com:Packets>`	number of packets lost from Gateway C to Gateway A
`<transnexus.com:Fraction>`	fraction (from 0 to 255) of packets lost from C to A
`<transnexus.com:LossReceived>`	loss information for packets sent by Gateway A
`<transnexus.com:Packets>`	number of packets lost from Gateway A to Gateway C
`<transnexus.com:Fraction>`	fraction (from 0 to 255) of packets lost from A to C
`<transnexus.com:OneWayDelay>`	one way delay measured from A to C
`<transnexus.com:Minimum>`	minimum measured value for delay, in seconds
`<transnexus.com:Mean>`	sample mean of delay measurements, in seconds
`<transnexus.com:Variance>`	sample variance of delay measurements, in squared seconds
`<transnexus.com:Samples>`	number of sample measurements
`<transnexus.com:RoundTripDelay>`	round trip delay between A and C measured during call
`<transnexus.com:Minimum>`	minimum measured value for delay, in seconds
`<transnexus.com:Mean>`	sample mean of delay measurements, in seconds
`<transnexus.com:Variance>`	sample variance of delay measurements, in squared seconds
`<transnexus.com:Samples>`	number of sample measurements

If using the TransNexus OSP Toolkit, Gateway C may call two Toolkit functions while the call is in progress: `OSPPTransactionAccumulateOneWayDelay()` for one-way delay, and `OSPPTransactionAccumulateRoundTripDelay()` for round-trip delay measurements. Note that these functions must be called before `OSPPTransactionReportUsage()`. That call is used to indicate loss statistics as well as the call's duration.

17) The OSP server responds with a `<UsageConfirmation>` message. If it has accepted the usage report, that message will contain a successful `<Status>` element (e.g. `<Code>200</Code>`. Gateway C will learn of this response in the return value from the `OSPPTransactionReportUsage()` function.

Session Initiation Protocol Redirect Servers

Session Initiation Protocol (SIP) redirect servers represent another example of the loosely coupled distributed architecture, particularly on the source side of a call. In this environment, the redirect server provides the call routing and authorization on behalf of the systems it serves, but those systems themselves report usage information.

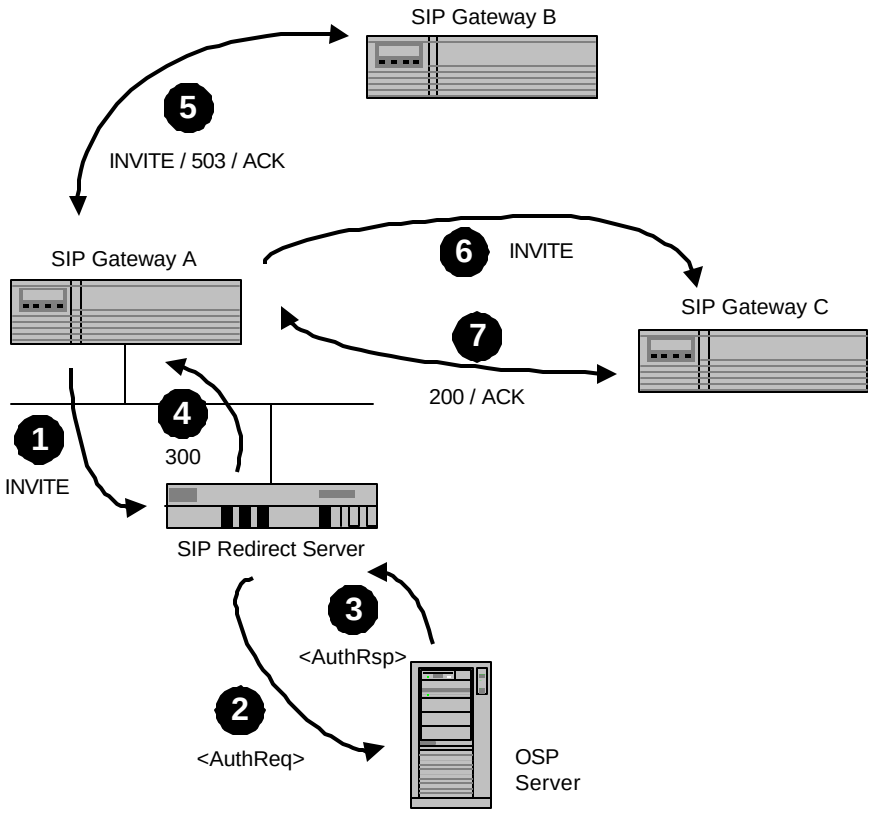
Call Routing and Authorization

Figure 11 shows a sample routing and authorization scenario for SIP redirect servers.

1) Gateway A begins a call by sending a SIP INVITE method to its redirect server. The INVITE indicates that the called party is identified by an E.164 phone number such as +33 4 92 94 42 99.

2) The Redirect Server sends an OSP `<AuthorisationRequest>` message to the OSP Server. The significant elements within the `<AuthorisationRequest>` include

`<Timestamp>`	time of request
`<CallId>`	SIP Call-ID to be used for the call
`<SourceInfo type="e164">`	calling party's E.164 number if available; otherwise a local E.164 number controlled by Gateway A (e.g. `14048724887`); this number must be passed to the destination gateway(s) in future INVITE messages
`<SourceAlternate type="transport">`	DNS name or IP address of Redirect Server, for example `redirect.carrier.com`
`<SourceAlternate type="h323">`	DNS name or IP address of Gateway A, for example `gatewayA.carrier.com`; the type "h323", in this case, implies a device-specific ID rather than a particular protocol



• Figure 11 Call Routing and Authorization with SIP Redirect Servers.

