[MS-OCSPROT]:

Lync and Lync Server Protocols Overview

This document provides a system overview for the protocols in the Communications Server system. It is intended for use in conjunction with the Microsoft protocol technical specifications, publicly available standard specifications, network programming art, and Microsoft Windows distributed systems concepts. It assumes that the reader is either familiar with the aforementioned material or has immediate access to it.

A protocol system document does not require the use of Microsoft programming tools or programming environments in order to implement the protocols in the system. Developers who have access to Microsoft programming tools and environments are free to take advantage of them.

specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Abstract

Communications Server is a client-server product that is based on the Session Initiation Protocol (SIP) to facilitate real-time communications between users. Protocol clients, such as Office Communicator, are used to sign in to Communications Server. Users can initiate calls to one or more users who are also signed in to Communications Server by using different protocol clients such as IM, audio, video, VoIP, and applications-sharing. These clients are enabled via other protocols. Communications Server aggregates the user's presence from all of the user's protocol clients and publishes that presence information for other users authorized to view it.

This document describes the intended functionality of the Communications Server system and how the protocols in this system interact. It provides examples of some of the common user scenarios. It does not restate the processing rules and other details that are specific for each protocol. These details are described in the protocol specifications for each of the protocols and data structures that make up this system.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	Major	Initial Availability
4/25/2008	0.2	Major	Revised and edited the technical content
6/27/2008	1.0	Major	Revised and edited the technical content
8/15/2008	1.01	Major	Revised and edited the technical content
12/12/2008	2.0	Major	Revised and edited the technical content
2/13/2009	2.01	Major	Revised and edited the technical content
3/18/2009	2.02	Editorial	Revised and edited the technical content
7/13/2009	2.03	Major	Changes made for template compliance
8/28/2009	2.04	Editorial	Revised and edited the technical content
11/6/2009	2.05	Editorial	Revised and edited the technical content
2/19/2010	2.06	Editorial	Revised and edited the technical content
3/31/2010	2.07	Major	Updated and revised the technical content
4/30/2010	2.08	Editorial	Revised and edited the technical content
6/7/2010	2.09	Editorial	Revised and edited the technical content
6/29/2010	2.10	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.10	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	3.0	Major	Significantly changed the technical content.
11/15/2010	3.0	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	3.0	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	4.0	Major	Significantly changed the technical content.
4/11/2012	4.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	4.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	4.1	Minor	Clarified the meaning of the technical content.
2/11/2013	4.2	Minor	Clarified the meaning of the technical content.
7/30/2013	4.3	Minor	Clarified the meaning of the technical content.

Date	Revision History	Revision Class	Comments
11/18/2013	4.3	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	4.3	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	4.3	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	4.4	Minor	Clarified the meaning of the technical content.
10/30/2014	4.4	None	No changes to the meaning, language, or formatting of the technical content.
3/30/2015	4.4	None	No changes to the meaning, language, or formatting of the technical content.
9/4/2015	5.0	Major	Significantly changed the technical content.
7/15/2016	5.1	Minor	Clarified the meaning of the technical content.
9/14/2016	5.1	None	No changes to the meaning, language, or formatting of the technical content.
9/29/2016	5.1	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The protocols in the Microsoft® Office Communications Server Protocols system support instant messaging (IM), presence notification, Web conferencing, **Voice over IP (VoIP)** telephony, and audio/video (A/V) conferencing functionality. The processing for the Communications Server components is handled by a set of specialized server roles that run as Windows® services. These roles form dependent and complimentary building blocks to create a communications infrastructure that is geared to meet specific types of user scenarios. The Windows services that represent these server roles run on Windows Server 2003 operating system or Windows Server 2008 operating system with Service Pack 2 (SP2). Many of these server roles are installed together by default to simplify the installation and configuration of Communications Server, while others can be collocated on the same physical server or installed on separate computers that are running Windows Server 2003 or Windows Server 2008.

Communications Server is available in two editions: Standard Edition for organizations with 5000 or fewer users and Enterprise Edition for organizations with more than 5000 users. The two editions are functionally equivalent, but their configuration is different to be able to scale up. A Communications Server infrastructure can include protocol servers for both editions installed and working together.

1.1 Glossary

This document uses the following terms:

200 OK: A response to indicate that the request has succeeded.

acknowledgment (ACK): A signal passed between communicating processes or computers to signify successful receipt of a transmission as part of a communications protocol.

Active Directory: A general-purpose network **directory service**. **Active Directory** also refers to the Windows implementation of a **directory service**. **Active Directory** stores information about a variety of objects in the network. Importantly, user accounts, computer accounts, groups, and all related credential information used by the Windows implementation of **Kerberos** are stored in **Active Directory**. **Active Directory** is either deployed as Active Directory Domain Services (AD DS) or Active Directory Lightweight Directory Services (AD LDS). [MS-ADTS] describes both forms. For more information, see [MS-AUTHSOD] section 1.1.1.5.2, Lightweight Directory Access Protocol (LDAP) versions 2 and 3, **Kerberos**, and **DNS**.

Address Book Server (ABS): A component that produces address book files on a daily basis.

agent: A device that is connected to a computer network. Also referred to as an endpoint.

Audio/Video Edge Server (A/V Edge Server): A protocol server that implements the Traversal Using Relay NAT (TURN) Extensions Protocol, as described in [MS-TURN]. The protocol server provides connectivity to a protocol client that is behind a network entity, if the network entity provides network address translation (NAT).

authentication: The act of proving an identity to a server while providing key material that binds the identity to subsequent communications.

bandwidth management endpoint: A protocol client that communicates with a protocol server to discover and enforce applicable bandwidth policies, and to track and send updates about bandwidth utilization to that server.

certificate: A certificate is a collection of attributes (1) and extensions that can be stored persistently. The set of attributes in a certificate can vary depending on the intended usage of the certificate. A certificate securely binds a public key to the entity that holds the corresponding private key. A certificate is commonly used for **authentication** and secure exchange of information on open networks, such as the Internet, extranets, and intranets. Certificates are

digitally signed by the issuing **certification authority (CA)** and can be issued for a user, a computer, or a service. The most widely accepted format for certificates is defined by the ITU-T X.509 version 3 international standards. For more information about attributes and extensions, see [RFC3280] and [X509] sections 7 and 8.

certification authority (CA): A third party that issues public key **certificates**. Certificates serve to bind public keys to a user identity. Each user and certification authority (CA) can decide whether to trust another user or CA for a specific purpose, and whether this trust should be transitive. For more information, see [RFC3280].

contact: A presence entity (presentity) whose presence information can be tracked.

directory service (DS): A service that stores and organizes information about a computer network's users and network shares, and that allows network administrators to manage users' access to the shares. See also **Active Directory**.

Domain Name System (DNS): A hierarchical, distributed database that contains mappings of domain names (1) to various types of data, such as IP addresses. DNS enables the location of computers and services by user-friendly names, and it also enables the discovery of other information stored in the database.

dual-tone multi-frequency (DTMF): In telephony systems, a signaling system in which each digit is associated with two specific frequencies. This system typically is associated with touch-tone keypads for telephones.

encryption: In cryptography, the process of obscuring information to make it unreadable without special knowledge.

endpoint: A device that is connected to a computer network.

Extensible Message and Presence Protocol (XMPP): An application profile of **XML** that enables the near-real-time exchange of structured yet extensible data between any two or more network entities.

fully qualified domain name (FQDN): In **Active Directory**, a **fully qualified domain name (FQDN)** that identifies a domain.

Globally Routable User Agent URI (GRUU): A URI that identifies a user agent and is globally routable. A URI possesses a GRUU property if it is useable by any user agent client (UAC) that is connected to the Internet, routable to a specific user agent instance, and long-lived.

in-band provisioning: A process in which a protocol client obtains configuration information from a protocol server.

Interactive Connectivity Establishment (ICE): A methodology that was established by the Internet Engineering Task Force (IETF) to facilitate the traversal of network address translation (NAT) by media.

Internet Information Services (IIS): The services provided in Windows implementation that support web server functionality. **IIS** consists of a collection of standard Internet protocol servers such as HTTP and FTP in addition to common infrastructures that are used by other Microsoft Internet protocol servers such as SMTP, NNTP, and so on. **IIS** has been part of the Windows operating system in some versions and a separate install package in others. **IIS** version 5.0 shipped as part of Windows 2000 operating system, **IIS** version 5.1 as part of Windows XP operating system, **IIS** version 6.0 as part of Windows Server 2003 operating system, and **IIS** version 7.0 as part of Windows Vista operating system and Windows Server 2008 operating system.

INVITE: A **Session Initiation Protocol (SIP)** method that is used to invite a user or a service to participate in a session.

Kerberos: An **authentication** system that enables two parties to exchange private information across an otherwise open network by assigning a unique key (called a ticket) to each user that logs on to the network and then embedding these tickets into messages sent by the users. For more information, see [MS-KILE].

network address translation (NAT): The process of converting between IP addresses used within an intranet, or other private network, and Internet IP addresses.

NT LAN Manager (NTLM) Authentication Protocol: A protocol using a challenge-response mechanism for **authentication** in which clients are able to verify their identities without sending a password to the server. It consists of three messages, commonly referred to as Type 1 (negotiation), Type 2 (challenge) and Type 3 (authentication). For more information, see [MS-NLMP].

private branch exchange (PBX): A server-based telephony solution that services a specific organization or office.

public switched telephone network (PSTN): Public switched telephone network is the voice-oriented public switched telephone network. It is circuit-switched, as opposed to the packet-switched networks.

Quality of Experience (QoE): A subjective measure of a user's experiences with a media service.

Real-Time Transport Control Protocol (RTCP): A network transport protocol that enables monitoring of Real-Time Transport Protocol (RTP) data delivery and provides minimal control and identification functionality, as described in [RFC3550].

Real-Time Transport Protocol (RTP): A network transport protocol that provides end-to-end transport functions that are suitable for applications that transmit real-time data, such as audio and video, as described in [RFC3550].

Secure Sockets Layer (SSL): A security protocol that supports confidentiality and integrity of messages in client and server applications that communicate over open networks. SSL uses two keys to encrypt data-a public key known to everyone and a private or secret key known only to the recipient of the message. SSL supports server and, optionally, client **authentication** using X.509 **certificates**. For more information, see [X509]. The SSL protocol is precursor to Transport Layer Security (TLS). The TLS version 1.0 specification is based on SSL version 3.0 [SSL3].

server: A replicating machine that sends replicated files to a partner (client). The term "server" refers to the machine acting in response to requests from partners that want to receive replicated files.

Session Initiation Protocol (SIP): An application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. **SIP** is defined in [RFC3261].

Simple Traversal of UDP through NAT (STUN): A protocol that enables applications to discover the presence of and types of network address translations (NATs) and firewalls that exist between those applications and the Internet.

Traversal Using Relay NAT (TURN): A protocol that is used to allocate a public IP address and port on a globally reachable server for the purpose of relaying media from one **endpoint** to another **endpoint**.

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

Voice over IP (VoIP): The use of the Internet Protocol (IP) for transmitting voice communications. VoIP delivers digitized audio in packet form and can be used to transmit over intranets, extranets, and the Internet.

XML: The Extensible Markup Language, as described in [XML1.0].

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

[MS-ABS] Microsoft Corporation, "Address Book File Structure".

[MS-AVEDGEA] Microsoft Corporation, "Audio Video Edge Authentication Protocol".

[MS-CONFAS] Microsoft Corporation, "Centralized Conference Control Protocol: Application Sharing Extensions".

[MS-CONFAV] Microsoft Corporation, "Centralized Conference Control Protocol: Audio-Video Extensions".

[MS-CONFBAS] Microsoft Corporation, "Centralized Conference Control Protocol: Basic Architecture and Signaling".

[MS-CONFIM] Microsoft Corporation, "Centralized Conference Control Protocol: Instant Messaging Extensions".

[MS-CONFPRO] Microsoft Corporation, "Centralized Conference Control Protocol: Provisioning".

[MS-CONMGMT] Microsoft Corporation, "Connection Management Protocol".

[MS-CVWREST] Microsoft Corporation, "Unified Communications Call Via Work Protocol".

[MS-DLX] Microsoft Corporation, "Distribution List Expansion Protocol".

[MS-DTMF] Microsoft Corporation, "RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals Extensions".

[MS-E911WS] Microsoft Corporation, "Web Service for E911 Support Protocol".

[MS-ECREST] Microsoft Corporation, "Unified Communications Event Channel Protocol".

[MS-EUMR] Microsoft Corporation, "Routing to Exchange Unified Messaging Extensions".

[MS-EUMSDP] Microsoft Corporation, "Exchange Unified Messaging Session Description Protocol Extension".

[MS-H264PF] Microsoft Corporation, "RTP Payload Format for H.264 Video Streams Extensions".

[MS-ICE2BWM] Microsoft Corporation, "Interactive Connectivity Establishment (ICE) 2.0 Bandwidth Management Extensions".

[MS-ICE2] Microsoft Corporation, "Interactive Connectivity Establishment (ICE) Extensions 2.0".

[MS-ICE] Microsoft Corporation, "Interactive Connectivity Establishment (ICE) Extensions".

[MS-MQSD] Microsoft Corporation, "Message Queuing (MSMQ): Directory Service Discovery Protocol".

[MS-NLMP] Microsoft Corporation, "NT LAN Manager (NTLM) Authentication Protocol".

[MS-OCAUTHWS] Microsoft Corporation, "OC Authentication Web Service Protocol".

[MS-OCDISCWS] Microsoft Corporation, "Lync Autodiscover Web Service Protocol".

[MS-OCER] Microsoft Corporation, "Client Error Reporting Protocol".

[MS-OCEXUM] Microsoft Corporation, "Call Control for Exchange Unified Messaging Protocol Extensions".

[MS-OCGCWEB] Microsoft Corporation, "Persistent Chat Web Protocol".

[MS-OCPSTN] Microsoft Corporation, "Session Initiation Protocol (SIP) for PSTN Calls Extensions".

[MS-OCSMP] Microsoft Corporation, "Microsoft Online Conference Scheduling and Management Protocol".

[MS-PRES] Microsoft Corporation, "Presence Protocol".

[MS-PSOM] Microsoft Corporation, "PSOM Shared Object Messaging Protocol".

[MS-QoE] Microsoft Corporation, "Quality of Experience Monitoring Server Protocol".

[MS-RDPBCGR] Microsoft Corporation, "Remote Desktop Protocol: Basic Connectivity and Graphics Remoting".

[MS-RGSWS] Microsoft Corporation, "Response Group Service Web Service Protocol".

[MS-RTASPF] Microsoft Corporation, "RTP for Application Sharing Payload Format Extensions".

[MS-RTPRADEX] Microsoft Corporation, "RTP Payload for Redundant Audio Data Extensions".

[MS-RTP] Microsoft Corporation, "Real-time Transport Protocol (RTP) Extensions".

[MS-RTVPF] Microsoft Corporation, "RTP Payload Format for RT Video Streams Extensions".

[MS-SDPEXT] Microsoft Corporation, "Session Description Protocol (SDP) Version 2.0 Extensions".

[MS-SIPAE] Microsoft Corporation, "Session Initiation Protocol (SIP) Authentication Extensions".

[MS-SIPAPP] Microsoft Corporation, "Session Initiation Protocol (SIP) Application Protocol".

[MS-SIPCOMP] Microsoft Corporation, "Session Initiation Protocol (SIP) Compression Protocol".

[MS-SIPREGE] Microsoft Corporation, "Session Initiation Protocol (SIP) Registration Extensions".

[MS-SIPRE] Microsoft Corporation, "Session Initiation Protocol (SIP) Routing Extensions".

[MS-SRTP] Microsoft Corporation, "Secure Real-time Transport Protocol (SRTP) Extensions".

[MS-SSRTP] Microsoft Corporation, "Scale Secure Real-time Transport Protocol (SSRTP) Extensions".

[MS-TURNBWM] Microsoft Corporation, "Traversal using Relay NAT (TURN) Bandwidth Management Extensions".

[MS-TURN] Microsoft Corporation, "Traversal Using Relay NAT (TURN) Extensions".

[MS-WOPI] Microsoft Corporation, "Web Application Open Platform Interface Protocol".

[MS-XCCOSIP] Microsoft Corporation, "Extensible Chat Control Over Session Initiation Protocol (SIP)".

[MS-XMLMC] Microsoft Corporation, "XML Schema for Media Control Extensions".

[RFC2118] Pall, G., "Microsoft Point-To-Point Compression (MPPC) Protocol", RFC 2118, March 1997, http://www.ietf.org/rfc/rfc2118.txt

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E., "SIP: Session Initiation Protocol", RFC 3261, June 2002, http://www.ietf.org/rfc/rfc3261.txt

[RFC3265] Roach, A. B., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002, http://www.ietf.org/rfc/rfc3265.txt

[RFC3325] Jennings, C., Peterson, J., and Watson, M., "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002, http://www.rfc-editor.org/rfc/rfc3325.txt

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V., "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003, http://www.ietf.org/rfc/rfc3550.txt

[RFC3551] Schulzrinne, H., and Casner, S., "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003, http://www.ietf.org/rfc/rfc3551.txt

[RFC3892] Sparks, R., "The Session Initiation Protocol (SIP) Referred-By Mechanism", RFC 3892, September 2004, http://www.rfc-editor.org/rfc/rfc3892.txt

[RFC6120] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Core", March 2011, http://www.rfc-editor.org/rfc/rfc6120.txt

[RFC6121] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", March 2011, http://www.rfc-editor.org/rfc/rfc6121.txt

2 Functional Architecture

The following section describes the functional architecture of the Office Communications Server Protocols system.

2.1 Overview

Communications Server is used to provide unified communications for real-time multimedia communications and collaboration. Communications Server is an enterprise software server solution that provides four different workloads in an integrated and unified user experience: instant messaging (IM) and presence, applications sharing, audio/video and Web conferencing, and enterprise voice. Voice over IP (VoIP) is part of enterprise voice, but enterprise voice also includes voice-specific server applications. Each workload uses different protocols and performs different functions.

Communications Server operates under the common client-server architecture, where a protocol client connects to Communications Server using the **Session Initiation Protocol (SIP)** . The protocol client initiates communications with other protocol clients that Communications Server establishes by using signaling and control channel protocols. Once the communication channel is established between two or more parties, the communication workload is transferred by using the media protocols.

Behind the simplicity of the client-server architecture lies a vast set of functionality that spans from basic storage to accessing, updating, and synchronizing user information configured in **Active Directory** (SIP URI, phone number, home server, and so on), presence information, **in-band provisioning** settings, and address book data.

The protocol clients that interoperate with the protocol server perform tasks such as subscribing to presence information of remote users (**contact**), updating the local user's presence, initiating and accepting communication workloads (instant messaging, Web conferencing, application-sharing, audio/video, and voice calls) with other protocol clients, and requesting ancillary supporting services (such as address book downloads and distribution list expansion).

Systems that interface with Communications Server include both internal and external protocol clients, Communications Server servers from other organizations connected over a federated link, public Instant Messaging (IM) providers using Public IM Connectivity (PIC), SIP/**public switched telephone network (PSTN)** as well as Remote Call Control gateways, Exchange Unified Messaging servers, and **server** applications built using Communications Server's Unified Communications Managed API (UCMA 2.0). Dependencies on these systems are listed in more detail in section 2.3.1.

Below are a high level architectural reference diagram(s) for the communications server and the component and protocol interactions for various workloads.