`<DestinationInfo type="e164">`	called party's E.164 number, e.g. `33492944299`
`<Service/>`	empty (for basic service)
`<MaximumDestinations>`	the maximum number of destinations, including alternatives, Redirect Server will consider

If using the TransNexus OSP Toolkit, the redirect server can generate this message by calling the `OSPPTransactionRequestAuthorisation()` function with the following significant parameters.

`ospvSource`	DNS name or IP address of redirect server, for example `"redirect.carrier.com"`
`ospvSourceDevice`	DNS name or IP address of Gateway A, for example `"gatewayA.carrier.com"`
`ospvCallingNumber`	calling party's E.164 number if available; otherwise a local E.164 number controlled by Gateway A (e.g. 14048724887); this number must be passed to the destination gateway(s) in future INVITE messages
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvUser`	optional user identification; empty string (`""`) in this example
`ospvNumberOfCallIds`	the number of SIP Call-IDs given to the OSP server; if all Setup attempts for this call will use the same Call-ID value (as in this example), this parameter value is `1`
`ospvCallIds`	an array (containing, in this example, but a single element) of SIP Call-IDs; the array structure includes a size field which indicates the size of the value
`ospvPreferredDestinations`	optional list of preferred destination gateways; empty string (`""`) in this example
`ospvNumberOfDestinations`	the maximum number of potential destinations that the Redirect Server is prepared to consider for the call; for example 3

3) OSP Server replies with an `<AuthorisationResponse>` message. The message identifies Gateway B and Gateway C as candidate destinations. In particular, the `<AuthorisationResponse>` contains the following elements.

`<Timestamp>`	time of response
`<Status>`	result of response, e.g. `<Code>200</Code>`
`<TransactionId>`	transaction identifier assigned by settlement provider
`<Destination>`	first destination gateway to try for call
`<DestinationSignalAddress type="transport">`	DNS name or IP address of Gateway B, for example `gatewayB.itsp.fr`
`<Token>`	authorization token to be passed to Gateway B
`<ValidAfter>`	time after which token for Gateway B is valid
`<ValidUntil>`	time until which token for Gateway B is valid
`<UsageDetail>`	how much service is authorized with Gateway B
`<Service/>`	empty (for basic service)
`<Amount>`	amount of authorized service, e.g. `3600`
`<Increment>`	increment of service measurement, e.g. `1`

`<Unit>`	unit of service measurement, e.g. `s` for seconds
`<CallId>`	SIP Call-ID to be used for the call to Gateway B
`<Destination>`	second destination gateway to try for call
`<DestinationSignalAddress type="transport">`	DNS name or IP address of Gateway C, for example `gatewayC.isp.fr`
`<Token>`	authorization token to be passed to Gateway C
`<ValidAfter>`	time after which token for Gateway C is valid
`<ValidUntil>`	time until which token for Gateway C is valid
`<UsageDetail>`	how much service is authorized with Gateway C
`<Service/>`	empty (for basic service)
`<Amount>`	amount of authorized service, e.g. `3600`
`<Increment>`	increment of service measurement, e.g. `1`
`<Unit>`	unit of service measurement, e.g. `s` for seconds
`<CallId>`	SIP Call-ID to be used for the call to Gateway C

If using the TransNexus OSP Toolkit, the Redirect Server will be informed of the receipt of a valid OSP `<AuthorisationResponse>` by the return value of the call to the function `OSPPTransactionRequestAuthorisation()`.

4) The Redirect Server returns a "300 Multiple Choices" redirection status code to Gateway A. This response relays the data from the `<AuthorisationResponse>` to the gateway, explicitly identifying Gateways B and C as candidate destinations. This response must also include the authorization tokens returned by the OSP Server as application/osp-token MIME types in the response body.

   If using the TransNexus OSP Toolkit, the Redirect Server can retrieve the information needed to identify and gain access to these destinations by calling the function `OSPPTransactionGetFirstDestination()` (once) and then the library function `OSPPTransactionGetNextDestination()` (repeatedly) until all destinations have been retrieved. Note that the server will not have a failure reason when calling the later function; it may use the value `OSPC_FAIL_NONE` for that parameter. As a example of the results from these functions, the following table shows the information available from the `OSPPTransactionGetFirstDestination()` call.

`ospvValidAfter`	time after which authorization token for Gateway B is valid
`ospvValidUntil`	time until which authorization token for Gateway B is valid
`ospvTimeLimit`	amount of service authorized with Gateway B, e.g. `3600` (seconds)
`ospvCallId`	SIP Call-ID to use in INVITE message to Gateway B
`ospvCalledNumber`	number to present to Gateway B as the called party's number; often the same as passed to `OSPPTransactionRequestAuthorisation()`, e.g. `"33492944299"`, but not, for example, if the OSP server performs number translation
`ospvDestination`	DNS name or IP address of the destination gateway, for example `"gatewayB.itsp.fr"`
`ospvDestinationDevice`	not needed when the source system will contact the destination device directly; empty string (`""`) in this example
`ospvToken`	Authorization token to present to Gateway B during setup

5) Gateway A sends an INVITE message to Gateway B. This INVITE message uses the SIP Call-ID from the <AuthorisationRequest>, and the message must include all authorization tokens associated with that destination in the server's OSP <AuthorisationResponse>.

If using the OSP Toolkit, Gateway A must also create and initialize a transaction to track its interaction with Gateway B. It does this with a call to the library function OSPPTransactionInitializeAtDevice(). The significant parameters to that function call include the following.

ospvIsSource	Indicates that the local system is a source for the phone call; 1 in the example
ospvSource	DNS name or IP address of Gateway A, for example "gatewayA.carrier.com"
ospvDestination	DNS name or IP address of Gateway B, for example, "gatewayB.itsp.fr"
ospvSourceDevice	not needed when the reporting system is the source device; empty string ("") in this example
ospvDestinationDevice	not needed when the source system will contact the destination device directly; empty string ("") in this example
ospvCallingNumber	calling party's E.164 number; this must be the same as was contained in the original <AuthorisationRequest>
ospvCalledNumber	called party's E.164 number, e.g. "33492944299"
ospvCallId	SIP Call-ID to be sent in INVITE message
ospvToken	authorization token to be presented to Gateway B during setup

Gateway B refuses the setup attempt with a 503 message, perhaps, for example, because no outbound PSTN ports are available.

6) Gateway A tries a second setup attempt, this time by sending an INVITE message to Gateway C.

If using the OSP Toolkit, Gateway A must also re-initialize the transaction to indicate that the peer is now expected to be Gateway C. A call to the library function OSPPTransactionReinitializeAtDevice() accomplishes this.

ospvFailureReason	an identification of the reason for the failure of the previous setup attempt, for example OSPC_FAIL_REMOTE_EXT
ospvIsSource	indicates that the local system is a source for the phone call; 1 in the example
ospvSource	DNS name or IP address of Gateway A, for example "gatewayA.carrier.com"
ospvDestination	DNS name or IP address of Gateway C, e.g. "gatewayC.isp.fr"
ospvSourceDevice	not needed when the reporting system is the source device; empty string ("") in this example
ospvDestinationDevice	not needed when the source system will contact the destination device directly; empty string ("") in this example
ospvCallingNumber	calling party's E.164 number; this must be the same as was contained in the original <AuthorisationRequest>

`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvCallId`	SIP Call-ID to be sent in INVITE message
`ospvToken`	authorization token to be presented to Gateway C during setup

7) Gateway C receives the INVITE message and accepts the call by responding with an 200 message, to which Gateway A replies with an ACK.

If using the TransNexus OSP Toolkit, Gateway C must call the Toolkit function `OSPPTransactionValidateAuthorisation()` to verify the authorization token in the INVITE message. It would call it with the following parameters.

`ospvSource`	DNS name or IP address of Gateway A, e.g. `"[172.16.1.1]"`
`ospvDestination`	DNS name or IP address of Gateway C, e.g., `"gatewayC.isp.fr"`
`ospvSourceDevice`	not needed when the setup message is received directly from a source device; ; empty string (`""`) in this example
`ospvDestinationDevice`	not needed when the setup message is received directly by the local system
`ospvCallingNumber`	calling party's E.164 number; this must be the same as was contained in the original `<AuthorisationRequest>`
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvCallId`	SIP Call-ID received in INVITE message
`ospvToken`	authorization token presented to Gateway C during setup

The `OSPPTransactionValidateAuthorisation()` function will return an indication of whether or not the call is authorized (`ospvAuthorised`), and, if so, how much service is authorized (`ospvTimeLimit`).

Usage Reports

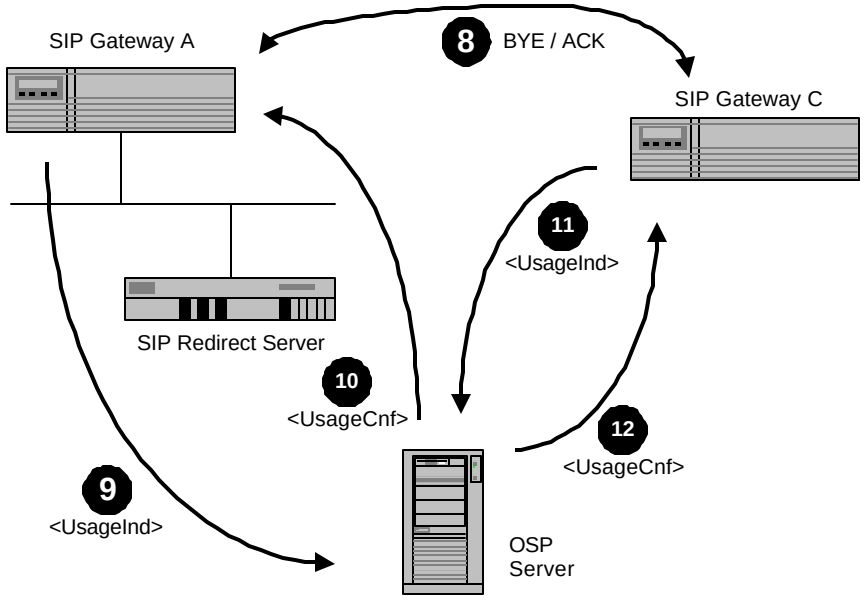
Once the call has ended, both gateways report usage details to an OSP server. As Figure 12 indicates, those reports are conveyed in OSP `<UsageIndication>` messages.

The steps shown in figure 12are straightforward.

8) Gateways A and C clear the call with a SIP BYE message.

9) Gateway A sends a `<UsageIndication>` message to the OSP server. In this example Gateway A's message will include two complete `<UsageIndication>` components, one for the failed attempt and one for the successful call. The sub-elements for each will include the following.

`<UsageIndication>`	usage information for the failed setup attempt
`<Timestamp>`	time of request
`<Role>`	for Gateway A, `source`
`<TransactionId>`	transaction ID assigned by OSP server in authorization response
`<CallId>`	SIP Call-ID used for the call



• Figure 12 Usage Reporting with SIP Redirect Servers.