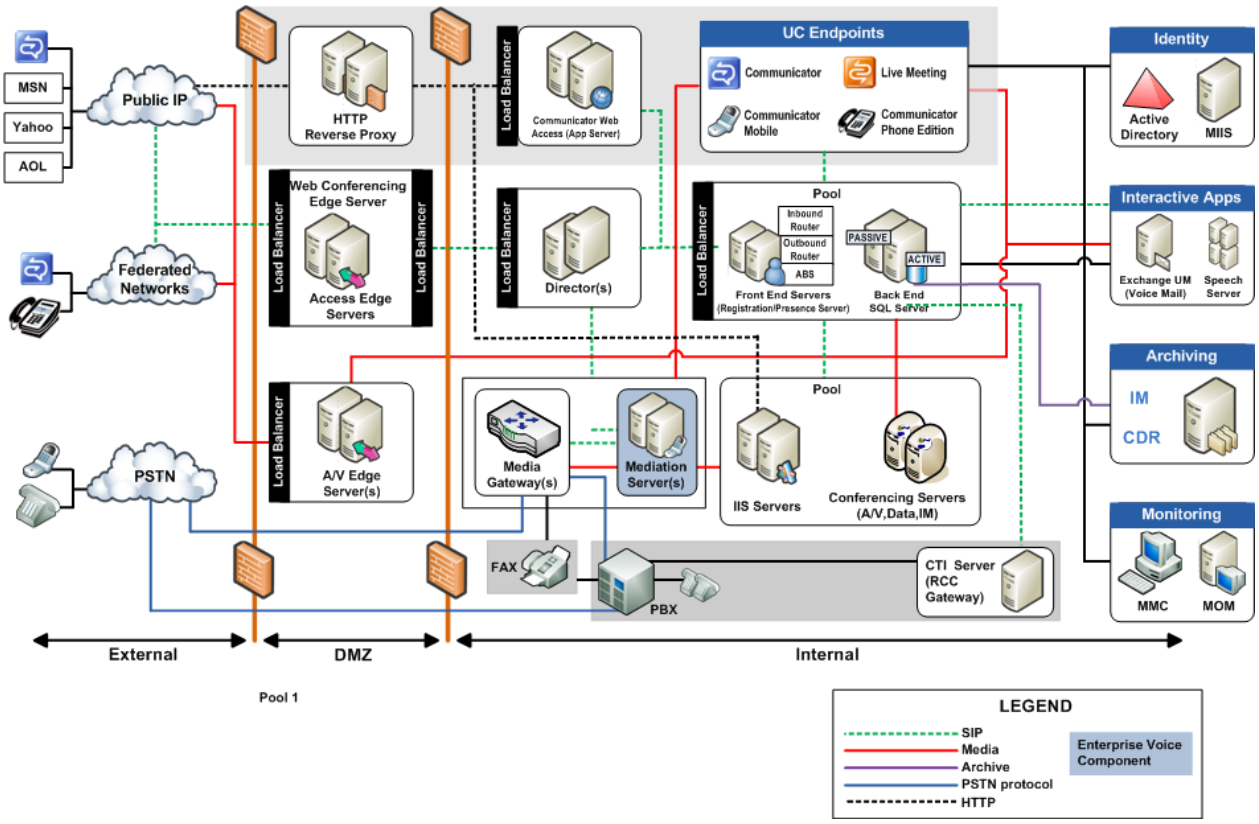

Figure 1: Communications server architectural reference

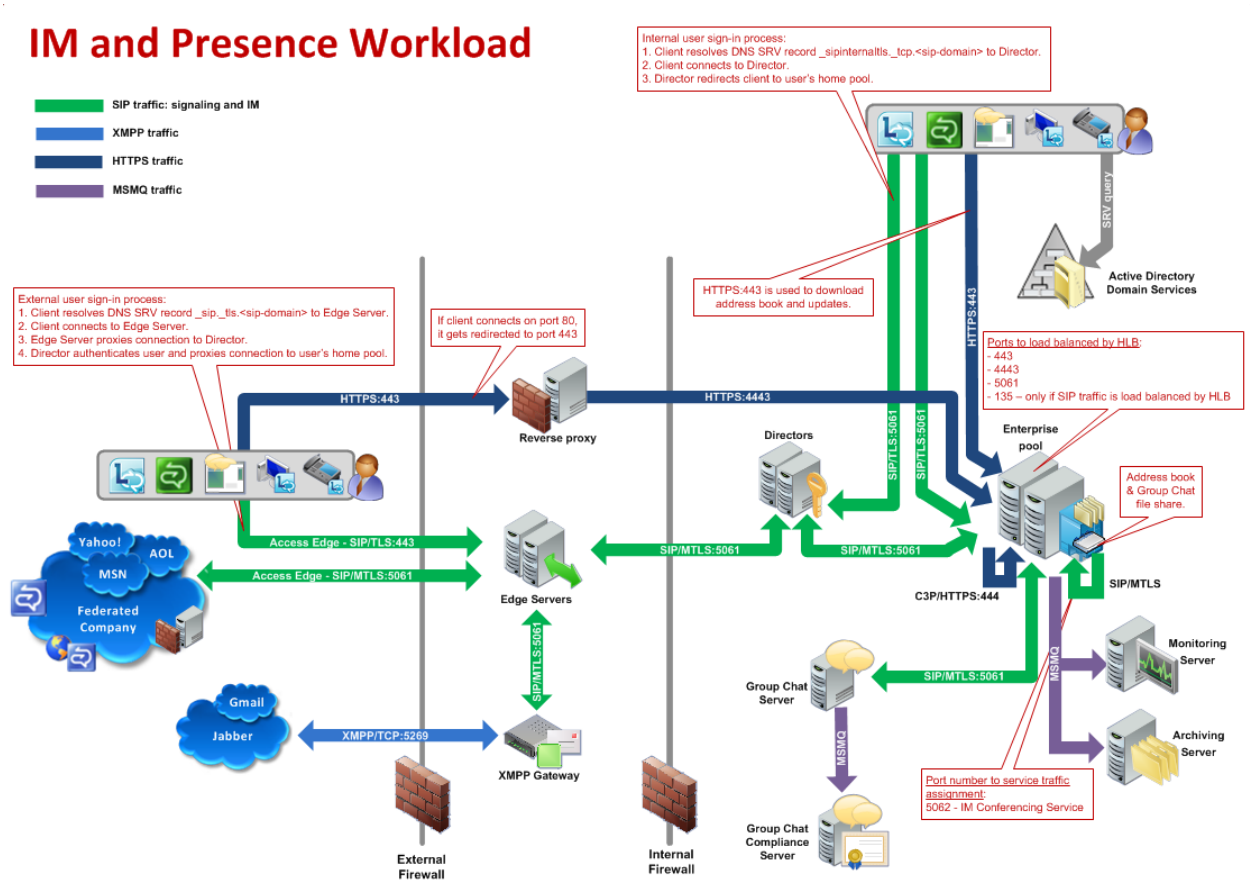

Figure 2: IM and presence workload

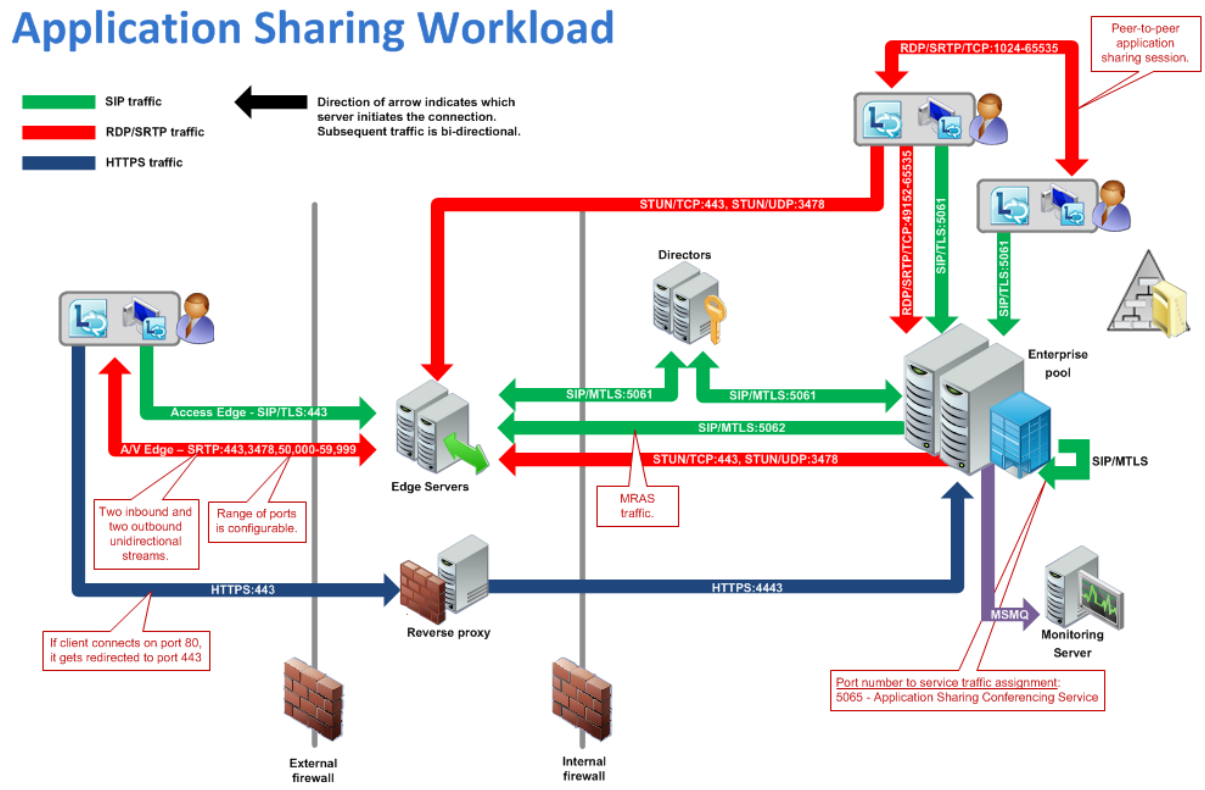

Figure 3: Application sharing workload

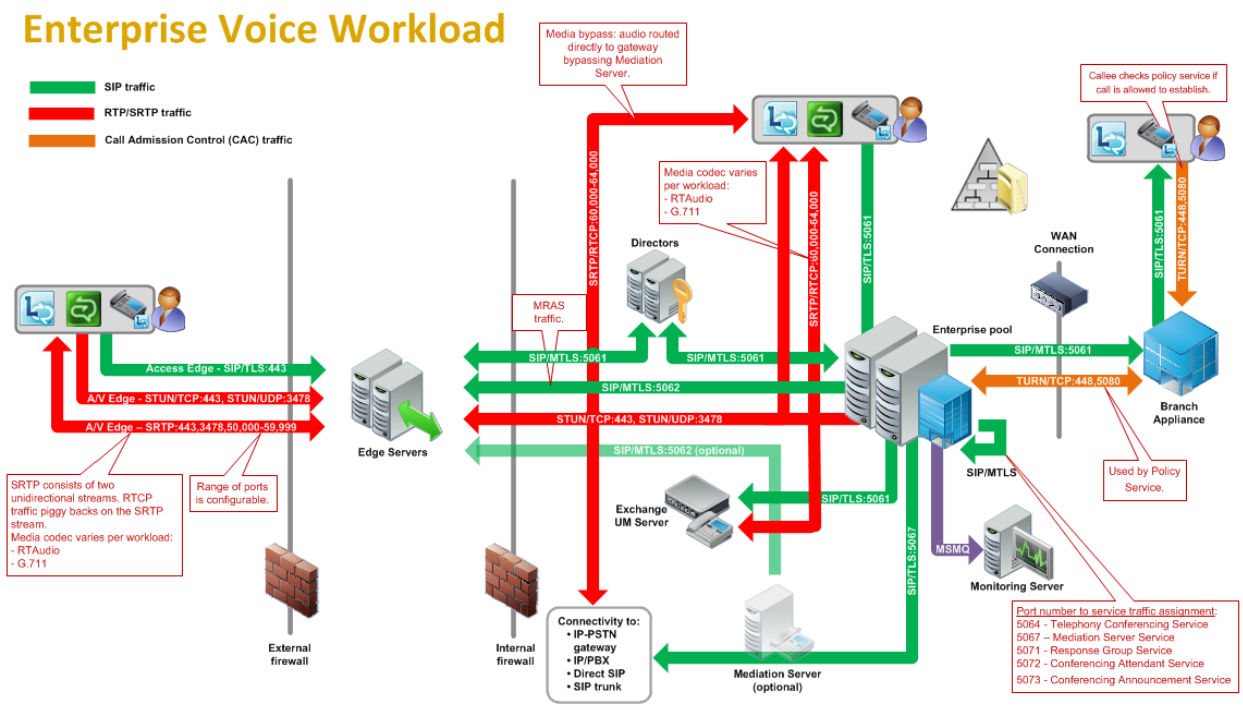

Figure 4: Enterprise voice workload

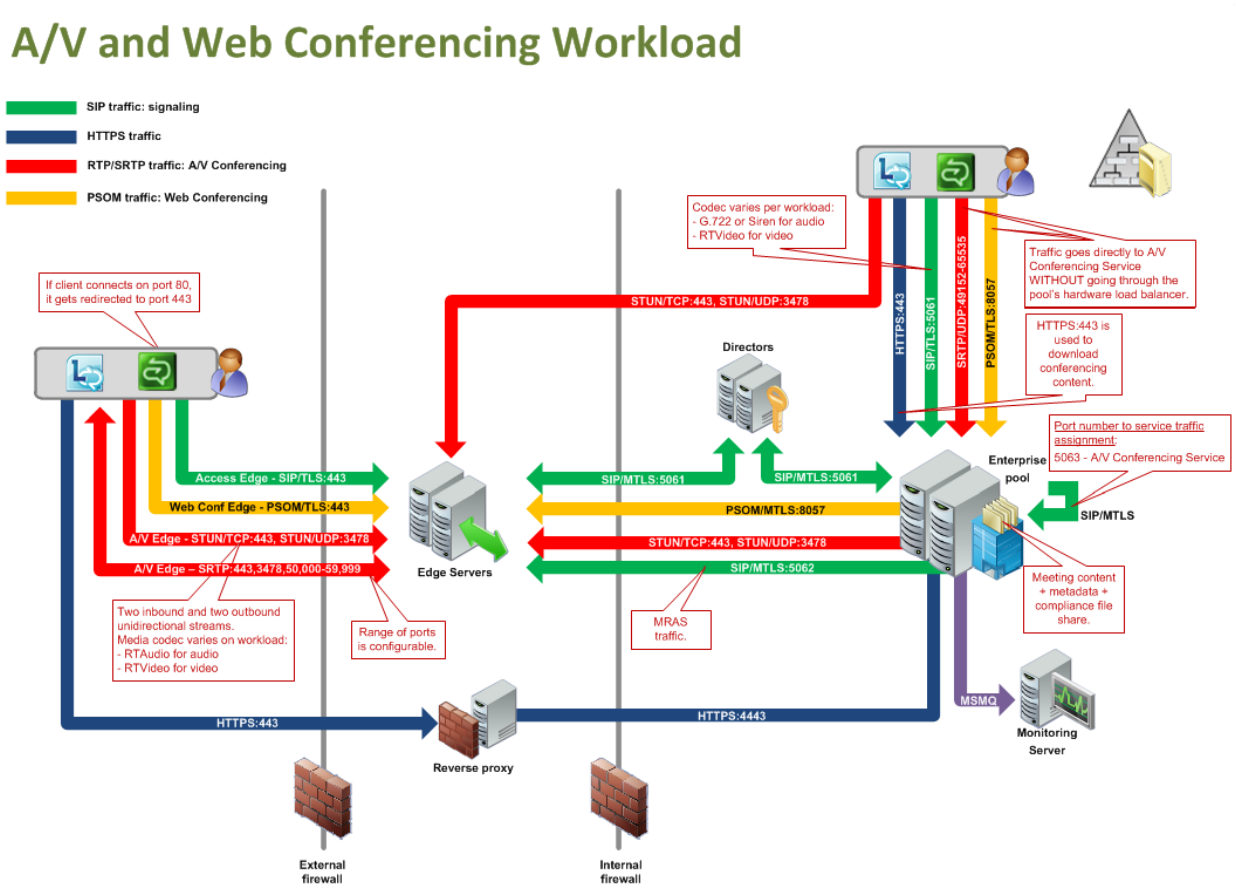

Figure 5: A/V and Web Conferencing Workload

2.2 Protocol Summary

The tables in this section provide a comprehensive list of the member protocols of the Office Communications Server system. The member protocols are grouped according to their primary purpose.

2.2.1 Directory Protocols

Protocols in this table enable protocol clients and protocol servers to authenticate, authorize, manage, and search for users. These protocols describe the data that Communications Server reads from the Active Directory **directory service (DS)** and stores in its data store while keeping this information synchronized with any changes made in Active Directory. The directory service (DS) serves as the authoritative source of user information and forest-level settings used by Communications Server. This information is used for many purposes, including **authentication**, authorization, management, and searching users.

Protocol name	Description	Short name
Address Book File Structure	Describes the format of the Address Book files that are produced daily by the **Address Book Server (ABS)** and accessed by protocol clients to search for users, contacts, and groups stored in Active Directory. In addition, this data can be used to perform reverse number lookup for voice calls.	[MS-ABS]

Protocol name	Description	Short name
Distribution List Expansion Protocol	Identifies a protocol for Office Communicator to discover members of a distribution list.	[MS-DLX]

2.2.2 Signaling and Control Channel Protocols

The following protocols describe the use of the Session Initiation Protocol (SIP) and conference protocols to enable multimedia and conferencing.

2.2.2.1 Session Initiation Protocols

Protocols in this table describe extensions made to the Session Initiation Protocol (SIP) that enhance the functionality provided by Communications Server. Communications Server is based on SIP. It acts like a SIP registrar and proxy, as described by [RFC3261]. SIP is used by terminals to establish, modify, and terminate multimedia sessions or calls.

Protocol name	Description	Short name
Connection Management Protocol	Describes the functional behavior for a protocol client to automatically discover the address of the protocol server, and for maintaining a persistent, reliable, in-order transport between them.	[MS-CONMGMT]
Routing to Exchange Unified Messaging Extensions	Includes Session Initiation Protocol (SIP) extensions that are used by Communications Server to route calls to Exchange Unified Messaging and to generate user notification e-mails on call events.	[MS-EUMR]
Call Control for Exchange Unified Messaging Protocol Extensions	Describes the SIP extensions that are used to integrate Office Communicator and Exchange Unified Messaging to play voice messages and use voice commands to manage Exchange Unified Messaging mailboxes.	[MS-OCEXUM]
Session Initiation Protocol (SIP) for PSTN Calls Extensions	Describes the SIP extensions for the interface between Office Communicator and Communications Server to communicate with public switched telephone network (PSTN) and **private branch exchange (PBX)**.	[MS-OCPSTN]
Client Error Reporting Protocol	Describes the protocol for Communications Server to report diagnostic and troubleshooting information to the SIP-based protocol client and for the SIP-based protocol client to report an error to Communications Server.	[MS-OCER]
Presence Protocol	Describes the extensions of SIP that make up the Presence Protocol used by Office Communicator and Communications Server to allow publishers and subscribers to exchange presence-related data over SIP.	[MS-PRES]
Quality of Experience Monitoring Server Protocol	Describes the protocol used for publishing audio and video **Quality of Experience (QoE)** metrics.	[MS-QoE]
Session Initiation Protocol (SIP) Compression Protocol	Describes a SIP extension to compress data between the protocol client and the protocol server. The protocol has two phases. The negotiation phase advertises and exchanges compression capabilities. The SIP Compression Protocol uses a modified form of the Microsoft Point-to-Point Compression (MPPC) Protocol, as described in [RFC2118], to	[MS-SIPCOMP]

Protocol name	Description	Short name
	compress SIP data.	
Session Initiation Protocol (SIP) Authentication Extensions	Describes SIP extensions used for authentication functionality. This protocol defines **NT LAN Manager (NTLM) Authentication Protocol**, **Kerberos**, and Transport Layer Security with Derived Session Key (TLS-DSK) authentication schemes based on the general authentication framework described in [RFC3261]. This protocol also describes the details and extensions for the Asserted Identity mechanism, which is based on [RFC3325], and the Referred-By mechanism, which is based on [RFC3892].	[MS-SIPAE]
Session Initiation Protocol (SIP) Routing Extensions	Describes SIP extensions for call routing used by SIP-based protocol clients, proxies, and protocol servers. SIP Routing Extensions also include extensions to SIMPLE-based presence, as described in [RFC3261] and [RFC3265].	[MS-SIPRE]
Session Initiation Protocol (SIP) Registration Extensions	Describes SIP extensions to enable Communications Server to provision the protocol clients as part of the registration process.	[MS-SIPREGE]
Response Group Service Web Service Protocol	Describes the procedure to enable a protocol client to access **agent** information exposed by a protocol server.	[MS-RGSWS]<1>
PSOM Shared Object Messaging Protocol	Describes the PSOM Shared Object Messaging (PSOM) protocol that is used to exchange messages between the protocol client and protocol server. A message typically represents a method invocation of a remote object, with a sequence of understood parameters. This protocol is designed to facilitate communications for data collaboration and Web conferencing applications.	[MS-PSOM]<2>
Session Initiation Protocol (SIP) Application Protocol	Describes the Session Initiation Protocol (SIP) Application Protocol. This protocol is a collection of independent proprietary client-server protocols that are used to provide enhanced functionality to Session Initiation Protocol (SIP)-based communication systems.	[MS-SIPAPP]<3>
OC Authentication Web Service Protocol	Describes the OC Authentication Web Service Protocol. This protocol defines the message formats, protocol server behavior, and protocol client behavior for the purposes of authentication and **certificate** enrollment.	[MS-OCAUTHWS]<4>
Web Service for E911 Support Protocol	Describes the Location Information Web Service interface that is used by protocol clients to retrieve locations associated with network identifiers, or locations within a city. A location is a civic address with up to room-level granularity. The network identifiers that can be specified are the Wireless Access Point, Received Signal Strength Indication, Media Access Control Address, Chassis, Port, Subnet, and Internet Protocol Address.	[MS-E911WS]<5>
Extensible Chat Control Over Session Initiation Protocol (SIP)	Describes an XML-based protocol for transmitting data between Group Chat servers and Lync clients by using SIP INFO methods. In addition to transporting the chat messages, it provides support for chat room invitations, activity notifications, and posting of files.	[MS-XCCOSIP]<6>
Persistent Chat Web Protocol	Describes a protocol that provides a mechanism to allow the client of a persistent chat system to start an external chat room management web application.	[MS-OCGCWEB]<7>

2.2.2.2 Conference Protocols

Protocols in this table enable protocol clients and protocol servers to establish and maintain the state of a conference. In the Communications Server system, Centralized Conference Control Protocol (C3P) is used by protocol clients, front-end servers, and conferencing servers to establish and maintain the state of a conference.