`<SourceInfo type="e164">`	calling party's E.164 number, e.g. `14048724887`
`<SourceAlternate type="transport">`	DNS name or IP address of Gateway A, for example `gatewayA.carrier.com`
`<DestinationInfo type="e164">`	called party's E.164 number as returned in the authorization response, e.g. `33492944299`
`<DestinationAlternate type="transport">`	DNS name or IP address of Gateway B, for example, `gatewayB.itsp.fr`
`<transnexus.com:FailureReason>`	reason for failure of attempted setup, e.g. `422`
`<UsageIndication>`	usage information for the successful setup attempt
`<Timestamp>`	time of request
`<Role>`	for Gateway A, `source`
`<TransactionId>`	transaction ID assigned by OSP server in authorization response
`<CallId>`	H.323 Call Identifier used for the call
`<SourceInfo type="e164">`	calling party's E.164 number, e.g. `14048724887`
`<SourceAlternate type="transport">`	DNS name or IP address of Gateway A, for example `gatewayA.carrier.com`
`<DestinationInfo type="e164">`	called party's E.164 number as returned in the authorization response, e.g. `33492944299`
`<DestinationAlternate type="transport">`	DNS name or IP address of Gateway C, for example, `gatewayC.isp.fr`
`<UsageDetail>`	usage information for the call
`<Service/>`	empty (for basic service)
`<Amount>`	amount of service used, e.g. `300`
`<Increment>`	increment of service measurement, e.g. `1`
`<Unit>`	unit of service measurement, e.g. `s` for seconds

If using the TransNexus OSP Toolkit, Gateway A can generate that message by calling `OSPPTransactionReportUsage()` with the following significant parameters

`ospvDuration`	length of the call in seconds, e.g. `300`
`ospvLossPacketsSent`	ignored (because of following parameter value)
`ospvLossFractionSent`	`-1` if not reported
`ospvLossPacketsReceived`	ignored (because of following parameter value)
`ospvLossFractionReceived`	`-1` if not reported

10) The OSP server responds with a `<UsageConfirmation>` message. If it has accepted the usage report, that message will contain a successful `<Status>` element (e.g. `<Code>200</Code>`). Gateway A will learn of this response in the return value from the `OSPPTransactionReportUsage()` function.

11) Gateway C also sends a `<UsageIndication>` to the OSP server. Assuming that it uses TransNexus extensions for statistics reporting, that message would include the following elements.

`<Timestamp>`	time of request
`<Role>`	for Gateway C, destination
`<TransactionId>`	transaction ID assigned by OSP server and passed to Gateway C in authorization token
`<CallId>`	SIP Call-ID used for the call
`<SourceInfo type="e164">`	calling party's E.164 number as presented in the INVITE message, e.g. `14048724887`
`<SourceAlternate type="transport">`	DNS name or IP address of Gateway A, for example `[172.16.1.1]`
`<DestinationInfo type="e164">`	called party's E.164 number, e.g. `33492944299`
`<DestinationAlternate type="transport">`	DNS name or IP address of Gateway C, for example, `gatewayC.isp.fr`
`<UsageDetail>`	usage information for the call
`<Service/>`	empty (for basic service)
`<Amount>`	amount of service used, e.g. `300`
`<Increment>`	increment of service measurement, e.g. `1`
`<Unit>`	unit of service measurement, e.g. `s` for seconds
`<transnexus.com:Statistics>`	statistical information for call
`<transnexus.com:LossSent>`	loss information for packets sent by Gateway C
`<transnexus.com:Packets>`	number of packets lost from Gateway C to Gateway A
`<transnexus.com:Fraction>`	fraction (from 0 to 255) of packets lost from C to A
`<transnexus.com:LossReceived>`	loss information for packets sent by Gateway A
`<transnexus.com:Packets>`	number of packets lost from Gateway A to Gateway C
`<transnexus.com:Fraction>`	fraction (from 0 to 255) of packets lost from A to C
`<transnexus.com:OneWayDelay>`	one way delay measured from Gateway A to C
`<transnexus.com:Minimum>`	minimum measured value for delay, in seconds

`<transnexus.com:Mean>`	sample mean of delay measurements, in seconds
`<transnexus.com:Variance>`	sample variance of delay measurements, in squared seconds
`<transnexus.com:Samples>`	number of sample measurements
`<transnexus.com:RoundTripDelay>`	round trip delay between Gateway A and C measured during call
`<transnexus.com:Minimum>`	minimum measured value for delay, in seconds
`<transnexus.com:Mean>`	sample mean of delay measurements, in seconds
`<transnexus.com:Variance>`	sample variance of delay measurements, in squared seconds
`<transnexus.com:Samples>`	number of sample measurements

If using the TransNexus OSP Toolkit, Gateway C may call two Toolkit function while the call is in progress: `OSPPTransactionAccumulateOneWayDelay()` for one-way delay measurements, and `OSPPTransactionAccumulateRoundTripDelay()` for round-trip delay measurements. Note that these functions must be called before `OSPPTransactionReportUsage()`. That call is used to indicate loss statistics as well as the call's duration.

12) The OSP server responds with a `<UsageConfirmation>` message. If it has accepted the usage report, that message will contain a successful `<Status>` element (e.g. `<Code>200</Code>`. Gateway C will learn of this response in the return value from the `OSPPTransactionReportUsage()` function.