Protocol name	Description	Short name
Centralized Conference Control Protocol: Basic Architecture and Signaling	Describes the use of C3P by Office Communicator for activating, modifying, and controlling conferences and remaining synchronized with the state of a conference that is hosted by Communications Server.	[MS-CONFBAS]
Centralized Conference Control Protocol: Provisioning	Supplements the Centralized Conference Control Protocol: Basic Architecture and Signaling protocol (as described in [MS-CONFBAS]) by describing the use of C3P by an organizer's protocol client for creating, modifying, and deleting conferences hosted by Communications Server.	[MS-CONFPRO]
Centralized Conference Control Protocol: Instant Messaging Extensions	Describes the extensions to the Centralized Conference Control Protocol: Basic Architecture and Signaling protocol (as described in [MS-CONFBAS]) that are used by protocol clients during multiparty IM conferences hosted by Communications Server.	[MS-CONFIM]
Centralized Conference Control Protocol: Audio-Video Extensions	Describes the extensions to the Centralized Conference Control Protocol: Basic Architecture and Signaling protocol (as described in [MS-CONFBAS]) that are used by protocol clients during multiparty audio/video conferences hosted by Communications Server.	[MS-CONFAV]
Centralized Conference Control Protocol: Application Sharing Extensions	Describes the extensions to the Centralized Conference Control Protocol: Basic Architecture and Signaling protocol (as described in [MS-CONFBAS]) that relate to application sharing media content that is transferred using the **Real-Time Transport Protocol (RTP)** [RFC3550] and hosted by Communications Server.	[MS-CONFAS]<8>
XML Schema for Media Control Extensions	Extends the **XML** message semantics for carrying video control messages in SIP INFO methods. In multiparty video sessions, these extensions provide a mechanism that freezes unused video streams, thereby minimizing the load on the network.	[MS-XMLMC]

2.2.2.3 HTTP Protocols

The following table describes protocols used by clients to communicate with communication server components using HTTP to consume real time communication services for signaling.

Protocol name	Description	Short name
Microsoft Online Conference Scheduling and Management Protocol	Describes the protocol used to communicate with Unified Communications Web API components of the Lync Server to enumerate, create, delete, and edit scheduled online conferences hosted by the Lync Server.	[MS-OCSMP]

Protocol name	Description	Short name
Lync Autodiscover Web Service Protocol	Describes the protocol used to determine where to access specific Lync resources, including Lync web services and SIP entry points.	[MS-OCDISCWS]
Unified Communications Call Via Work Protocol	Specifies the protocol that defines how an application can create a PSTN phone call between a desktop or mobile phone and a remote party.	[MS-CVWREST]
Unified Communications Event Channel Protocol	Specifies the protocol that describes a mechanism that web applications can use to retrieve notifications about changes to resources.	[MS-ECREST]

2.2.3 Media Protocols

The following protocols describe the use of the Real-Time Transport Protocol (RTP) and **Interactive Connectivity Establishment (ICE)** protocols to authenticate protocol clients for Communications Server and identify the way audio and video traffic is established over the Internet.

2.2.3.1 Real-Time Protocols

Protocols in this table enable transmission of real-time data between multimedia **endpoints**. The Real-Time Transport Protocol (RTP) is a set of network transport functions suitable for applications transmitting real-time data, such as audio and video, from one multimedia endpoint to one or more multimedia endpoints. During a Communications Server conference that includes audio, video, desktop, or application-sharing data, the protocol client connects to the Audio/Video/Application Sharing Conferencing Server, and media is exchanged through the RTP. An RTP session is established using SIP/SDP, which manages the negotiation for the RTP session, including defining the transport, payload, and security parameters. The RTP and its associated control protocol, **Real-Time Transport Control Protocol (RTCP)**, are formally described in [RFC3550]. In addition, [RFC3551] defines the set of payload-type codes and payload formats for audio and video.

Protocol name	Description	Short name
Exchange Unified Messaging Session Description Protocol Extension	Describes the extensions to SDP that negotiate and establish audio calls between protocol servers and unified messaging servers to play or record voice messages and to manage the unified messaging mailbox by using touch-tone commands.	[MS-EUMSDP]
RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals Extensions	Describes the payload format for transmitting **dual-tone multi-frequency (DTMF)** signaling, tone signals, and telephony events in RTP packets.	[MS-DTMF]
RTP Payload Format for H.264 Video Streams Extensions	Describes the payload format for encapsulating an H.264 video stream.	[MS-H264PF]
Real-time Transport Protocol (RTP) Extensions	Extends the standard Real-Time Transport Protocol (RTP) [RFC3550]. The extensions define features such as dominant speaker notification, enhanced host security,	[MS-RTP]

Protocol name	Description	Short name
	bandwidth estimation, and lost packet notification.	
RTP Payload for Redundant Audio Data Extensions	Describes a payload format that contains redundant audio encoding to help reduce packet loss. If a packet is dropped, redundant data is carried in a subsequent packet so that the lost data can be reconstructed.	[MS-RTPRADEX]
RTP for Application Sharing Payload Format Extensions	Extends the Real-time Transport Protocol (RTP) Extensions protocol (as described in [MS-RTP]) with a set of Microsoft® proprietary extensions to the base Real-Time Transport Protocol (RTP) [RFC3550], to transfer the application-sharing payload that is encoded in the graphics format described by the Remote Desktop Protocol: Basic Connectivity and Graphics Remoting Specification [MS-RDPBCGR].	[MS-RTASPF]<9>
RTP Payload Format for RT Video Streams Extensions	Describes the RTP payload format for encapsulating an RTVideo (real-time video) stream.	[MS-RTVPF]
Session Description Protocol (SDP) Version 2.0 Protocol Extensions	Describes the extensions to SDP that enable protocol clients to negotiate advanced media session capabilities with Communications Server.	[MS-SDPEXT]
Secure Real-time Transport Protocol (SRTP) Extensions	Describes a framework for **encryption** and message authentication for both the RTP and RTCP streams. The protocol client and protocol server use SRTP when exchanging RTP traffic in either direction.	[MS-SRTP]
Scale Secure Real-time Transport Protocol (SSRTP) Extensions	Describes the extensions to SRTP that improve performance in scenarios where the same RTP payload is distributed to a large number of recipients. This includes cryptographic and message authentication processes that differ from the Secure Real-time Transport Protocol (SRTP) Extensions, as described in [MS-SRTP].	[MS-SSRTP]

2.2.3.2 Interactive Connectivity Establishment Protocols

Protocols in this table enable the process of setting up media channels between endpoints. Interactive Connectivity Establishment (ICE) describes a protocol for setting up media channels between two endpoints, for example, Office Communicator clients, in a way that allows them to traverse **network address translation (NAT)** computers and firewalls.

Protocol name	Description	Short name
Audio Video Edge Authentication Protocol Specification	Describes how to provide protocol clients with Communications Server security tokens, which are used to authenticate protocol clients with the Audio/Video Edge Servers.	[MS-AVEDGEA]
Interactive Connectivity Establishment (ICE) Extensions	Establishes audio and video RTP streams between two endpoints in a way that allows them to traverse network address translation (NAT) computers and firewalls. Describes generalized **Simple Traversal of UDP through NAT (STUN)** processing and event timers.	[MS-ICE]

Protocol name	Description	Short name
Interactive Connectivity Establishment (ICE) Extensions 2.0	Establishes audio and video RTP streams between two endpoints in a way that allows them to traverse network address translation (NAT) computers and firewalls.<10> Specifies generalized Simple Traversal of UDP through NAT (STUN) processing and event timers.	[MS-ICE2]<11>
Interactive Connectivity Establishment (ICE) 2.0 Bandwidth Management Extensions	Describes how to determine and enforce bandwidth policy constraints for RTP media streams.<12> This protocol facilitates communication with a **Traversal Using Relay NAT (TURN)** Bandwidth Management Extensions protocol-based server, also referred to as a bandwidth policy server, which supports network bandwidth utilization management and access control. This protocol enforces bandwidth policy constraints and ensures that policy-restricted paths are not used for media flow. This protocol describes a reporting mechanism used by a **bandwidth management endpoint** to report the path and the bandwidth being utilized by the media session to a bandwidth policy server.	[MS-ICE2BWM]<13>
Traversal Using Relay NAT (TURN) Extensions	Enables a protocol client behind a NAT or a firewall to acquire a transport address from a TURN server that is located on the Internet. The protocol client can then provide this transport address to the external peer, which can use it to establish connectivity and to exchange media with Communications Server.	[MS-TURN]
Traversal Using Relay NAT (TURN) Bandwidth Management Extensions	Extends the Traversal Using Relay NAT (TURN) protocol described in [MS-TURN] to provide support for controlling access to network bandwidth.<14>	[MS-TURNBWM]<15>

2.3 Environment

The following sections identify the context in which the system exists. This includes the systems that use the interfaces provided by this system of protocols, other systems that depend on this system, and, as appropriate, how components of the system communicate.

2.3.1 Dependencies on This System

The following systems depend on the Communications Server system:

- SIP-based protocol clients
- Federated links
- Public IM providers
- Gateways
- Server applications

The following sections summarize these systems.

Systems such as gateways and public IM providers can interface with Communications Server at the protocol level over the IP network. Communications Server also provides a number of programmable

interfaces (APIs) to abstract these wire protocols, simplify connectivity, and make it possible to support a wide variety of systems that can connect to Communications Server:

- Unified Communications Client SDK (UCC)
- Office Communicator SDK (OC Automation)
- Office Communicator "14" SDK

2.3.1.1 SIP-Based Clients

Protocol clients capable of communicating with Communications Server directly over SIP are referred to as SIP-based clients, because they support a native SIP stack. Such protocol clients offer a SIP stack that is interoperable with the SIP and media extensions of Communications Server. Examples of SIP-based clients are software-based protocol clients such as Office Communicator, and SIP-based phones such as Office Communicator Phone Edition and Office Communicator Mobile.

2.3.1.2 Federated Links

Organizations using Communications Server can allow their users to communicate with users from other enterprises over a federated link. A federated link is established between the two organizations to allow these communications. Some federated links can be established using **Extensible Message and Presence Protocol (XMPP)** described in [RFC6120] and [RFC6121].

2.3.1.3 Public IM Providers

Communications Server can interoperate with public IM providers such as AIM, Yahoo!, and MSN. This interoperability allows external users signed in to any of these providers to communicate over IM to an enterprise user connected to Communications Server as long as there is a public IM connectivity established between the enterprise and the public IM provider.

2.3.1.4 Gateways

Gateways provide interconnectivity between the Communications Server network and other networks such as PBX, PSTN, XMPP, and other non-SIP-based networks. Gateways extend the connectivity reach of users signed in to Communications Server into non-SIP-based networks. Examples of gateways include:

- SIP/PSTN gateways
- RCC gateways
- IP-PBX

2.3.1.5 Server Applications

Server applications can be built as services using Communications Server's highly scalable API, UCMA 2.0, or MSPL services. Such services provide specialized functions in addition to the functionality provided by Communications Server. Examples of such services include Exchange Unified Messaging and ForeFront Security for Communications Server.

2.3.2 Dependencies on Other Systems/Components

The Communications Server system depends on these systems in order to function:

- Active Directory directory service

- DNS service
- Certificate authority service
- **Internet Information Services (IIS)**
- Microsoft Service Message Queue
- Hardware load balancers
- Exchange Unified Messaging
- Gateways
- Microsoft Office Web Access Companion Server

The following sections outline these systems.

2.3.2.1 Active Directory

Communications Server is dependent on Active Directory domain controllers to provide authentication services and security policies. These domain controllers provide an LDAP-enabled directory service (DS) that stores users' information such as name and SIP URI.

2.3.2.2 DNS Service

Domain Name System (DNS) is required so that Communications Server and protocol clients can resolve host names to IP addresses (A records), resolve SRV records, and route SIP traffic accordingly. The DNS service plays an integral role for both internal (within the organization) and external communications routing.

2.3.2.3 Certificate Authority Service

Communications Server uses certificates to perform strong authentication of protocol servers before Transport Layer Security (TLS) communications can be established. This authentication mechanism relies on a trusted **certification authority (CA)**.

2.3.2.4 Internet Information Services

Communications Server requires Internet Information Services (IIS), to be configured in order to service users using the HTTPS protocol for address book downloads, distribution list expansion, and Web conferencing document-sharing.

2.3.2.5 Microsoft Service Message Queue

To enable archival of IMs, the archiving server role requires the Microsoft Service Message Queue (MSMQ) feature, as described in [MS-MQSD], to be configured on all Standard Edition servers and Enterprise pool front-end servers where archiving is enabled.

Similarly, to enable monitoring of services, the monitoring server role requires MSMQ to be installed on all Standard Edition Servers, Enterprise pool front-end servers, and mediation servers that are monitoring Call Data Records.

2.3.2.6 Hardware Load Balancers

To perform load-balancing of protocol client connections across multiple protocol servers, Communications Server relies on hardware load balancers. This assures higher availability of service.

For simpler Standard Edition deployments of Communications Server, a hardware load balancer is not required.

2.3.2.7 Exchange Unified Messaging

For voicemail, missed call notifications, and auto attendant support, Communications Server requires Exchange Unified Messaging. SIP and RTP traffic are routed by Communications Server to Exchange Unified Messaging to store this information in the user's Microsoft Exchange Server mailbox.

2.3.2.8 Gateways

To connect to different and proprietary networks such as public switch telephone network (PSTN) and private branch exchange (PBX) systems, Communications Server relies on gateways to translate SIP and media protocols into the proprietary protocols used by these systems.

2.3.2.9 Microsoft Office Web Access Companion Server

To share the content of Microsoft Office documents between conference participants Communications Server relies on Microsoft Office Web Access Companion Server. The Data Conferencing Server component of Communication Server implements Web Application Open Platform Interface (WOPI) host as described in [MS-WOPI]. The Data Conferencing Server component utilizes WopiSrc and access_token query parameters described in [MS-WOPI] in URLs that it distributes to protocol clients to provide access to Microsoft Office documents.

2.4 Assumptions and Preconditions

This section summarizes the assumptions and preconditions required by the system. The scope of this discussion is intended to be implementation-independent and is limited to the system level.

- A directory service (DS) domain controller is required to service the protocol server domain, authenticate requests, and handle management tasks.
- The directory service (DS) is accessible to Communications Server. Servers within Communications Server are accessible among themselves. Any intermediate firewalls, routers, or connection points between components of the system have all the required ports and gateways open for communication between them.
- The servers within Communications Server are members of the domain.
- Domain users are provisioned for Unified Communications before they can sign in to the Communications Server infrastructure.
- For the Enterprise pool, a DNS SRV record is configured to map the pool's **fully qualified domain name (FQDN)** to the Virtual IP address of the hardware load balancer.
- Communications Server is reachable by external protocol clients via an established public IP address (or IP addresses).
- The appropriate DNS SRV records are configured to map the SIP domain to the public IP addresses corresponding to the externally available Communications Server. The SIP SRV records are propagated to the public networks so that all intended protocol clients can resolve the domain name.
- The Communications Server functional components are started collectively, and Communications Server accepts protocol client and protocol server requests.
- For Unified Messaging (UM), Microsoft Exchange UM is deployed in the same Active Directory forest as the Communications Server infrastructure to be integrated together.

2.5 Use Cases

The following use cases are provided to facilitate an understanding of the Office Communications Server Protocols system overall:

- Discover the server and establish a connection
- Perform registration and authentication
- Perform client bootstrap
- Get an address location
- Perform the sign-in process
- Change presence information
- Download the address book
- Expand a distribution list
- Initiate instant messaging
- Add a contact
- Use multiple endpoints
- Initiate a call from a client
- Add video to a voice call
- Accept a voice call
- Terminate a voice call
- Send a Quality of Experience report
- Start and join a multiparty audio conference
- Subscribe to conference events
- Share a desktop
- Share a whiteboard
- Join a Chat Room

These use cases provide a high-level summary of the functions that are executed between Office Communicator and Communications Server, and include the core types of activity that a typical protocol client conducts with the system. The examples in section 3 present a number of scenarios that illustrate how one or more of the use cases can work in conjunction to achieve specific results.

These use cases are not intended to provide a thorough and complete model of the system for any implementation. For example, they do not include all the messages for the protocol exchange, and the individual document references need to be used to find the protocol message details.

2.5.1 Discover the Server and Establish a Connection

This use case, illustrated in the following diagram, describes how a protocol client discovers the protocol server and establishes the connection to the server.

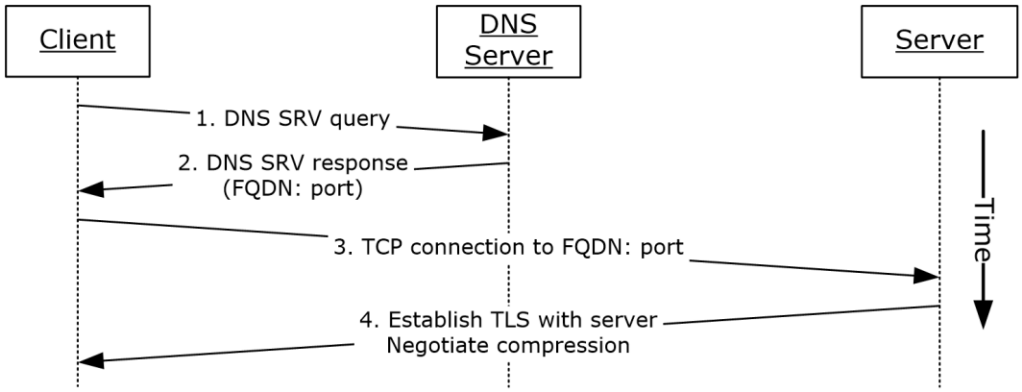

Figure 6: Steps for discovering the protocol server and establishing a connection

References

- [MS-CONMGMT]
- [MS-SIPCOMP]

Preconditions

- DNS has been populated with the appropriate DNS SRV records, as described in [MS-CONMGMT].

Steps

1. The protocol client uses the domain portion of SIP-URI for DNS lookup to discover the hostname of the user's home server or pool, as described in [MS-CONMGMT].
2. The protocol client processes the DNS SRV response, as described in [MS-CONMGMT], to identify the protocol server FQDN and port to connect to, and then initiates a TCP connection to the protocol server FQDN and port.
3. The protocol client optionally negotiates Transport Layer Security (TLS) with the protocol server; that is, it verifies the server certificate.
4. If the connection is encrypted (TLS) and if compression is enabled based on group policy settings, the protocol client can request compression on the connection, as described in [MS-SIPCOMP].