Calling Card Services

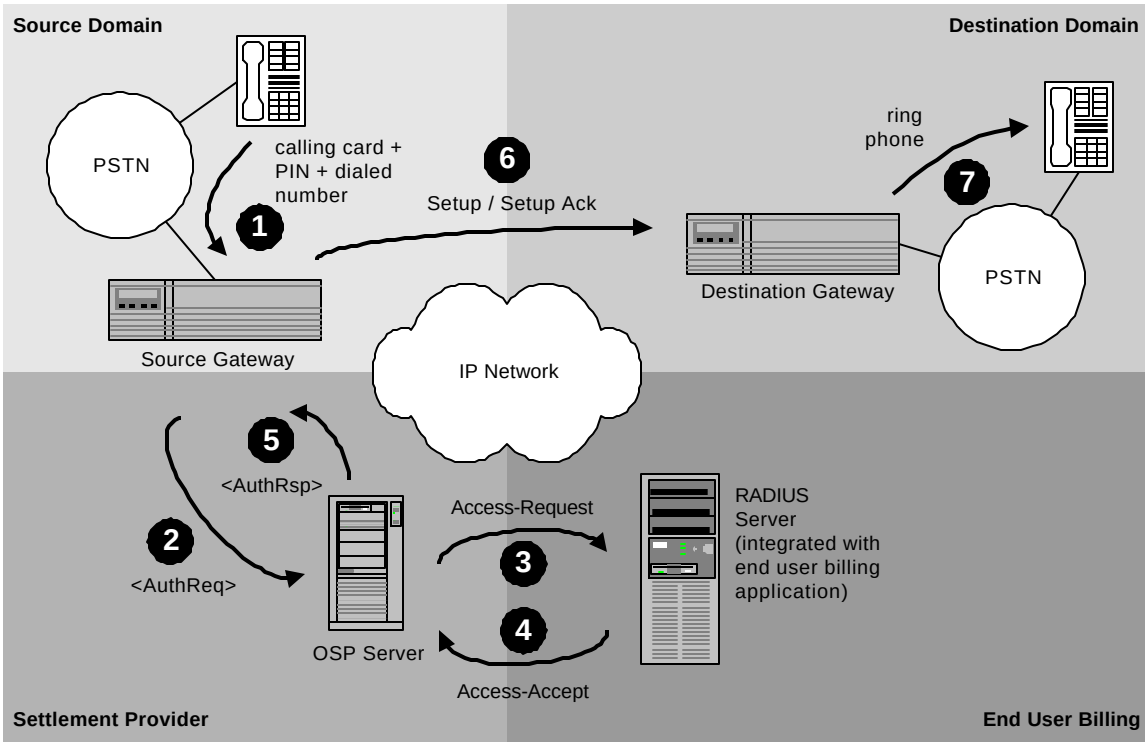
This section details how the Open Settlement Protocol and the TransNexus OSP software development kit may be used to support telephony calling card services. Such services include prepaid and debit cards, as well as calling card support for roaming users. The first subsection presents an overview of the required network and business architecture; the following subsection provides implementation details, and the concluding subsection demonstrates how the Toolkit is used to support that architecture.

Service Architecture

As Figure 13 shows, the most general environment requires support from four different domains: the source and destination domains of the IP telephony gateways, the settlement service provider, and the end user billing domain. User billing may distinct from the source domain in the case, for example, or a roaming user.

The figure also shows the general operational procedure for authorization divided into seven discrete steps.

1) The user accesses a gateway in the source domain. The gateway, perhaps using an IVR application, collects the user's calling card number and personal identification number, in addition to the called number.



- Figure 13 Calling Card Service Architecture.

2) The source gateway forwards this information to the settlement provider in an OSP `<AuthorisationRequest>`.

3) The settlement provider, in addition to authenticating the source gateway, also authenticates the end user. That authentication procedure is outside the scope of OSP, but may, as the example shows, rely on another standard protocol such as RADIUS (RFC 2138).

4) The end user billing application authenticates and authorizes the user.

5) The settlement provider returns an `<AuthorisationResponse>` to the source gateway indicating acceptance of the end user and providing authorization tokens for the destination gateway.

6) The call proceeds normally with a Setup message from the source to the destination gateway.

7) The destination gateway completes the call to the called party.

These steps represent the beginning of a calling card transaction. Additional phases, including refreshing the user's authorization and reporting usage information, can be found in the following section on implementation details.

Implementation Details

This section takes the architecture that Figure 13 illustrates and provides the details necessary to accomplish the required transactions. Those transactions include the initial

authorization of the figure, as well as reauthorizations and usage reports. The section documents both the protocol-level interactions, and the Toolkit interface functions.

Call Routing and Authorization

OSP <AuthorisationRequest> Fields

The OSP `<AuthorisationRequest>` message shown in step 2 contains the following significant elements.

`<Timestamp>`	Time of request
`<CallId>`	H.323 Call Identifier to be used for the call
`<SourceInfo type="e164">`	Calling party's E.164 number if available; otherwise a local E.164 number controlled by the source gateway, e.g. `14048724887`; this number must be passed to the destination gateway(s) in Setup messages
`<SourceAlternate type="transport">`	DNS name or IP address of source gateway, for example `sourcegateway.carrier.com`
`<SourceAlternate type="subscriber">`	The user's calling card and PIN; following conventions established by the Voice over IP Forum, these should be combined into a single character string, with the two components separated by the pound sign (#). For example, the calling card number 12345678, combined with the PIN 4444, should be represented as "12345678#4444".
`<DestinationInfo type="e164">`	Called party's E.164 number, e.g. `33492944299`
`<Service/>`	Empty (for basic service)
`<MaximumDestinations>`	The maximum number of destinations, including alternatives, the source gateway will consider

Toolkit OSPPTransactionRequestAuthorisation() Parameters

If using the TransNexus OSP Toolkit, the source gateway can generate this message by calling `OSPPTransactionRequestAuthorisation()` with the following significant parameters.

`ospvSource`	DNS name or IP address of source gateway, for example `"sourcegateway.carrier.com"`
`ospvSourceDevice`	not needed in peer-to-peer environments; empty string (`""`) in this example
`ospvCallingNumber`	calling party's E.164 number if available; otherwise a local E.164 number controlled by source gateway, e.g. `"14048724887"`; this number must be passed to the destination gateway during Setup
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvUser`	user identification; consisting of the calling card number and PIN, separated by a pound sign (`"12345678#4444"`) in this example)
`ospvNumberOfCallIds`	the number of H.323 Call Identifiers given to the OSP server; if all Setup attempts for this call will use the same Call Identifier value (as in this example), this parameter value is `1`
`ospvCallIds`	an array (containing, in this example, but a single element) of H.323 Call Identifiers; the array structure includes a size field which

	indicates the size of the value
`ospvPreferredDestinations`	optional list of preferred destination gateways; empty string ("") in this example
`ospvNumberOfDestinations`	the maximum number of potential destinations that the source gateway is prepared to consider for the call; for example 3

OSP <AuthorisationResponse> Fields

The OSP server replies with an `<AuthorisationResponse>` message as Figure 13 indicates in step 5. The message indicates candidate destination gateways, in order of priority. In this example (which only shows a single destination gateway, the OSP `<AuthorisationResponse>` contains the following elements.

`<Timestamp>`	time of response
`<Status>`	result of response, e.g. `<Code>200</Code>`
`<TransactionId>`	transaction identifier assigned by settlement provider
`<Destination>`	first destination gateway to try for call
`<DestinationSignalAddress type="transport">`	DNS name or IP address of destination gateway, for example `destgateway.itsp.fr`
`<Token>`	authorization token to be passed to destination gateway
`<ValidAfter>`	time after which token for destination gateway is valid
`<ValidUntil>`	time until which token for destination gateway is valid
`<UsageDetail>`	how much service is authorized with destination gateway
`<Service/>`	empty (for basic service)
`<Amount>`	amount of authorized service, e.g. `3600`
`<Increment>`	increment of service measurement, e.g. `1`
`<Unit>`	unit of service measurement, e.g. `s` for seconds
`<CallId>`	H.323 Call Identifier to be used for the call to destination gateway

Toolkit OSPPTransactionGetFirstDestination() Parameters

If using the TransNexus OSP Toolkit, the source gateway will be informed of the receipt of a valid OSP `<AuthorisationResponse>` by the return value of the call to the function `OSPPTransactionRequestAuthorisation()`. It can then retrieve the information it needs to identify and access the destination gateway by calling the Toolkit function `OSPPTransactionGetFirstDestination()`. That function will return the following information.

`ospvValidAfter`	time after which authorization token for destination gateway is valid
`ospvValidUntil`	time until which authorization token for destination gateway is valid
`ospvTimeLimit`	amount of service authorized with destination gateway, e.g. `3600` (seconds)
`ospvCallId`	H.323 Call Identifier to use in Setup message to destination gateway

`ospvCalledNumber`	number to present to the destination gateway as the called party's number; often this value is the same as the value passed to the Toolkit function `OSPPTransactionRequestAuthorisation()`, e.g. `"33492944299"`, but not, for example, if the OSP server performs number translation
`ospvDestination`	DNS name or IP address of the destination gateway, for example `"destgateway.itsp.fr"`
`ospvDestinationDevice`	not needed in peer-to-peer environments; empty string (`""`) in this example
`ospvToken`	authorization token to present to Gateway B during setup

At this point the source gateway can proceed to establish the call.

Toolkit OSPPTransactionValidateAuthorisation() Parameters

When the destination gateway receives the Setup request, it can validate that request (and potentially determine whether to accept it or not) with a call to the Toolkit function `OSPPTransactionValidateAuthorisation()`. That function validates the token in the Setup message. The destination gateway should call it with the following parameters.

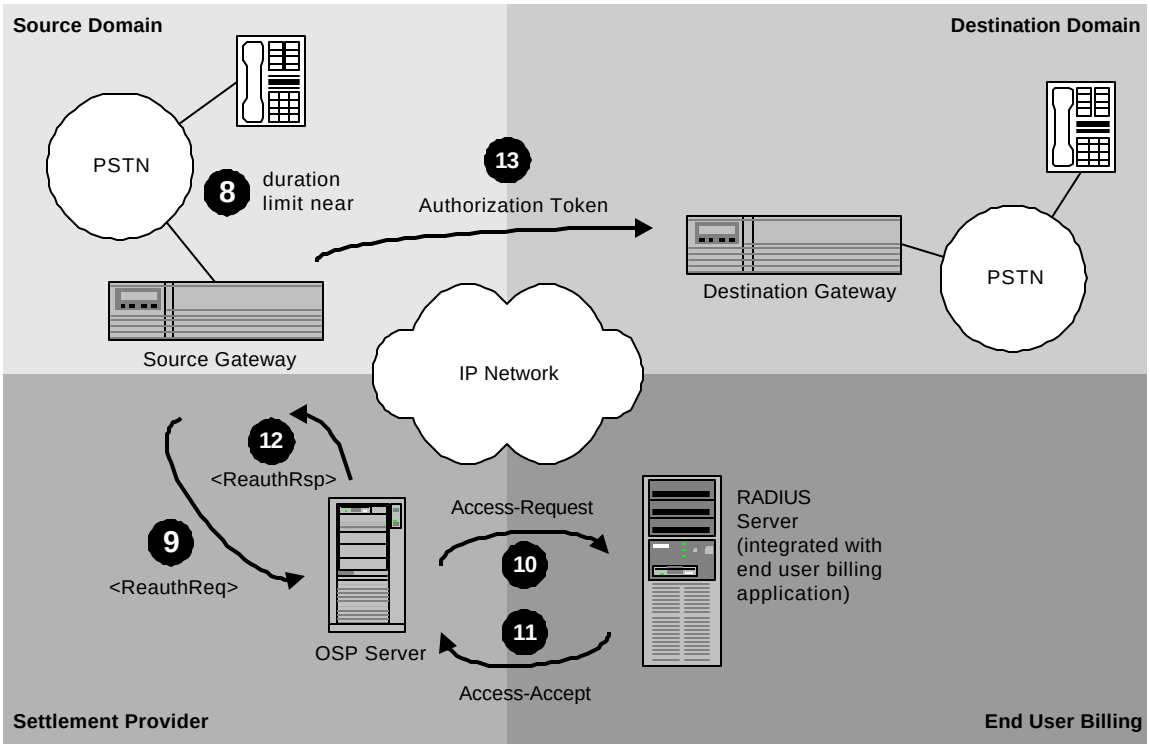
`ospvSource`	DNS name or IP address of the source gateway, for example `"[172.16.1.1]"`
`ospvDestination`	DNS name or IP address of the destination gateway for example `"destgateway.itsp.fr"`
`ospvSourceDevice`	not needed in peer-to-peer environments; empty string (`""`) in this example
`ospvDestinationDevice`	not needed in peer-to-peer environments; empty string (`""`) in this example
`ospvCallingNumber`	calling party's E.164 number, e.g. `"14048724887"`
`ospvCalledNumber`	called party's E.164 number, e.g. `"33492944299"`
`ospvCallId`	H.323 Call Identifier received in Setup message
`ospvToken`	authorization token presented to destination gateway during setup