Post-conditions

- The protocol client has discovered and connected to the protocol server and is now ready to sign in.

2.5.2 Perform Registration and Authentication

This use case, illustrated in the following diagram, describes how a protocol client registers and authenticates to the protocol server.

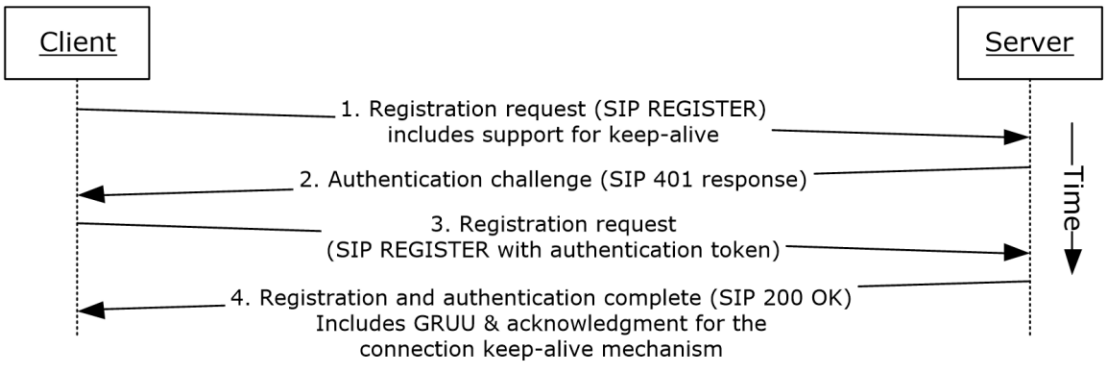

Figure 7: Steps for performing registration and authentication (2)

References

- [MS-CONMGMT]
- [MS-PRES]
- [MS-SIPAE]
- [MS-SIPRE]
- [MS-SIPREGE]

Preconditions

- The protocol client is connected to the server.

Steps

1. The protocol client sends a REGISTER request to the user's home server. The request asks the server to provide the following information:
   - A **Globally Routable User Agent URI (GRUU)**, as described in [MS-SIPRE].
   - Acknowledgment of support for Resource lists for enhanced presence, as described in [MS-PRES].
   - Acknowledgment of support for an XML document conforming to the enhanced presence XML schema, as described in [MS-PRES].
   - Acknowledgment of support for the connection keep-alive mechanism described in [MS-CONMGMT].
2. In response to the protocol client's REGISTER request, the server requests user authentication and offers the protocol client a choice of using either the Kerberos authentication protocol or the NT LAN Manager (NTLM) Authentication Protocol by sending a SIP authentication (2) challenge, such as a SIP 401 or 407 response, to the protocol client.
3. The protocol client then sends the appropriate authentication token in another REGISTER request to the server, as described in [MS-SIPAE].
4. The server verifies the protocol client's authentication token, as provided by the authentication extensions described in [MS-SIPAE]. The server returns a response to the protocol client that includes the following:

- The server generates a GRUU for the newly registered endpoint and returns it to the protocol client, as described in [MS-SIPRE] and [MS-SIPREGE].
- The server can also confirm support for the keep-alive mechanism, provide encrypted proof for the protocol client of the server's own authenticity, and offer a way to verify that the protocol client and server are in synch for user presence, as described in [MS-SIPREGE].

Post-conditions

- The protocol client is now authenticated and registered with the server.

2.5.3 Perform Client Bootstrap

This use case, illustrated in the following diagram, describes how a client completes sign-in, obtains information such as the contact list, and other relevant parameters from the protocol server after sign-in.

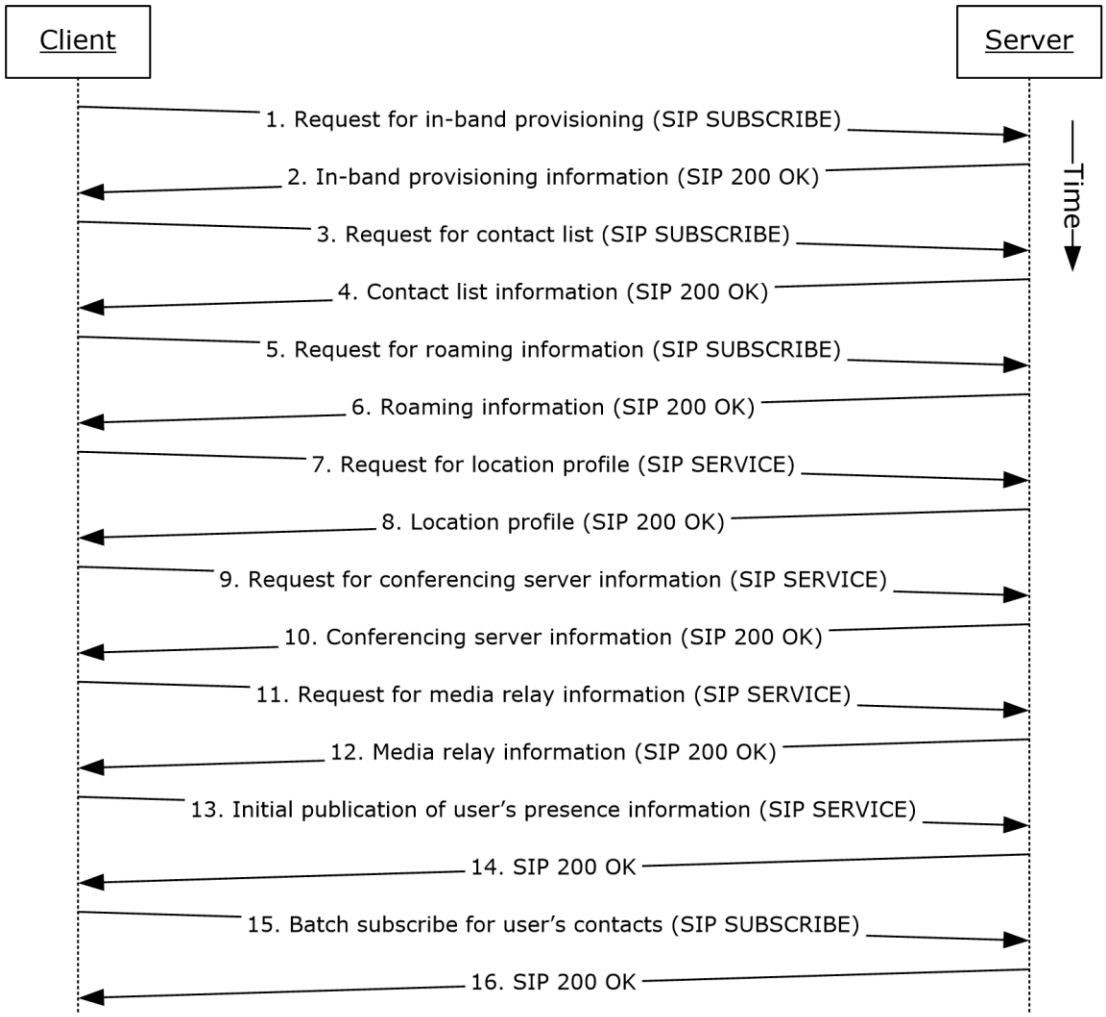

Figure 8: Steps for performing client bootstrap

References

- [MS-PRES]

- [MS-SIPREGE]
- [MS-SIPRE]
- [MS-AVEDGEA]

Preconditions

- The protocol client is connected to the protocol server.
- The protocol client is authenticated and registered with the protocol server.

Steps

1. The protocol client sends a request for in-band provisioning, as described in [MS-SIPREGE]. In-band provisioning is a mechanism through which a protocol server can provide a protocol client with initial configuration information, at a point when the protocol client does not yet have access to global policies stored in Active Directory, or with server configuration information, such as the ABS **Uniform Resource Locator (URL)** and Group Expansion web service URL. The protocol client sends a SUBSCRIBE request, as described in [MS-SIPREGE].
2. The protocol server sends a **200 OK** response to the SUBSCRIBE request that includes the protocol server configuration, various policies that the protocol client enforces, the URL of the ABS, and information essential for protocol client control of the user's desktop phone. These categories of information are defined in provisioning extensions, as described in [MS-SIPREGE].
3. The protocol client sends a SUBSCRIBE request for the Contacts and Groups information, as described in [MS-PRES] and [MS-SIPREGE].
4. The protocol server returns the Contact List and Groups, as described in [MS-SIPREGE].
5. The protocol client sends a SUBSCRIBE request for the user's own presence information, such as the user's contact card and calendar information, as described in [MS-PRES] and [MS-SIPREGE].
6. The protocol server sends a 200 OK response with the information listed in step 5.
7. The protocol client sends a SERVICE request to retrieve the user's location profile for a VoIP call, as described in [MS-SIPRE] and [MS-SIPREGE].
8. The protocol server sends a 200 OK response with the information listed in step 7.

   Note: This step assumes EnhancedEmergencyServices and LocationRequired settings are enabled in the user's location profile.
9. The protocol client sends a SERVICE request to retrieve information about available conferencing servers (MCUs), as described in [MS-SIPREGE].
10. The protocol server sends a 200 OK response with the information listed in step 9.
11. The protocol client sends a SERVICE request to obtain Media Relay authentication tokens, as described in [MS-AVEDGEA].

    Note: This step assumes an **Audio/Video Edge Server (A/V Edge Server)** is configured and both clients' support [MS-ICE] or [MS-ICE2]. If an A/V Edge Server is not configured or one or both of the clients do not support [MS-ICE] or [MS-ICE2] the protocol exchange will differ; see the detailed protocol documents for those protocol exchanges.
12. The protocol server sends a 200 OK response with the information listed in step 11.
13. The protocol client sends a SERVICE request to publish the user's presence information, as described in [MS-PRES].

14. The protocol server acknowledges publishing the information listed in step 13.

15. The protocol client issues a batch subscription request, as described in [MS-SIPREGE], for enhanced presence information, as described in [MS-PRES], for all members of the contact list that were returned by the server in step 4.

16. The protocol server acknowledges the batch subscription request sent in step 15.

Note: The order in which the protocol client fetches the information need not be the exact order specified above and can be in any order of its choice.

Post-conditions

- The protocol client is finished with bootstrapping and is ready to receive presence updates and initiate any communication.

2.5.4 Get an Address Location

This use case, illustrated in the following diagram, describes the Location Information Web Service interface that is used by protocol clients to retrieve a civic address location. Locations are associated with network identifiers that allow varying degrees of detail. If the client network identifiers are updated, a new request is invoked to the Location Information Server. This section follows the behavior described in product behavior note.<16>

Figure 9: Steps for getting an address location

References

- [MS-E911WS]

Preconditions

- The protocol client obtains the FQDN of the protocol server.

Steps

1. The protocol client sends an HTTPS-SOAP GetLocations request to the protocol server, as described in [MS-E911WS]. The request includes network identifiers as an input for locations.

2. The protocol server returns an HTTPS-SOAP GetLocations response, as described in [MS-E911WS].The locations description includes a list of civic addresses. If the list includes more than one address, then the protocol client selects one address, based on the user's input.

Post-conditions

- The protocol client caches its location and uses it in services that depend on location, such as an emergency call.

2.5.5 Perform the Sign-In Process

This use case describes how a protocol client signs in for the first time. This use case essentially is a combination of the use cases in the previous sections: 2.5.1, 2.5.2, 2.5.3, and 2.5.4.

References

- [MS-AVEDGEA]
- [MS-CONMGMT]
- [MS-PRES]
- [MS-SIPCOMP]
- [MS-SIPAE]
- [MS-SIPRE]
- [MS-SIPREGE]
- [MS-E911WS]

Preconditions

None.

Steps

1. Use case for server discovery, section 2.5.1.
2. Use case for registration and authentication, section 2.5.2.
3. Use case for performing client bootstrap, section 2.5.3.
4. Use case for getting the address location, section 2.5.4.

Post-conditions

- The protocol client is now signed on to the protocol server and is ready to initiate or receive communication.

2.5.6 Change Presence Information

This use case, illustrated in the following diagram, describes the publication of a change in a user's presence information.

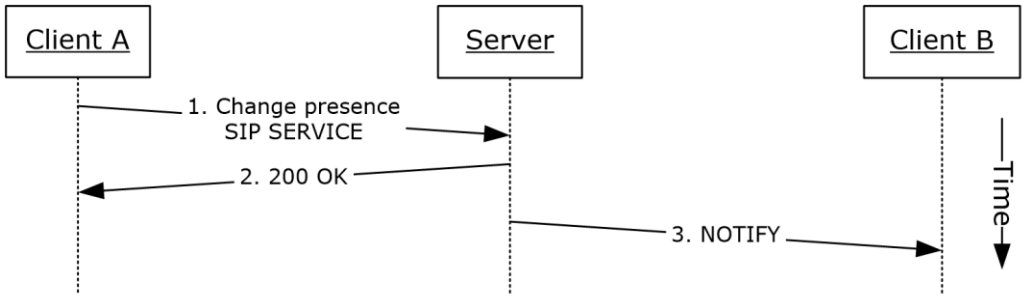

Figure 10: Steps for changing presence information

References

- [MS-PRES]

Preconditions

- The protocol clients are signed in, as described in section 2.5.5.

Steps

1. When a user's presence information changes, the user's protocol client (Client A) sends a SIP SERVICE request to the user's home server. The SIP SERVICE request contains one or more of the following:
   - The user's calendar and meetings obtained from Microsoft Exchange.
   - The device on which the protocol client is running and the device's capabilities.
   - The user's activity on a particular device.
   - The user's contact information, such as phone numbers, office location, and title.
2. The server responds with a 200 OK message, which contains the user's updated presence information.
3. The server sends a SIP NOTIFY request to all other protocol clients (for example, Client B) that subscribe to the user's presence information, as described in [MS-PRES].

Post-conditions

- The user's presence state has been changed and this change has been communicated to the protocol clients that have subscribed to the user's presence information.

2.5.7 Download the Address Book

This use case, illustrated in the following diagram, describes how the address book is downloaded or refreshed on the protocol client.

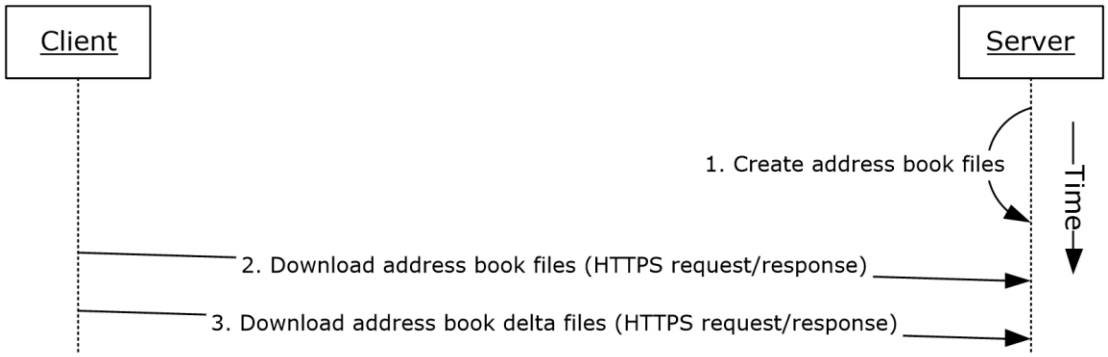

Figure 11: Steps for downloading the address book

References

- [MS-ABS]

Preconditions

- The protocol client is signed in, as described in section 2.5.5. The URL of the ABS is obtained by the protocol client during client bootstrap.

Steps

1. The protocol server creates both full and delta address book files, as described in [MS-ABS] and normalizes user phone numbers.
2. The protocol client downloads a copy of the address book by using the ABS URL that it received through in-band provisioning during client bootstrap to download the address book files.
3. The protocol client refreshes the local copy of the address book by using the address book server URL that it received through in-band provisioning during client bootstrap to download the address book delta files.

Post-conditions

- The protocol client has downloaded the address book files, which can then be used by the protocol client for search and reverse number lookup (RNL) for inbound calls.

2.5.8 Expand a Distribution List

This use case, illustrated in the following diagram, describes the expansion of a distribution list when a user selects a distribution list from the contact list and clicks + to expand the list.

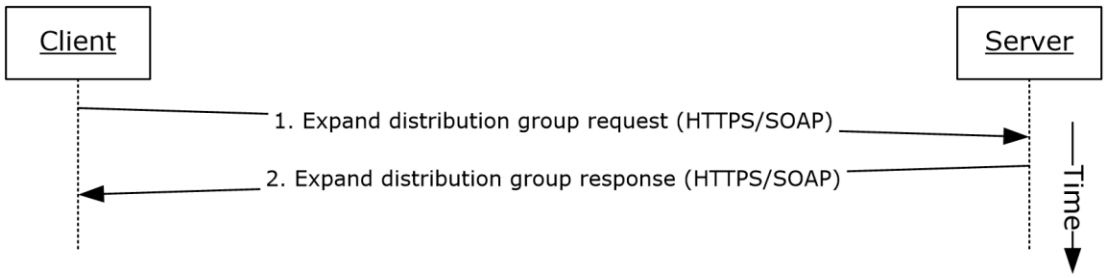

Figure 12: Steps for expanding a distribution list

References

- [MS-DLX]

Preconditions

- The protocol client is signed in, as described in section 2.5.5. The URL of the Distribution Group Expansion service is obtained by the protocol client during client bootstrap.

Steps

1. The protocol client sends an HTTP-SOAP request to the protocol server, as described in [MS-DLX].
2. The protocol server returns a distribution group, as described in [MS-DLX], in an HTTP-SOAP response. The response includes data for each member of the distribution group, such as the SIP URI, e-mail address, mail nickname, and display name.

Post-conditions

- The distribution list is expanded and a presence-polling request is initiated to all members of that list. The user can then optionally initiate communication to the list.

2.5.9 Initiate Instant Messaging

This use case, illustrated in the following diagram, describes how a user sends an instant message to another user.

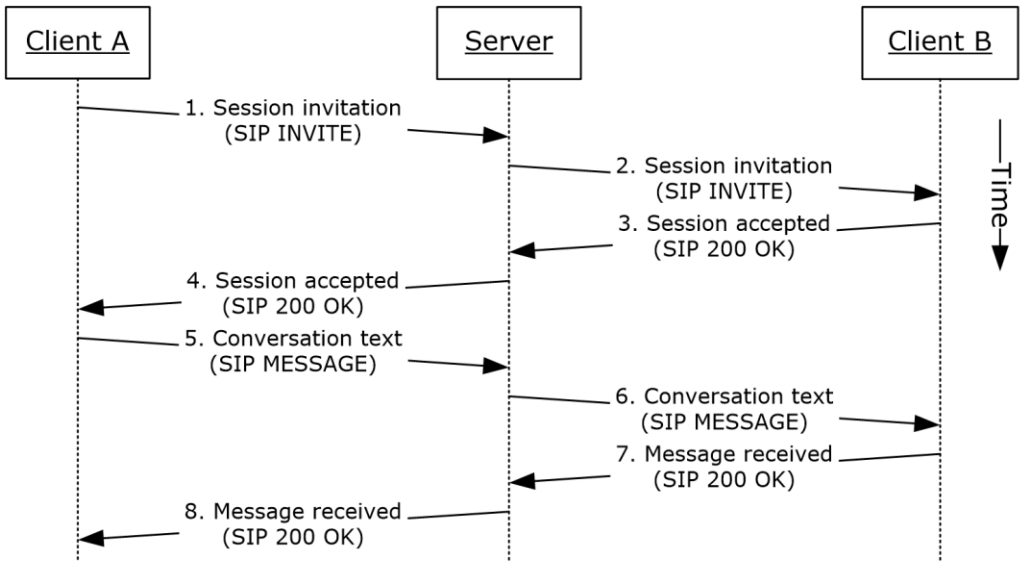

Figure 13: Steps for initiating instant messaging

References

- [MS-SIPRE]

Preconditions

- The protocol clients are signed in, as described in section 2.5.5.

Steps

1. Client A sends a session initiation message (SIP **INVITE**) with the target address of Client B to the protocol server by using the Session Initiation Protocol (SIP) Routing Extensions, as described in [MS-SIPRE].
2. The protocol server routes the session invitation to Client B by using the Session Initiation Protocol (SIP) Routing Extensions, as described in [MS-SIPRE].
3. Client B sends a session accepted response (SIP 200 OK) to the protocol server to accept the session.
4. The protocol server routes the session accepted response (SIP 200 OK) to Client A. The session is now established. The client A sends an **acknowledgment (ACK)** request in response to the 200 OK. Please note in the interest of brevity this step is not explicitly called out in the diagram above.
5. Client A sends the conversation text (SIP MESSAGE) and target address of Client B to the protocol server by using the Session Initiation Protocol (SIP) Routing Extensions, as described in [MS-SIPRE].
6. The protocol server routes the session conversation text to Client B by using the Session Initiation Protocol (SIP) Routing Extensions, as described in [MS-SIPRE].
7. Client B sends a message received response (SIP 200 OK) to the protocol server.

8. The protocol server routes the message received response (SIP 200 OK) to Client A.

Post-conditions

- The instant messaging (IM) conversation between Client A and Client B is in progress.

2.5.10 Add a Contact

This use case, illustrated in the following diagram, describes how a user adds a contact.

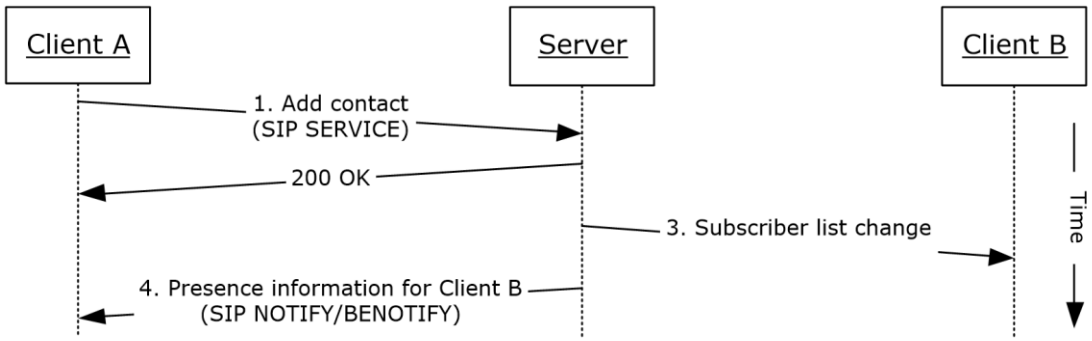

Figure 14: Steps for adding a contact

References

- [MS-PRES]

Preconditions

- The protocol clients are signed in, as described in section 2.5.5.

Steps

1. When a user (Client A) adds another user (Client B) to his or her contact list by entering the contact's SIP address, the user's protocol client (Client A) sends a service (SIP SERVICE) request to the protocol server, as described in [MS-PRES].
2. The protocol server adds the user (Client B) to the contact list and sends an acceptance of the same request (SIP 200 OK), as described in [MS-PRES].
3. The protocol server sends a subscriber list change notification to Client B, as described in [MS-PRES].
4. The protocol server sends the contact's presence information to Client A by using the NOTIFY/BENOTIFY mechanism, as described in [MS-PRES].

Step 3 and Step 4 can occur in different order.

Post-conditions

- The contact is added to the user's contact list.

2.5.11 Use Multiple Endpoints

This use case, illustrated in the following diagram, describes how a user can sign in using multiple endpoints and receive communications, such as an instant message or a VoIP call, initiated from another protocol client.

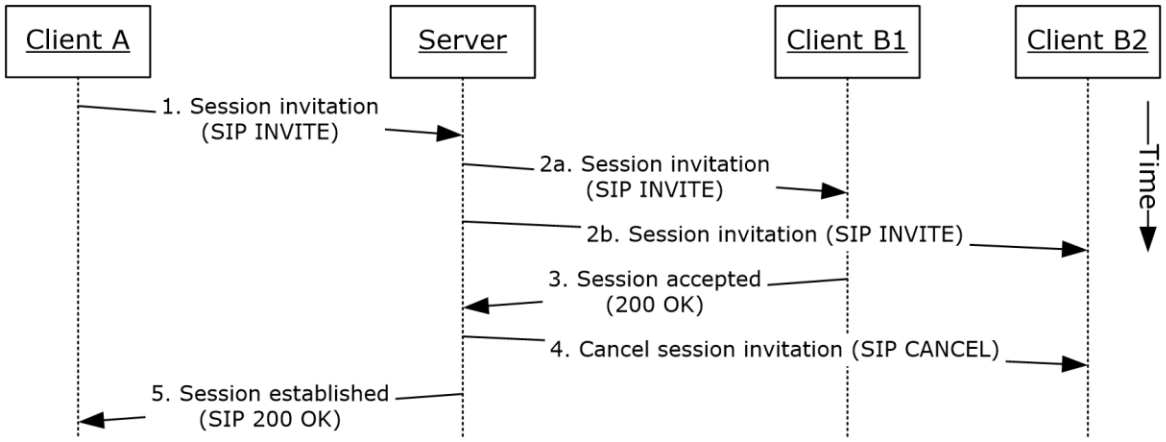

Figure 15: Steps for using multiple endpoints (5)

References

- [MS-PRES]
- [MS-SIPRE]

Preconditions

- The protocol clients are signed in, as described in section 2.5.5.
- User B is signed in with two protocol clients (Client B1 and Client B2).

Steps

1. A user (Client A) initiates a session invitation request (SIP INVITE) to User B.
2. The protocol server forks the session invitation (SIP INVITE) request to both of User B's endpoints, Client B1 and Client B2, as described in [MS-SIPRE].
3. User B's protocol clients elect the most suitable endpoint to accept the session invitation request on the user's behalf. This process is facilitated by publishing capabilities through the presence channel, as described in [MS-PRES]. In this case, Client B1 accepts the session invitation request and sends a session accepted response (SIP 200 OK) to the protocol server.
4. The protocol server cancels the session invitation request (SIP INVITE) that has been forked to the Client B2 endpoint, as described in [MS-SIPRE].
5. The protocol server sends a session invitation accepted response (SIP 200 OK) to Client A.

Post-conditions

- Client A is now communicating with User B (Client B1).

2.5.12 Initiate a Call from a Client

This use case, illustrated in the following diagram, describes how a protocol client makes a voice call to a remote user.

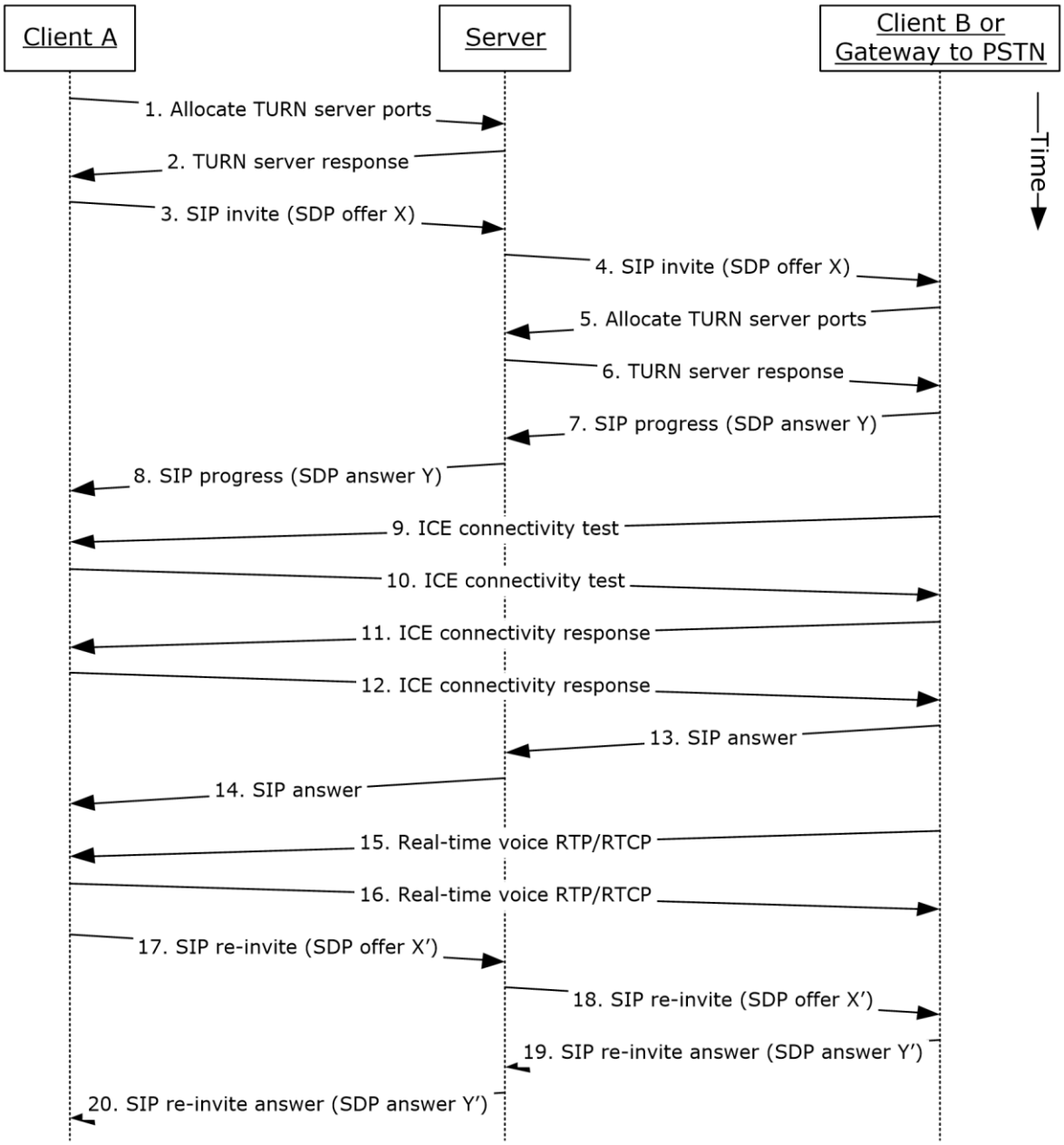

Figure 16: Steps for initiating a call from a protocol client

References

- [MS-AVEDGEA]
- [MS-ICE]
- [MS-ICE2]
- [MS-ICE2BWM]
- [MS-OCPSTN]
- [MS-RTP]

- [MS-RTPRADEX]
- [MS-SDPEXT]
- [MS-SIPRE]
- [MS-SRTP]
- [MS-SSRTP]
- [MS-TURN]
- [MS-TURNBWM]

Preconditions

- The protocol clients are signed in, as described in section 2.5.5.

Steps

> Note: This use case assumes an A/V Edge Server is configured and both clients' support [MS-ICE] or [MS-ICE2]. If an A/V Edge Server is not configured or one or both of the clients do not support [MS-ICE] or [MS-ICE2] the protocol exchange will differ; see the detailed protocol documents for those protocol exchanges.

1. Client A sends a request to allocate media ports on an edge protocol server for ICE candidates. For more information, see [MS-AVEDGEA], [MS-TURN], and [MS-TURNBWM].
2. The protocol server returns the allocated media ports to Client A. For more information, see [MS-TURN].
3. Client A sends a session invitation message with a media description offer and target address of Client B to the protocol server. For more information, see [MS-SIPRE], [MS-OCPSTN], and [MS-SDPEXT].
4. The protocol server routes the session invitation to Client B, or the protocol client's protocol server. For more information, see [MS-SIPRE].
5. Client B sends a request to allocate media ports on an edge protocol server for ICE candidates. For more information, see [MS-AVEDGEA], [MS-TURN], and [MS-TURNBWM].
6. The protocol server returns the allocated media ports to Client B. For more information, see [MS-TURN].
7. Client B sends a call progress response with a media description answer to the protocol server. For more information, see [MS-SIPRE], [MS-OCPSTN], and [MS-SDPEXT].
8. The protocol server routes the call progress response to Client A. For more information, see [MS-SIPRE].
9. Client B sends an ICE connectivity test message to Client A. For more information, see [MS-ICE] or [MS-ICE2]<17> and [MS-ICE2BWM].<18>
10. Client A sends an ICE connectivity test message to Client B. For more information, see [MS-ICE] or [MS-ICE2]<19> and [MS-ICE2BWM].<20>
11. Client B responds to the ICE connectivity test message with an ICE connectivity response message. For more information, see [MS-ICE] or [MS-ICE2]<21> and [MS-ICE2BWM].<22>
12. Client A responds to the ICE connectivity test message with an ICE connectivity response message. For more information, see [MS-ICE] or [MS-ICE2]<23> and [MS-ICE2BWM].<24>

13. Client B answers the call. For more information, see [MS-SIPRE].

14. The protocol server forwards the answer to Client A.

15. Client B sends real-time voice packets to Client A. For more information, see [MS-RTP], [MS-RTPRADEX], [MS-SRTP], and [MS-SSRTP].

16. Client A sends real-time voice packets to Client B. For more information, see [MS-RTP], [MS-RTPRADEX], [MS-SRTP], and [MS-SSRTP].

17. Client A sends an updated media description offer to the protocol server that reflects the media ports that were selected. For more information, see [MS-SIPRE] and [MS-SDPEXT].

18. The protocol server forwards the updated media description offer to Client B. For more information, see [MS-SIPRE].

19. Client B sends a media description answer to the protocol server. For more information, see [MS-SIPRE] and [MS-SDPEXT].

20. The protocol server forwards the media description answer to Client A. For more information, see [MS-SIPRE].

Post-conditions

- A session is established between Client A and Client B, and real-time voice packets are exchanged between Client A and Client B.

2.5.13 Add Video to a Voice Call

This use case, illustrated in the following diagram, describes how a protocol client adds video to a voice call with a remote user.

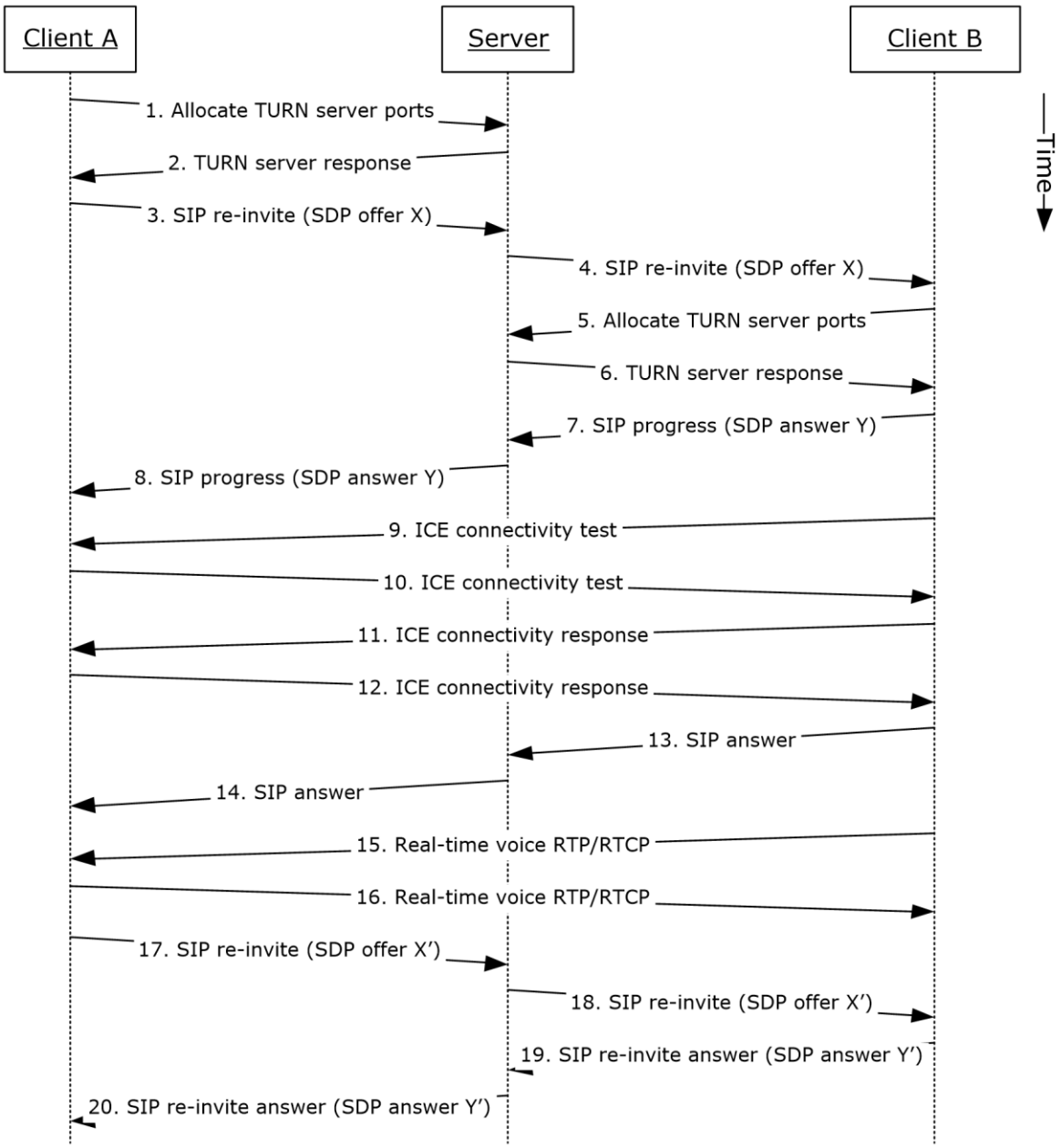

Figure 17: Steps for adding video to a voice call

References

- [MS-AVEDGEA]
- [MS-H264PF]
- [MS-ICE]
- [MS-ICE2]
- [MS-ICE2BWM]
- [MS-OCPSTN]

- [MS-RTP]
- [MS-RTPRADEX]
- [MS-RTVPF]
- [MS-SDPEXT]
- [MS-SIPRE]
- [MS-SRTP]
- [MS-SSRTP]
- [MS-TURN]
- [MS-TURNBWM]

Preconditions

- The protocol clients are signed in, as described in section 2.5.5.
- Initiate a voice call, as described in section 2.5.12, or accept a voice call, as described in section 2.5.14.

Steps

- These steps are the same as the steps in section 2.5.12.

Post-conditions

- Real-time voice and video packets are exchanged between Client A and Client B.

2.5.14 Accept a Voice Call

This use case, illustrated in the following diagram, describes how a protocol client accepts and answers a voice call. The user, which receives the call, has two or more protocol clients that are signed in (Multiple Points Of Presence).