Information returned from this function call will include the following:

`ospvAuthorised`	indication of whether or not the reauthorization was successful
`ospvTimeLimit`	total number of seconds for which call is now authorized (including any prior usage)

Reauthorization

In many calling card services, it may be necessary to refresh the authorization of a call that is already in progress. For example, some debit card applications will only authorize a limited amount of service at any given time; this can minimize the risk of fraudulent, simultaneous use of the debit card by multiple users. In such scenarios, and when the users wish to continue their conversation past the limited amount of time initially authorized, it will be necessary for the supporting devices to request additional authorization for the call. Figure 14 shows the message flow for this process. The six steps



• Figure 14 Reauthorization of In-Progress Call.

in the figure begin when the originating gateway recognizes that the currently authorized service limit is approaching.

8) Source gateway recognizes that the duration currently authorized for the call is nearing its limit.

9) Source gateway sends an OSP `<ReauthorisationRequest>` to the OSP server.

10) The OSP server uses some other means (such as RADIUS, in the example) to authorize additional service for the user.

11) The OSP server confirms that additional service is authorized.

12) The OSP server returns a `<ReauthorisationResponse>` to the source gateway, granting the additional authorized service. The response tells the source gateway the new authorization limits explicitly, and it includes an updated authorization token.

13) The source gateway passes the new authorization token to the destination gateway. The exact method of transfer depends on the gateway implementations and the particular call signaling protocol; as an example, the source gateway may include the token in an H.323 Facility message.

OSP <ReauthorisationRequest> Fields

The OSP `<ReauthorisationRequest>` message shown in step 9 contains the following significant elements.

<Timestamp>	Time of request
<Role>	for the source gateway, source
<CallId>	H.323 Call Identifier used for the call
<SourceInfo type="e164">	Calling party's E.164 number if available; otherwise a local E.164 number controlled by the source gateway, e.g. 14048724887; this number must be passed to the destination gateway(s) in Setup messages
<SourceAlternate type="transport">	DNS name or IP address of source gateway, for example sourcegateway.carrier.com
<SourceAlternate type="subscriber">	The user's calling card and PIN; following conventions established by the Voice over IP Forum, these should be combined into a single character string, with the two components separated by the pound sign (#). For example, the calling card number 12345678, combined with the PIN 4444, should be represented as "12345678#4444".
<DestinationInfo type="e164">	Called party's E.164 number, e.g. 33492944299
<DestinationAlternate type="transport">	DNS name or IP address of destination gateway, for example, destgateway.itsp.fr
<TransactionId>	transaction identifier assigned by settlement provider
<UsageDetail>	usage information for the call so far
<Service/>	empty (for basic service)
<Amount>	amount of service used so far, e.g. 300
<Increment>	increment of service measurement, e.g. 1
<Unit>	unit of service measurement, e.g. s for seconds
<Token>	authorization token to be passed to destination gateway

Toolkit OSPPTransactionRequestReauthorisation() Parameters

If using the TransNexus OSP Toolkit, the source gateway can generate this message by calling OSPPTransactionRequestAuthorisation() with the following significant parameters.

ospvDuration	length of the call in seconds, e.g. 300

Information returned from this function call will include the following:

ospvToken	updated authorization token to be presented to destination gateway
ospvAuthorised	indication of whether or not the reauthorization was successful
ospvTimeLimit	total number of seconds for which call is now authorized (including any prior usage)

OSP <ReauthorisationResponse> Fields

The OSP server returns that information within an <ReauthorisationResponse> message, as Figure 14 indicates in step 12. The message refreshes the authorization

information for the call. In this example, the OSP <AuthorisationResponse> contains the following elements.

<Timestamp>	time of response
<Status>	result of response, e.g. <Code>200</Code>
<TransactionId>	transaction identifier assigned by settlement provider
<Destination>	destination gateway to try for call
<DestinationSignalAddress type="transport">	DNS name or IP address of destination gateway, for example destgateway.itsp.fr
<Token>	updated authorization token to be passed to destination gateway
<ValidAfter>	time after which token for destination gateway is valid
<ValidUntil>	time until which token for destination gateway is valid
<UsageDetail>	how much (cumulative) service is authorized with destination gateway
<Service/>	empty (for basic service)
<Amount>	(cumulative) amount of authorized service, e.g. 3600
<Increment>	increment of service measurement, e.g. 1
<Unit>	unit of service measurement, e.g. s for seconds
<CallId>	H.323 Call Identifier to be used for the call to destination gateway

Toolkit OSPPTransactionValidateAuthorisation() Parameters

When the destination gateway receives the updated authorization token request, it can validate it with a call to OSPPTransactionValidateReauthorisation(). The destination gateway should call it with the following parameters.