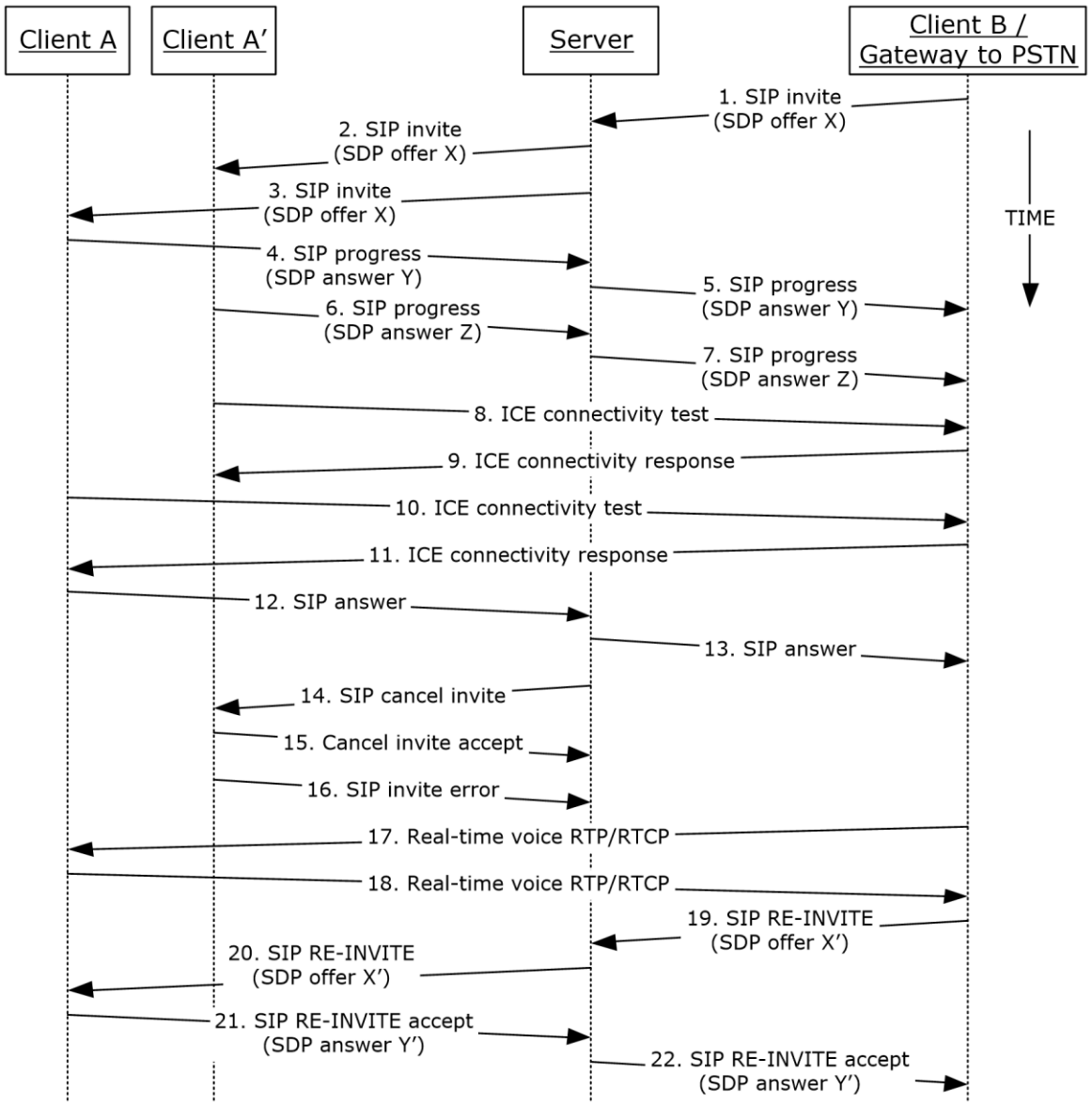

Figure 18: Steps for accepting a voice call

References

- [MS-AVEDGEA]
- [MS-ICE]
- [MS-ICE2]
- [MS-ICE2BWM]
- [MS-OCPSTN]
- [MS-RTP]
- [MS-RTPRADEX]

- [MS-SDPEXT]
- [MS-SIPRE]
- [MS-SRTP]
- [MS-SSRTP]
- [MS-TURN]
- [MS-TURNBWM]

Preconditions

- The protocol clients are signed in, as described in section 2.5.5.

Steps

- The steps to allocate media ports on the edge protocol server for firewall and NAT traversal are omitted for clarity. These steps are described in see [MS-AVEDGEA], [MS-TURN], and [MS-TURNBWM].

Post-conditions

- A session is established between Client A and Client B, and real-time voice packets are exchanged between Client A and Client B.

2.5.15 Terminate a Voice Call

This use case, illustrated in the following diagram, describes how a protocol client terminates a voice call.

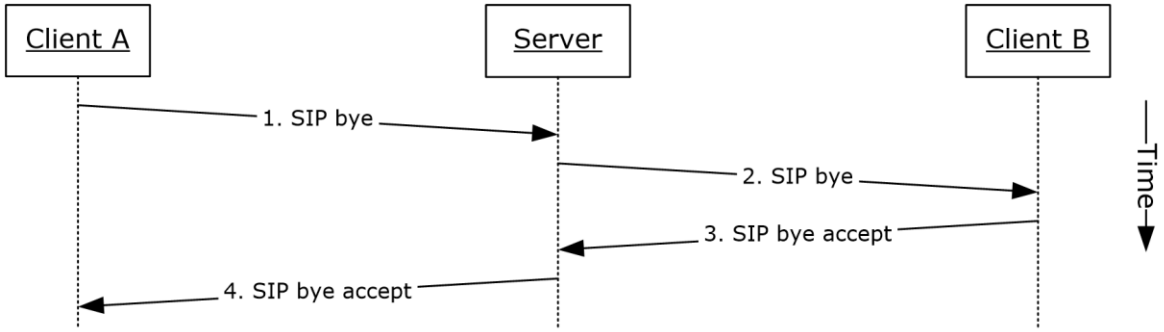

Figure 19: Steps for terminating a voice call

References

- [MS-SIPRE]

Preconditions

- The protocol clients are signed in, as described in section 2.5.5.
- Initiate a call, as described in section 2.5.12, or accept a call, as described in section 2.5.14.

Steps

1. Client A sends a SIP bye message to the protocol server to terminate the call, as described in [MS-SIPRE].

2. The protocol server forwards the SIP bye message to Client B, as described in [MS-SIPRE].

3. Client B sends a SIP response message, which accepts the request to the protocol server, as described in [MS-SIPRE].

4. The protocol server forwards the response to Client A, as described in [MS-SIPRE].

Post-conditions

- A session between Client A and the Client B is cleared and the real-time voice packets are terminated.
- If a call traverses through a media edge protocol server, the protocol clients deallocate these ports after the call is terminated.

2.5.16 Send a Quality of Experience Report

This use case, illustrated in the following diagram, describes how a protocol client sends a Quality of Experience (QoE) report after a call is terminated.

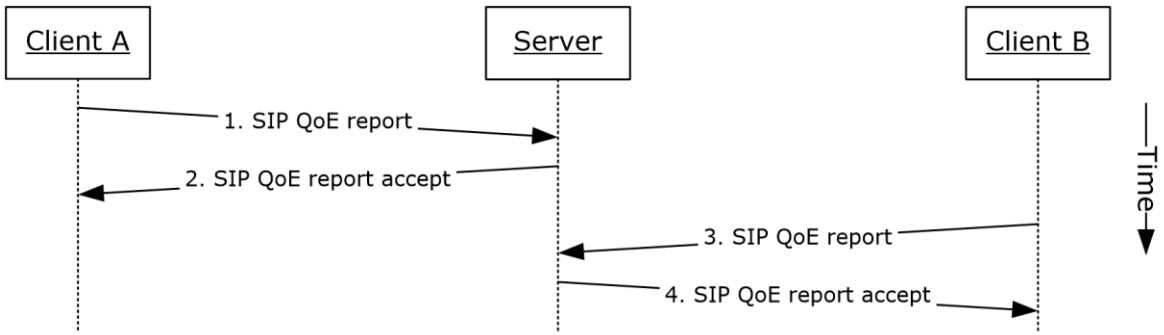

Figure 20: Steps for sending a Quality of Experience report

References

- [MS-QoE]
- [MS-SIPRE]

Preconditions

- The protocol clients are signed in, as described in section 2.5.5.
- Initiate a voice call, as described in section 2.5.12.
- Terminate a voice call, as described in section 2.5.15.

Steps

1. Client A sends a Quality of Experience report to the protocol server, as described in [MS-SIPRE] and [MS-QoE].

2. The protocol server sends an accept response to Client A.

3. Client B sends a Quality of Experience report to the protocol server, as described in [MS-SIPRE] and [MS-QoE].

4. The protocol server sends an accept response to Client B.

Note: A Quality of Experience report is sent by each protocol client independently. Each protocol client sends the report to its provisioned protocol server.

Post-conditions

- A Quality of Experience report from each protocol client is stored in the database.

2.5.17 Start and Join a Multiparty Audio Conference

This use case, illustrated in the following diagram, describes how a protocol client can start and join a multiparty audio conference. The diagram in example call flow below is composed of multiple asynchronous sub-flows, described in references after the diagram, so messages can appear in different order than in the diagram.

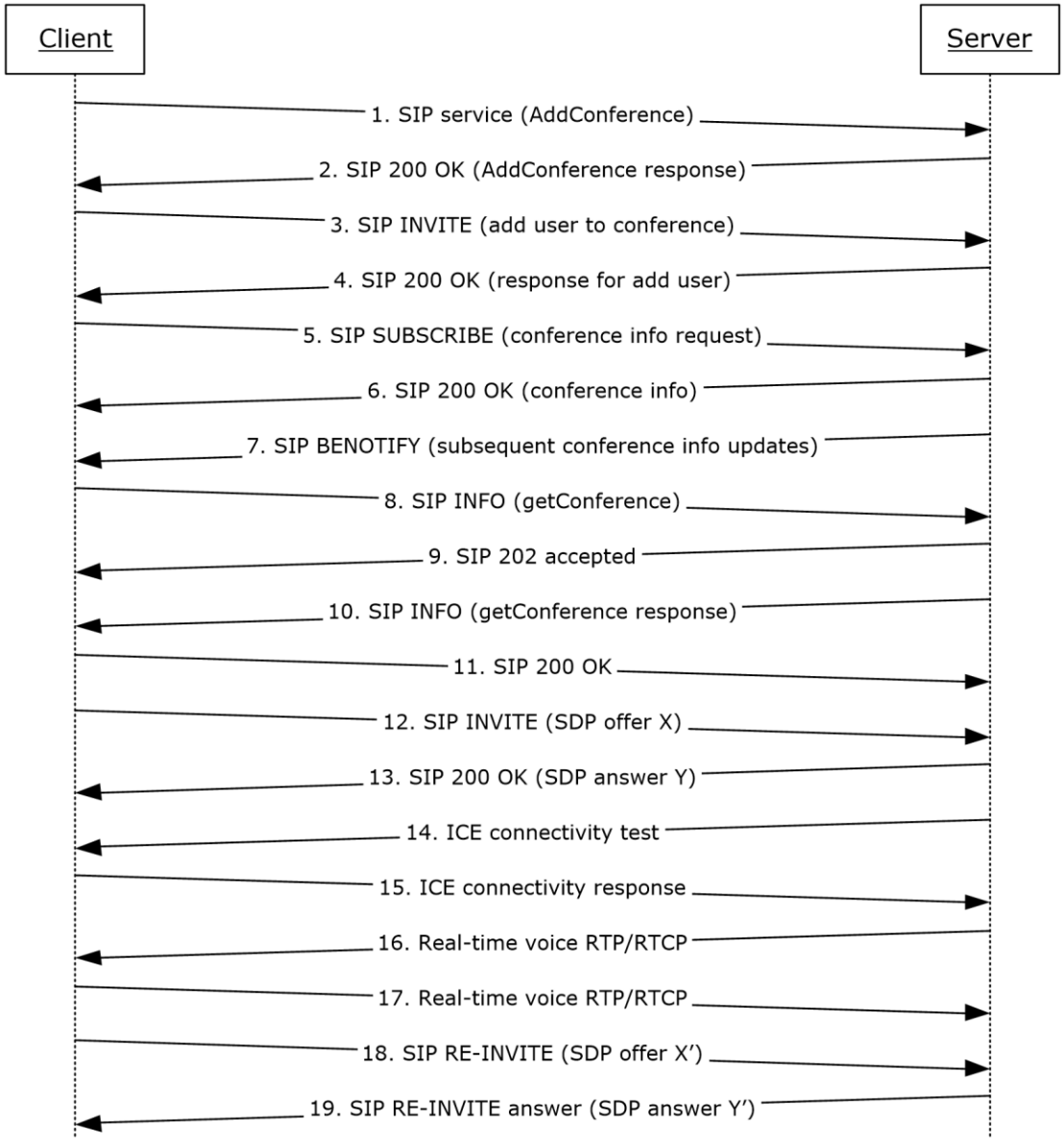

Figure 21: Steps for starting and joining a multiparty audio conference

References

- [MS-CONFPRO]
- [MS-CONFAV]
- [MS-CONFBAS]
- [MS-ICE]
- [MS-ICE2]
- [MS-ICE2BWM]
- [MS-RTP]
- [MS-RTPRADEX]
- [MS-SDPEXT]
- [MS-SIPRE]
- [MS-SRTP]
- [MS-SSRTP]

Preconditions

- The protocol client is signed in, as described in section 2.5.5.

Steps

1. The protocol client sends a request (addConference) to Communications Server to instantiate a conference, as described in [MS-CONFPRO] and [MS-SIPRE].
2. Communications Server responds with a conference URI, as described in [MS-SIPRE] and [MS-CONFPRO].
3. The protocol client sends a SIP INVITE message to Communications Server to join the conference (addUser) instantiated in step 1, as described in [MS-SIPRE] and [MS-CONFBAS].
4. Communications Server sends a SIP 200 OK response containing the join response (addUser response) to the protocol client, as described in [MS-SIPRE] and [MS-CONFBAS].
5. The protocol client sends a SIP SUBSCRIBE message to subscribe to conference information, as described in [MS-SIPRE] and [MS-CONFBAS].
6. Communications Server sends a SIP 200 OK with the conference information document and the MCU URI, as described in [MS-SIPRE], [MS-CONFAV], and [MS-CONFBAS].
7. Communications Server sends a SIP BENOTIFY to the protocol client containing subsequent roster updates, as described in [MS-SIPRE] and [MS-CONFBAS].
8. The protocol client sends a SIP INFO message with a getConference request, as described in [MS-SIPRE] and [MS-CONFPRO].
9. Communications Server sends a SIP 202 accepted response, as described in [MS-SIPRE].
10. Communications Server sends a SIP INFO message with a getConference response to the protocol client, as described in [MS-SIPRE] and [MS-CONFPRO].
11. The protocol client sends a SIP 200 OK to Communications Server, as described in [MS-SIPRE].

12. The protocol client sends a SIP INVITE with an SDP offer to Communications Server, as described in [MS-SIPRE] and [MS-SDPEXT].

13. Communications Server sends a SIP 200 OK with an SDP answer to the protocol client, as described in [MS-SIPRE] and [MS-SDPEXT].

14. Communications Server initiates ICE connectivity tests to the protocol client, as described in [MS-ICE] or [MS-ICE2]<25> and [MS-ICE2BWM].<26>

15. The protocol client sends an ICE connectivity response to Communications Server, as described in [MS-ICE] or [MS-ICE2]<27> and [MS-ICE2BWM].<28>

16. Communications Server sends RTP/RTCP voice packets to the protocol client, as described in [MS-RTP], [MS-RTPRADEX], [MS-SRTP], and [MS-SSRTP].

17. The protocol client sends RTP/RTCP voice packets to Communications Server, as described in [MS-RTP], [MS-RTPRADEX], [MS-SRTP], and [MS-SSRTP].

18. The protocol client sends an updated media description offer to Communications Server, as described in [MS-SIPRE] and [MS-SDPEXT].

19. Communications Server sends a media description answer to the protocol client, as described in [MS-SIPRE] and [MS-SDPEXT].

Note: The steps to allocate media ports on the edge protocol server for firewall and NAT traversal are omitted for clarity. These steps are described in see [MS-AVEDGEA], [MS-TURN], and [MS-TURNBWM]

Post-conditions

- A conference session is established between the protocol client and Communications Server, and real-time voice packets are exchanged between the protocol client and Communications Server. Multiple protocol clients can join the same conference.

2.5.18 Subscribe to Conference Events

This use case, illustrated in the following diagram, describes how a protocol client can subscribe to Communications Server conference events.

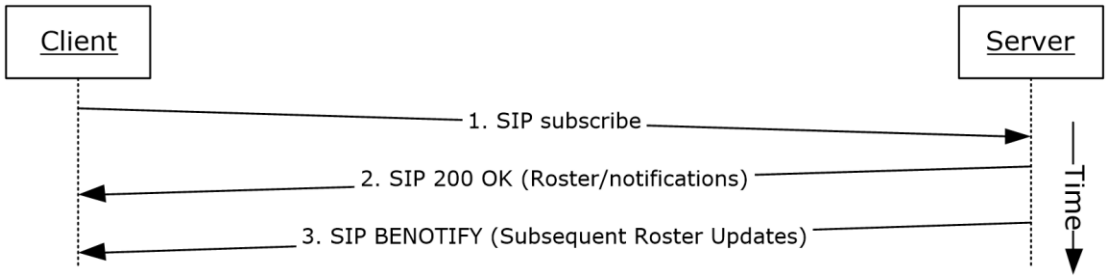

Figure 22: Steps for subscribing to conference events

References

- [MS-CONFBAS]
- [MS-SIPRE]

Preconditions

- The protocol client is present in a Communications Server conference, similar to what is described in section 2.5.17.

Steps

1. The protocol client sends a SIP subscribe message to Communications Server to subscribe to conference events, as described in [MS-SIPRE] and [MS-CONFBAS].
2. Communications Server sends a SIP 200 OK with Roster to the protocol client, as described in [MS-SIPRE] and [MS-CONFBAS].
3. Subsequent Roster Updates are sent by Communications Server to the protocol client with SIP BENOTIFY, as described in [MS-SIPRE] and [MS-CONFBAS].

Post-conditions

- The protocol client gets notifications from Communications Server every time the conference state changes.

2.5.19 Share a Desktop

This use case, illustrated in the following diagram, describes how a protocol client can share a desktop in a conference.

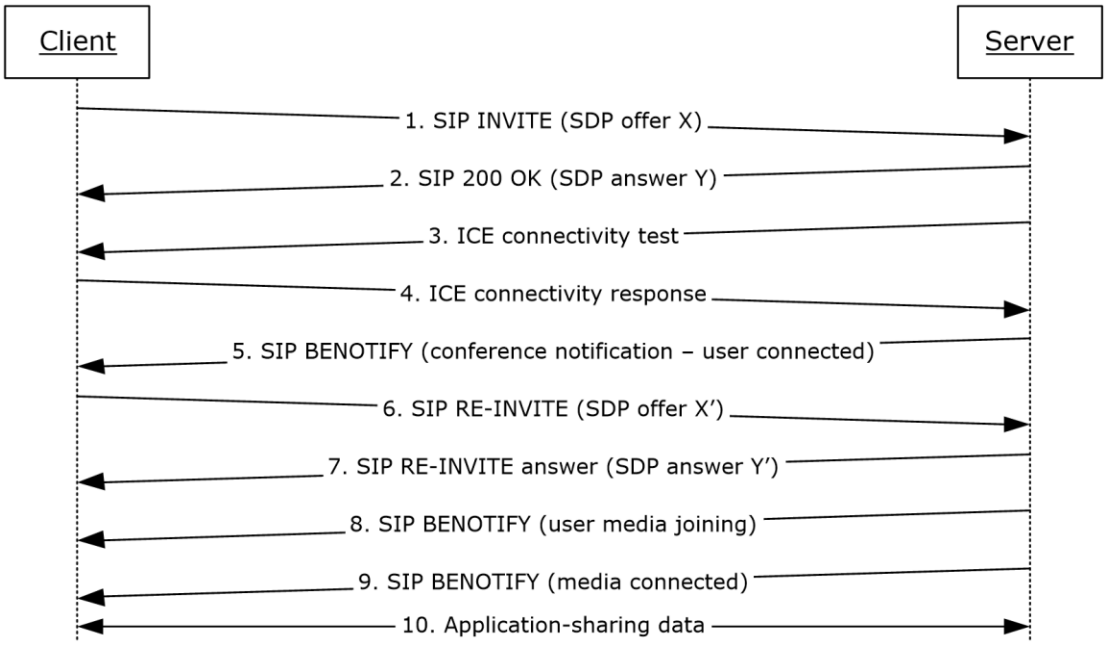

Figure 23: Steps for sharing a desktop

References

- [MS-CONFAS]
- [MS-CONFBAS]
- [MS-ICE2]
- [MS-ICE2BWM]

- [MS-RTASPF]
- [MS-RTP]
- [MS-RTPRADEX]
- [MS-SRTP]
- [MS-SSRTP]
- [MS-SIPRE]
- [MS-SDPEXT]

Preconditions

- The protocol client is present in a Communications Server conference, similar to what is described in section 2.5.17, and is subscribed to conference events, as described in section 2.5.18.