ospvToken	authorization token presented to destination gateway during setup

Information returned from this function call will include the following:

ospvAuthorised	indication of whether or not the reauthorization was successful
ospvTimeLimit	total number of seconds for which call is now authorized (including any prior usage)

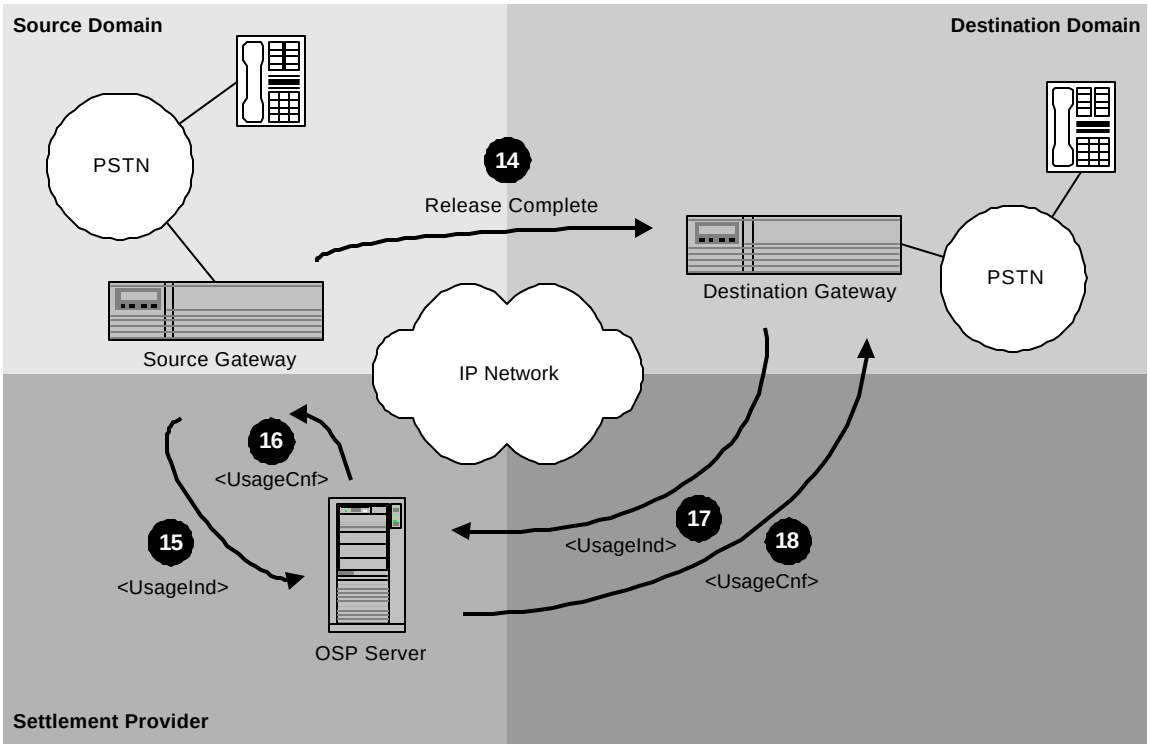
Usage Reports

Once the call has ended, both gateways report usage details to an OSP server. As Figure 15 indicates, those reports are conveyed in OSP <UsageIndication> messages.

The steps shown in the figure are straightforward.

14) The gateways clear the call by exchanging an H.225.0 Release Complete message.

15) The source gateway sends a <UsageIndication> message to the OSP server, reporting its usage details for the call.



• Figure 15 Usage Reporting.

16) The OSP server acknowledges receipt with a `<UsageConfirmation>` message.

17) The destination gateway also reports its usage details with a `<UsageIndication>` message.

18) The server acknowledges this message as well with a `<UsageConfirmation>`.

OSP <UsageIndication> Fields

The `<UsageIndication>` messages from both gateways will be substantially the same. As an example, here are the significant fields of the source gateway's message.

`<Timestamp>`	time of request
`<Role>`	for source gateway, `source`
`<TransactionId>`	transaction ID assigned by OSP server in authorization response
`<CallId>`	H.323 Call Identifier used for the call
`<SourceInfo type="e164">`	calling party's E.164 number as returned in the authorization response, e.g. `14048724887`
`<SourceAlternate type="transport">`	DNS name or IP address of source gateway, for example `sourcegateway.carrier.com`
`<DestinationInfo type="e164">`	called party's E.164 number, e.g. `33492944299`
`<DestinationAlternate type="transport">`	DNS name or IP address of destination gateway, for example, `destgateway.isp.fr`
`<UsageDetail>`	usage information for the call

`<Service/>`	empty (for basic service)
`<Amount>`	amount of service used, e.g. `600`
`<Increment>`	increment of service measurement, e.g. `1`
`<Unit>`	unit of service measurement, e.g. `s` for seconds

Toolkit OSPPTransactionReportUsage() Parameters

If using the TransNexus OSP Toolkit, both gateways can generate the appropriate OSP message by calling `OSPPTransactionReportUsage()` Here are the significant parameters that the source gateway would supply.

`ospvDuration`	total length of the call in seconds, e.g. `600`
`ospvLossPacketsSent`	value ignored (because of following parameter value)
`ospvLossFractionSent`	`-1` if not reported
`ospvLossPacketsReceived`	value ignored (because of following parameter value)
`ospvLossFractionReceived`	`-1` if not reported

OSP <UsageConfirmation> Fields

The OSP server responds with a `<UsageConfirmation>` message. If it has accepted the usage report, that message will contain a successful `<Status>` element (e.g. `<Code>200</Code>`. Each gateway will learn of this response in the return value from the `OSPPTransactionReportUsage()` function.