Steps

1. The protocol client sends a SIP INVITE with a media description offer to Communications Server to start an application-sharing session, as described in [MS-SIPRE], [MS-SDPEXT], and [MS-CONFAS].
2. Communications Server sends a SIP RESPONSE to the protocol client with the media description answer, as described in [MS-SIPRE], [MS-SDPEXT], and [MS-CONFAS].
3. Communications Server initiates ICE connectivity tests to the protocol client, as described in [MS-ICE2] and [MS-ICE2BWM].<29>
4. The protocol client sends an ICE connectivity response to Communications Server, as described in [MS-ICE2] and [MS-ICE2BWM].<30>
5. Communications Server sends the conference state change notifications to the client, as described in [MS-SIPRE] and [MS-CONFBAS].
6. The protocol client sends an updated media description offer to Communications Server, as described in [MS-SIPRE], [MS-SDPEXT], and [MS-CONFAS].
7. Communications Server sends the protocol client an updated media description answer, as described in [MS-SIPRE], [MS-SDPEXT], and [MS-CONFAS].
8. Communications Server sends the conference state change notifications regarding user media joining to the protocol client, as described [MS-SIPRE] and [MS-CONFBAS].
9. Communications Server sends a notification to the protocol client that the user's application-sharing media is connected, as described in [MS-SIPRE] and [MS-CONFBAS].
10. Application-sharing data flows between the protocol client and Communications Server, as described in [MS-RTP], [MS-RTPRADEX], [MS-SRTP], [MS-RTASPF], and [MS-SSRTP].

Note: The steps to allocate media ports on the edge protocol server for firewall and NAT traversal are omitted for clarity. These steps are described in see [MS-AVEDGEA], [MS-TURN], and [MS-TURNBWM]

Post-conditions

- Application-sharing data is flowing between the protocol client and Communications Server.

2.5.20 Share a Whiteboard

This use case, illustrated in the following diagram, describes how a protocol client can share a whiteboard in a conference. This section follows the behavior as described in product behavior note.<31>

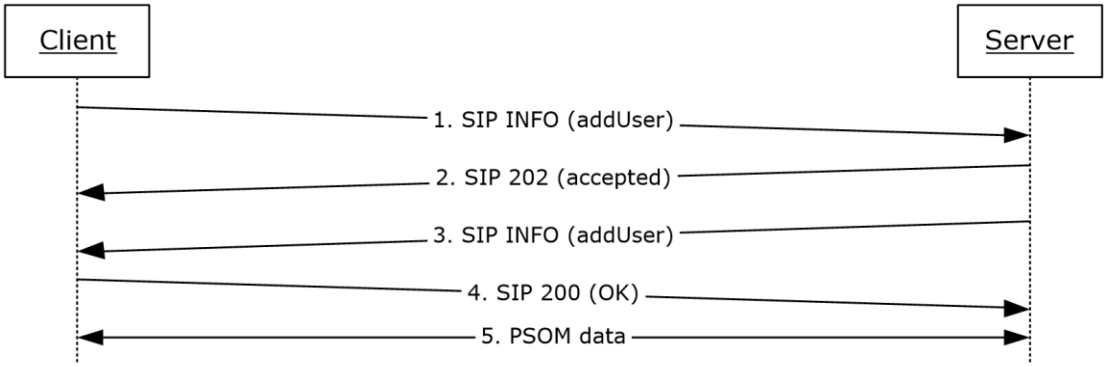

Figure 24: Steps for sharing a whiteboard

References

- [MS-CONFBAS]
- [MS-PSOM]

Preconditions

- The protocol client is present in a Communications Server conference, similar to what is described in section 2.5.17, and is subscribed to conference events, as described in section 2.5.18.

Steps

1. The protocol client sends a SIP INFO addUser request to the protocol server to start a data-conferencing session with Communications Server, as described in [MS-CONFBAS].
2. Communications Server accepts the incoming message with a SIP 202 acknowledgement.
3. Communications Server sends a SIP INFO addUser response to the protocol client with response details, as described in [MS-CONFBAS]. These details allow the protocol client to connect and authenticate to the PSOM Shared Object Messaging (PSOM) media session.
4. The protocol client sends a SIP 200 OK message to Communications Server to acknowledge successful dialog communication.
5. The protocol client connects and authenticates to Communications Server with a PSOM data session, as described in [MS-PSOM]. The protocol client then adds a whiteboard.

Post-conditions

- PSOM data is flowing between the protocol client and Communications Server.

2.5.21 Join a Chat Room

This use case, illustrated in the following diagram, describes how a protocol client can join a group chat room and post a chat message<32>.

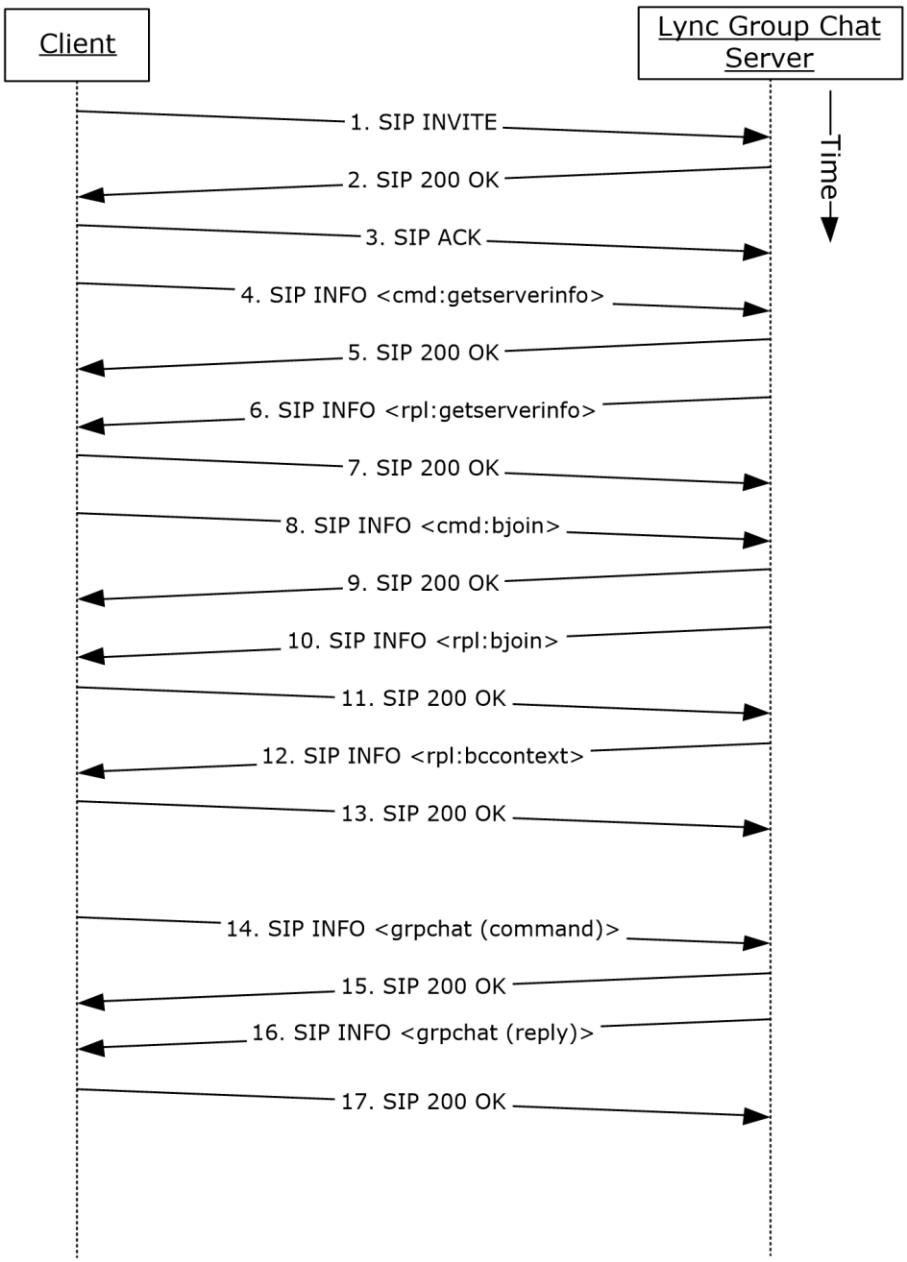

Figure 25: Steps for joining a chat room

References

- [MS-XCCOSIP]

Preconditions

- The protocol clients are signed in, as described in section 2.5.5.

Steps

1. The protocol client sends a SIP INVITE request to the protocol server to start a group chat session with protocol server.

2. Group Chat Server accepts the incoming message with a SIP 200 response.

3. The protocol client sends a SIP acknowledgment (ACK) message; the SIP dialog is now established.

4. The protocol client requests the protocol server information by sending a "cmd:getserverinfo" command over a SIP INFO message, as described in [MS-XCCOSIP].

5. The protocol server acknowledges the message from previous step with a SIP 200 response.

6. The protocol server replies with a "rpl:getserverinfo" reply over a SIP INFO message, as described in [MS-XCCOSIP].

7. The protocol client acknowledges the message from the previous step with a SIP 200 response.

8. The protocol client requests joining a set of chat rooms by sending a "cmd:bjoin" command over a SIP INFO message, as described in [MS-XCCOSIP].

9. The protocol server acknowledges the message from previous step with a SIP 200 response.

10. The protocol server replies with a "rpl:bjoin" reply over a SIP INFO message, as described in [MS-XCCOSIP]. The chat session is now established.

11. The protocol client acknowledges the message from the previous step with a SIP 200 response.

12. The protocol server also sends the historic backchat to the protocol client by sending a "rpl:bccontext" reply over a SIP INFO message, as described in [MS-XCCOSIP].

13. The protocol client acknowledges the message from the previous step with a SIP 200 response.

14. The protocol client can post a chat to the room by sending a "grpchat" command over a SIP INFO message, as described in [MS-XCCOSIP].

15. The protocol server acknowledges the message from previous step with a SIP 200 response.

16. The protocol server replies with a "grpchat" reply over a SIP INFO message, as described in [MS-XCCOSIP].

17. The protocol client acknowledges the message from the previous step with a SIP 200 response.

Post-conditions

- Protocol client has an established chat session that can be sued for transferring messages, notification of activity (like chat participants joined/left), invitation status, and so on.

2.6 Versioning, Capability Negotiation, and Extensibility

2.6.1 Versioning

Communications Server provides the capability for IT administrators to control which versions of Office Communicator can be used to can sign in. This control is enforced by checking the UserAgent attribute in the SIP header sent by the protocol clients. If the UserAgent header does not meet the minimum version number required, Communications Server rejects the protocol client’s sign-in request for the user.

2.6.2 Extensibility

Communications Server provides enhanced presence, which makes it possible for IT administrators to extend the presence model with additional information about users. Such extensions can be publishing location coordinates (GPS), displaying cellphone status, and so on. Exposing these additional presence extensions requires modifying the protocol clients to be aware of them and to appropriately display them in the user interface (UI).

2.7 Error Handling

Many error conditions can occur with Communications Server. Also, Communications Server is dependent on the successful operation of other systems. An exhaustive list of possible failures is not within the scope of this overview document; however, the following list provides categories where error conditions can occur.

Possible failures caused by dependent systems:

- Unavailability of a local Global Catalog (GC) server for Active Directory requests.
- DNS is not properly resolving protocol server fully qualified domain name (FQDN) or SRV records are not configured properly.
- The certification authority (CA) that is used to issue certificates to Communications Server is not configured to be trusted by a protocol server or protocol client.
- The hardware load balancer is not properly configured.
- The SIP/PSTN gateway is not properly configured.
- The RCC gateway is not properly configured.
- Internal firewalls are blocking connectivity between protocol servers or between Communications Server and dependent systems such as GCs and CAs.
- The reverse proxy is not properly configured.
- A protocol server that is not running Communications Server is using incompatible protocols and extensions.

Possible failures caused by Communications Server:

- An improperly configured or missing certificate.
- Dependent components or redistributables are not installed.
- Users are not configured for Unified Communications.
- Administrative settings are not configured.
- Incompatible protocols and extensions are being used by protocol clients.

For more information about these errors, see [MS-OCER].

2.8 Coherency Requirements

This system has no special coherency requirements.

2.9 Security

Security is a necessary feature of the system of protocols described by this overview. These protocols enable an Internet-based collaboration system whose requirements include protocol-level security.

2.9.1 Protocol Security

This system of protocols builds upon the security features designed in SIP to address the concerns described in [RFC3261] section 26. The following sections describe the security-related features of this system of protocols.

2.9.1.1 Audio Video Edge Authentication Protocol

The Audio Video Edge Authentication protocol, as described in [MS-AVEDGEA], describes a protocol used by protocol clients to get security tokens needed to authenticate themselves with a protocol server that implements the Traversal Using Relay NAT (TURN) Extensions protocol, as described in [MS-TURN].

2.9.1.2 Distribution List Expansion Protocol

The Distribution List Expansion protocol, as described in [MS-DLX], indicates that HTTP connections to a distribution list expansion server can only be made over **Secure Sockets Layer (SSL)**. Users are authenticated using Kerberos V5 and NT LAN Manager (NTLM) Authentication Protocol, as described in [MS-NLMP], authentication methods.

2.9.1.3 Interactive Connectivity Establishment (ICE) Extensions Protocol

The Interactive Connectivity Establishment (ICE) Extensions protocol, as described in [MS-ICE], describes mitigations used to defeat several kinds of attacks such as attacks on address gathering, attacks on connectivity checks, voice amplification attacks, and Simple Traversal of UDP through NAT (STUN) amplification attacks.

2.9.1.4 Client Error Reporting Protocol

The Client Error Reporting protocol, as described in [MS-OCER], suggests that an implementer remove the Microsoft diagnostics header from SIP responses sent to users outside an enterprise, because the header can contain private or sensitive enterprise information.

2.9.1.5 Session Description Protocol (SDP) Version 2.0 Protocol Extensions

The Session Description Protocol (SDP) Version 2.0 Protocol Extensions, as described in [MS-SDPEXT], suggest that an implementer exchange media encryption information within SIP traffic over a TLS connection. The Session Description Protocol (SDP) Version 2.0 Protocol Extensions do not describe the encryption of the media encryption information.

2.9.1.6 Secure Real-time Transport Protocol (SRTP) Extensions

The Secure Real-time Transport Protocol (SRTP) Extensions, as described in [MS-SRTP], describe the generation and exchange of random master keys between SIP entities that send/receive the RTP. All necessary aspects of the exchanged master key are described within the Secure Real-time Transport Protocol (SRTP) Extensions and the Scale Secure Real-time Transport Protocol (SSRTP) Extensions, as described in [MS-SSRTP].

2.9.1.7 Traversal Using Relay NAT (TURN) Extensions

The Traversal Using Relay NAT (TURN) Extensions, as described in [MS-TURN], describe the usage of a long-term credential in a digest challenge/response exchange.

2.10 Additional Considerations

There are no additional considerations.

3 Examples

The examples in sections 3.1 through 3.6 extend the section 2.5 use cases by illustrating how one or more use cases can be combined to achieve specific results for users. This document provides the following examples:

- Send an instant message to a contact.
- Make a call from Office Communicator.
- Accept an inbound call to Office Communicator.
- Add video to a voice call from Office Communicator.
- Start a conference, join with multiparty audio, and start application-sharing.
- Get current location and publish presence.

Unlike the use cases sections, which focus on abstract communications between general entities (for example, protocol server and protocol client), these sections discuss details that are specific to a Communications Server implementation. The examples help explain how the use cases can be applied to specific messaging tasks that map to typical user scenarios. These examples are not meant to be exhaustive. However, they can be easily applied to other similar scenarios. The protocol-level examples can be found in the individual protocol documents.

3.1 Example 1: Send an Instant Message to a Contact

This example illustrates how a series of use cases can be used to initiate IM from Office Communicator to a contact.

Use Cases

- Sign in, as described in section 2.5.5.
- Download Address Book and Search in section 2.5.7.
- Add a contact, as described in section 2.5.10.
- Initiate IM, as described in section 2.5.9.
- Initiate IM, also covers the Multiple Endpoints in section 2.5.11. The IM is accepted by one of the multiple destination endpoints of a given user.

Details

The following diagram illustrates how the actors (user and administrator) interact with the use cases that are used as building blocks for this example.

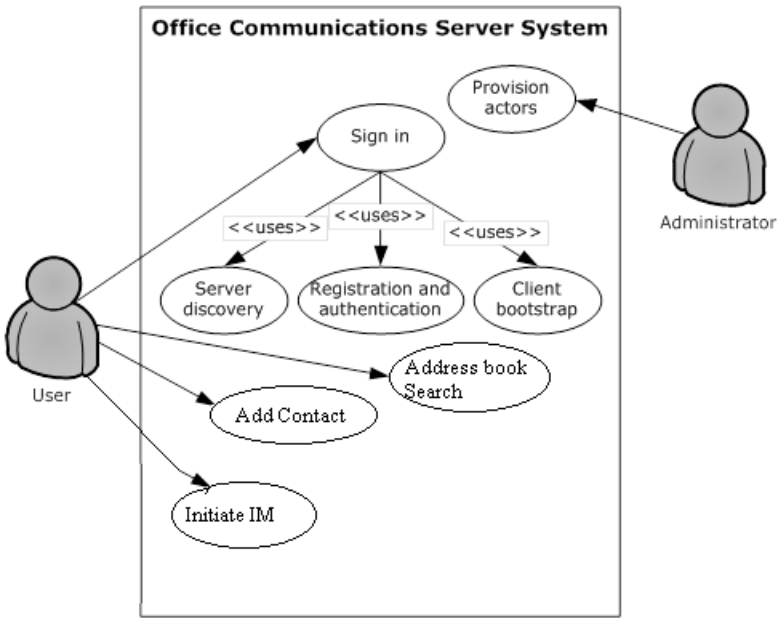

Figure 26: Process for sending an instant message to a contact

The following diagram illustrates the sequence in which the use cases are invoked in this example.

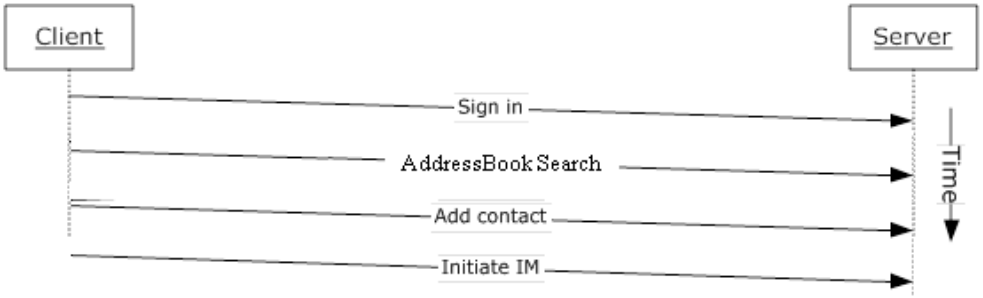

Figure 27: Sequence for sending an instant message to a contact

3.2 Example 2: Make a Call from Office Communicator

This example illustrates how a series of use cases can be used to make a voice call from Office Communicator.

Use Cases

- Sign in, as described in section 2.5.5.
- Initiate a call from a client, as described in section 2.5.12.
- Terminate a voice call, as described in section 2.5.15.
- Send a Quality of Experience report, as described in section 2.5.16.

Details

The following diagram illustrates how the actors (user and administrator) interact with the use cases that are used as building blocks for this example.

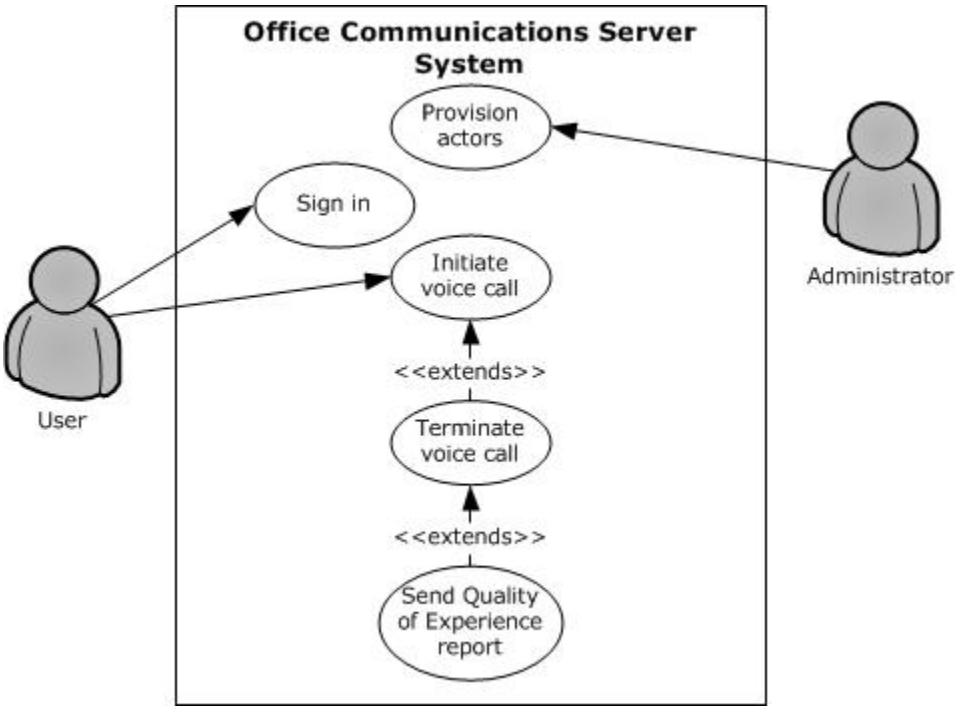

Figure 28: Process for making a call from Office Communicator

The following diagram illustrates the sequence in which the use cases are invoked in this example.

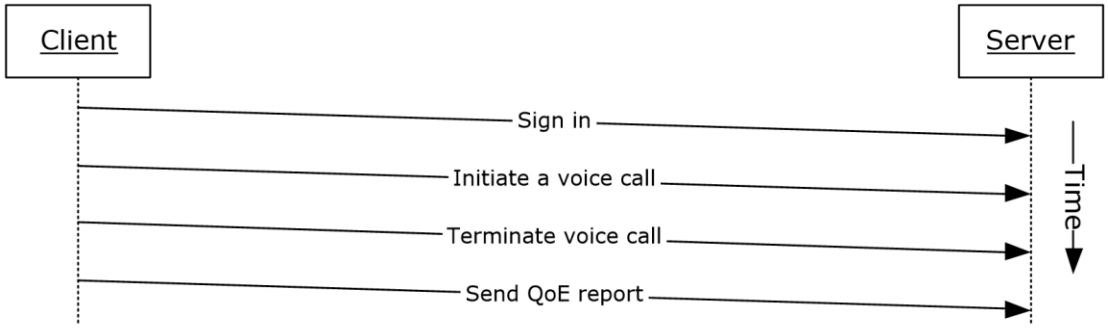

Figure 29: Sequence for making a call from Office Communicator

3.3 Example 3: Accept an Inbound Call to Office Communicator

This example illustrates how a series of use cases can be used to accept and answer a voice call from Office Communicator.

Use Cases

- Sign in, as described in section 2.5.5.
- Accept a voice call, as described in section 2.5.14.
- Terminate a voice call, as described in section 2.5.15.

- Send a Quality of Experience report, as described in section 2.5.16.

Details

The following diagram illustrates how the actors (user and administrator) interact with the use cases that are used as building blocks for this example.

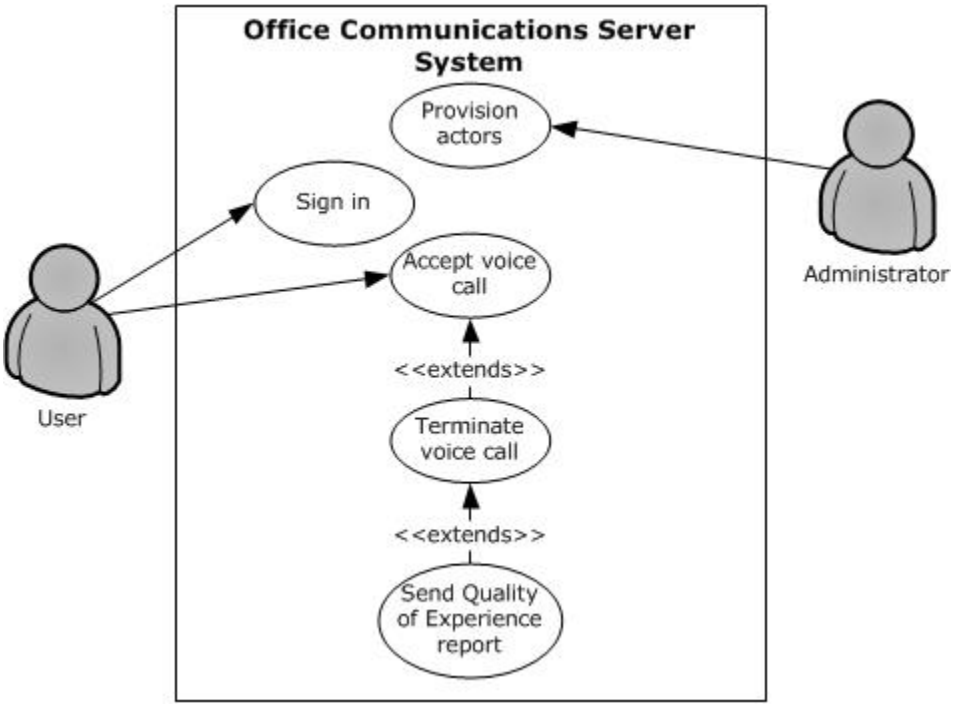

Figure 30: Process for accepting an inbound call to Office Communicator

The following diagram illustrates the sequence in which the use cases are invoked in this example.

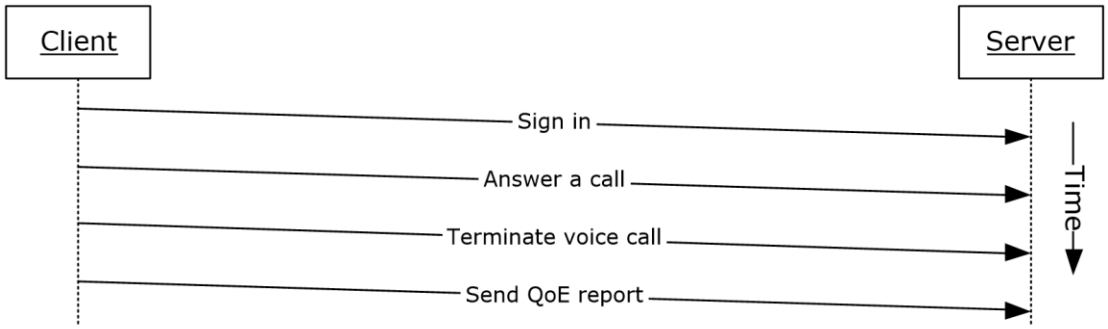

Figure 31: Sequence for accepting an inbound call to Office Communicator

3.4 Example 4: Add Video to a Voice Call from Office Communicator

This example illustrates how a series of use cases can be used to add video to a voice call from Office Communicator.

Use Cases

- Sign in, as described in section 2.5.5.

- Initiate a call from a Client, as described in section 2.5.12.
- Add video to a voice call, as described in section 2.5.13.
- Terminate a voice call, as described in section 2.5.15.
- Send a Quality of Experience report, as described in section 2.5.16.

Details

The following diagram illustrates how the actors (user and administrator) interact with the use cases that are used as building blocks for this example.

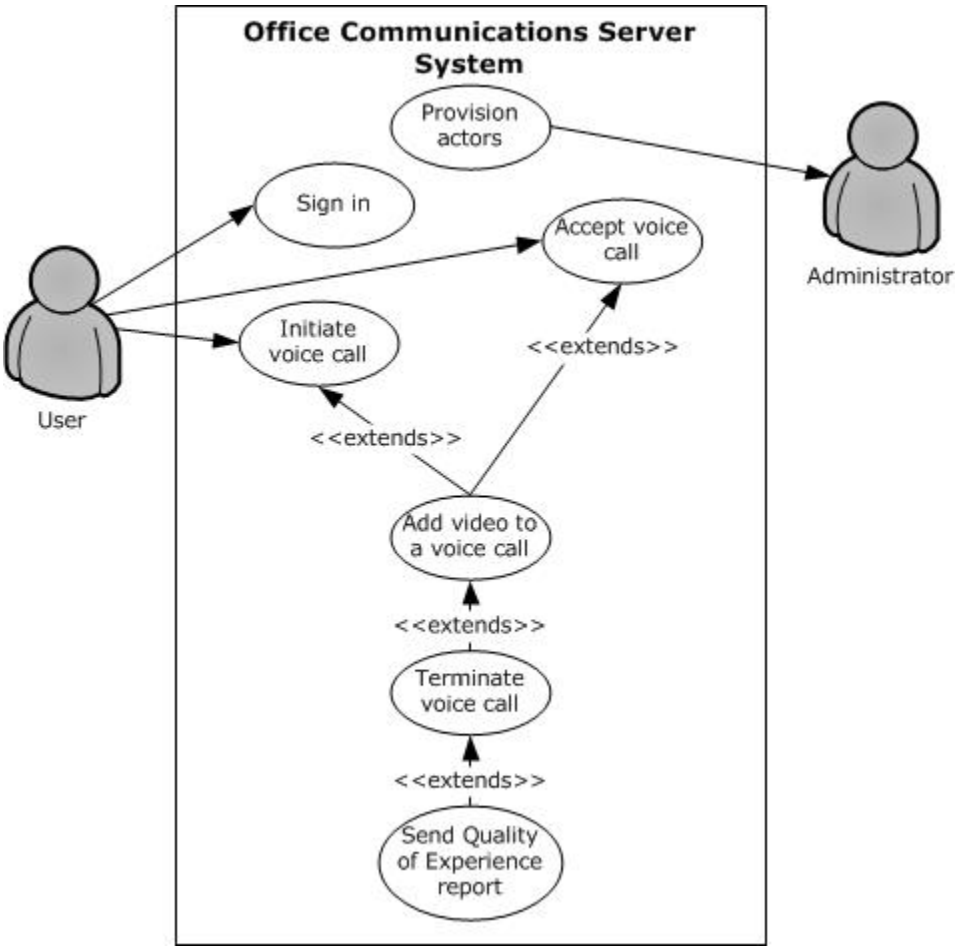

Figure 32: Process for adding video to a voice call from Office Communicator

The following diagram illustrates the sequence in which the use cases are invoked in this example.

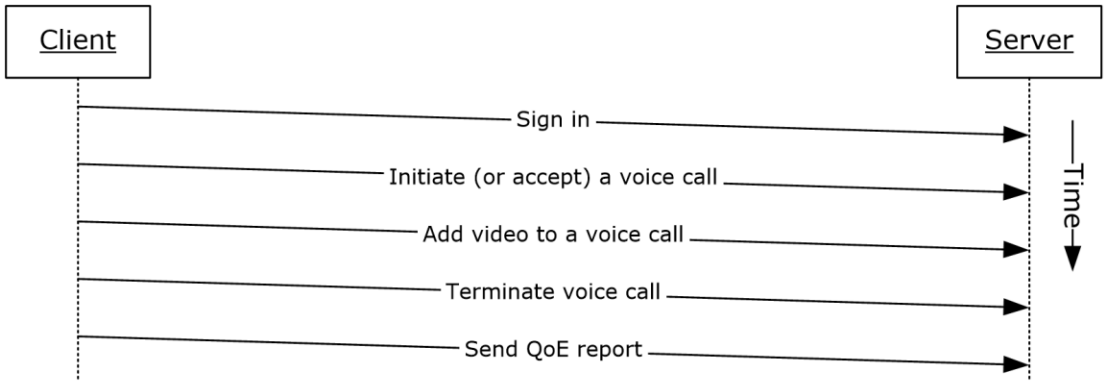

Figure 33: Sequence for adding video to a voice call from Office Communicator

3.5 Example 5: Start a Conference, Join with Multiparty Audio, and Start Application-Sharing

This example illustrates how a series of use cases can be used to start and join a multiparty conference with audio, and start application-sharing using Office Communicator.

Use Cases

- Sign in, as described in section 2.5.5.
- Start and join a multiparty audio conference, as described in section 2.5.17.
- Subscribe to conference events, as described in section 2.5.18.
- Share a desktop, as described in section 2.5.19.
- Distribution List (DL) Expansion to add more users to the conference, as described in section 2.5.8.

Details

The following diagram illustrates how the actors (user and administrator) interact with the use cases that are used as building blocks for this example.

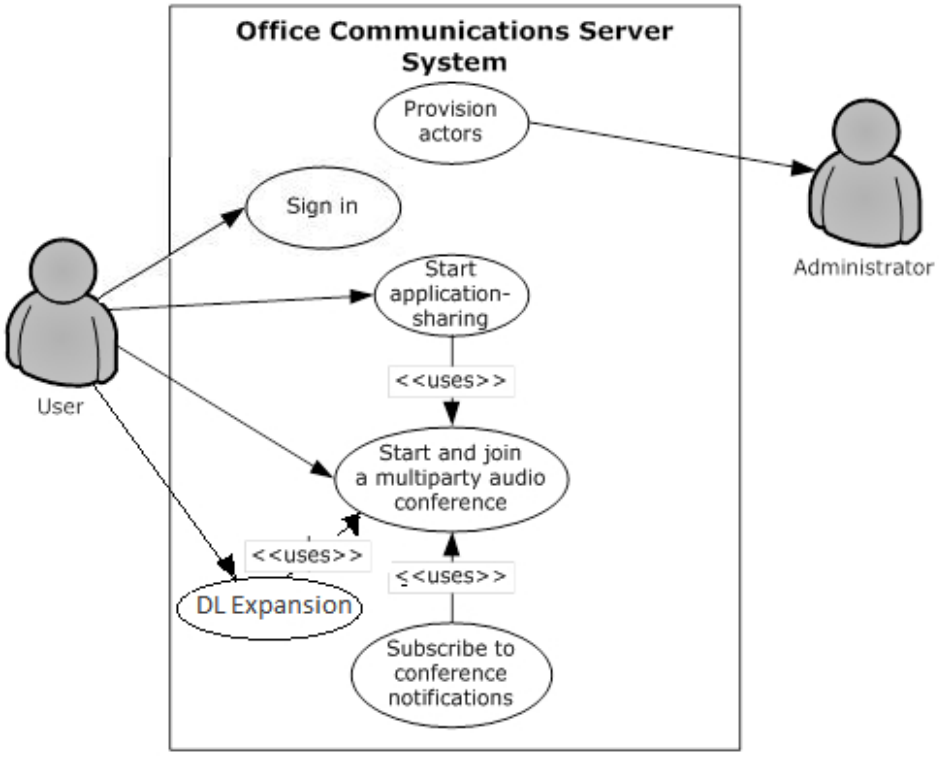

Figure 34: Process for starting a conference, joining with multiparty audio, starting application-sharing, Distribution list expansion to add more users to the conference

The following diagram illustrates the sequence in which the use cases are invoked in this example.

Figure 35: Sequence for starting a conference, joining with multiparty audio, and starting application-sharing, Distribution List (DL) Expansion

3.6 Example 6: Get Current Location, Publish presence

This example illustrates use cases getting current location, and publishing/changing presence.

Use Cases

- Sign in, as described in section 2.5.5.
- Get Current Address Location, as described in section 2.5.4.
- Publish location in presence information as described in section 2.5.6.

Details

The following diagram illustrates how the actors (user and administrator) interact with the use cases that are used as building blocks for this example.

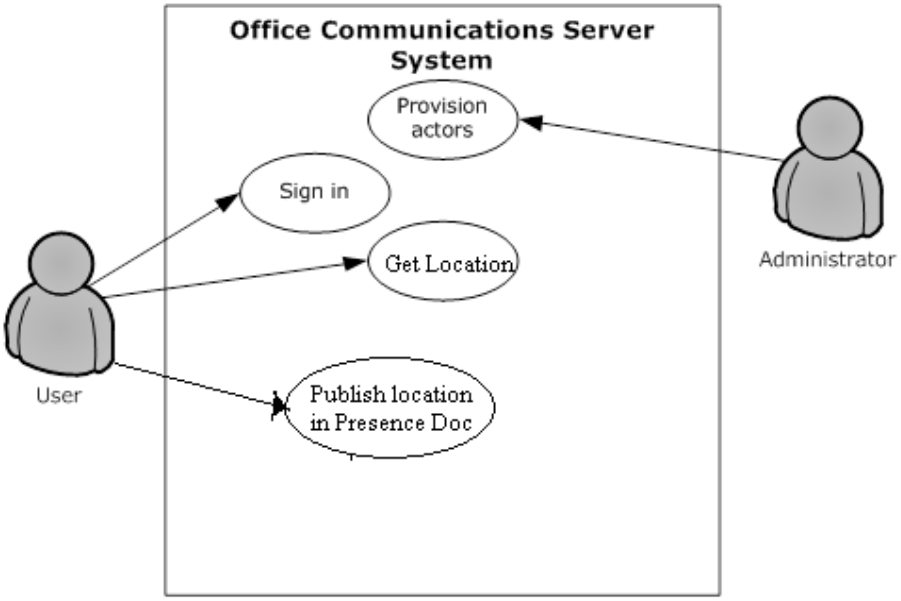

Figure 30: Process for getting location and changing presence.

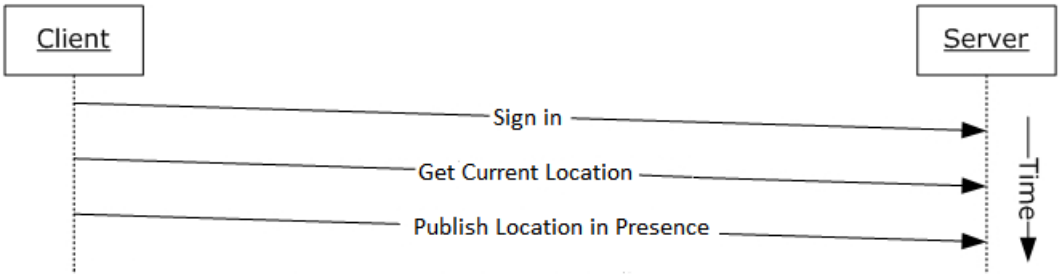

Figure 31: Sequence for getting current location and publishing location in to presence document.

4 Microsoft Implementations

There are no variations in the behavior of the Office Communications Server system in different versions of Microsoft® Communications Server and Microsoft Communicator beyond those described in the specifications of the protocols supported by the system, as listed in section 2.2.

The information in this specification is applicable to the following versions of Microsoft Office Communications Server, Microsoft Communications Server, Microsoft Office Communicator, Microsoft Communicator, Microsoft Lync, and Microsoft Lync Server:

- Microsoft Office Communications Server 2007
- Microsoft Office Communicator 2007
- Microsoft Office Communications Server 2007 R2
- Microsoft Office Communicator 2007 R2
- Microsoft Lync Server 2010
- Microsoft Lync 2010
- Microsoft Lync Server 2013
- Microsoft Lync Client 2013/Skype for Business
- Microsoft Skype for Business 2016
- Microsoft Skype for Business Server 2015

Exceptions, if any, are noted in the following section.

4.1 Product Behavior

<1> Section 2.2.2.1: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<2> Section 2.2.2.1: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<3> Section 2.2.2.1: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<4> Section 2.2.2.1: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<5> Section 2.2.2.1: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<6> Section 2.2.2.1: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2, Lync Server 2010, Lync 2010: This behavior is not supported.

<7> Section 2.2.2.1: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2, Lync Server 2010, Lync 2010: This behavior is not supported.

<8> Section 2.2.2.2: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<9> Section 2.2.3.1: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<10> Section 2.2.3.2: Office Communications Server 2007, Office Communicator 2007: This protocol is not supported.

<11> Section 2.2.3.2: Office Communications Server 2007, Office Communicator 2007: This protocol is not supported.

<12> Section 2.2.3.2: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This protocol is not supported.

<13> Section 2.2.3.2: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This protocol is not supported.

<14> Section 2.2.3.2: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This protocol is not supported.

<15> Section 2.2.3.2: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This protocol is not supported.

<16> Section 2.5.4: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<17> Section 2.5.12: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<18> Section 2.5.12: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<19> Section 2.5.12: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<20> Section 2.5.12: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<21> Section 2.5.12: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<22> Section 2.5.12: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<23> Section 2.5.12: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<24> Section 2.5.12: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<25> Section 2.5.17: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<26> Section 2.5.17: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<27> Section 2.5.17: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<28> Section 2.5.17: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<29> Section 2.5.19: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<30> Section 2.5.19: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<31> Section 2.5.20: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<32> Section 2.5.21: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2, Lync Server 2010, Lync 2010: This behavior is not supported.

5 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

6 Index

A

C

D

E

F

G

H

I

J

M

O

P

R

S

T

U

V