Music-On-Hold / Park Server
User Documentation

Author

Scott Godin – SIP Spectrum Inc.

Revision History

January 5, 2011 – Created Original Version
January 12, 2011 – added Performance section

Table of Contents

Introduction

The MOHParkServer is designed to allow SIP user agents to provide music on hold and call park / retrieve capabilities. Although these are two distinct capabilities they are very similar in implementation and therefor are both provided in the same server.

Music-On-Hold (MOH) Server

It is recommended that SIP user agents use the MOH call flow that is specified in RFC 5359 – Session Initiation Protocol Service Examples (http://tools.ietf.org/html/rfc5359). RFC 5359 specifies that a REFER message is sent to the MOH server to request that it "pick up" the call on-hold and play music to it by sending an INVITE message with a Replaces header towards the held party.

The MOH server is also capable of accepting direct inbound calls and playing music to them. This mode can be useful for endpoints that do not implement the REFER music-on-hold mechanism described in RFC 5359 (for example SNOM SIP phones).

The music source is specified in the configuration as a WAV file. Future versions of this server may allow for more complex configuration of music sources and schedules.

The music on hold server is capable of registering a music-on-hold user with a SIP proxy, in order to facilitate easier routing of MOH requests from SIP user agents in a network. If registration is not used, then SIP user agents must be configured with the appropriate machine name or IP address of the MOHParkServer, and this configuration must match the MOHUri configuration setting.

Park / Retrieve Server

The Park mechanism implemented is consistent with that specified in RFC 5359 – Session Initiation Protocol Service Examples (http://tools.ietf.org/html/rfc5359). An extension to allow park orbits that are either selected by the parking user agent or by the park server has also been implemented. Orbit assignment uses the guidelines discussed in the following IETF document: Implementing Call Park and Retrieve using the Session Initiation Protocol (http://tools.ietf.org/id/draft-procter-bliss-call-park-extension-05.html). This document discusses using a SIP URI parameter called orbit in the To header of the REFER request used to park a call (ie. sip:park@mohparkserver.com;orbit=6000).

The park server is capable of registering a park user with a SIP proxy, in order to facilitate easier routing of park requests from SIP user agents in a network. If registration is not used, then SIP user agents must be configured with the appropriate machine name or IP address of the MOHParkServer, and this configuration must match the ParkUri configuration setting. The park server is also capable of registering the range of park orbit numbers that are configured, in order to facilitate easier of routing of park retrieval requests from SIP user agents in a network. If registration of orbit numbers is not used,

then similar restrictions apply. The passwords on the SIP proxy for the range of orbit numbers must all be the same, since only one may be configured at this time.

The park server can be configured to automatically ring back the party that performed the park, if the call is not retrieved within a configurable amount of time.

Park Procedure

Send a REFER request to the Park Server URI

- If the To header of the REFER request contains an orbit URI parameter specifying a valid orbit number, then the call is parked at the specified orbit. If a call is already present at the specified orbit, then this call is "queued" behind the existing call(s).
- If the To header of the REFER request does not contain an orbit URI parameter, then a free orbit is selected by the server and a 302 response is returned to the REFER request. The 302 response will contain a contact header specifying an orbit parameter that the server selected. The "parker" should then send a new refer containing an orbit parameter in the To header that matches the one provided in the Contact header of the REFER/302 response.

The Park server also supports parking of calls directly to orbits by accepting transferred calls on the orbit URIs. The To header of INVITE request can take one of the following two forms:

1. sip:<orbit>@<hostname> ie. sip:6000@server.com
2. sip:<configuredparkUri>;orbit=<orbit> ie. sip:park@server.com;orbit=6000

The Park server will look in an incoming INVITE for a Referred-By header, if present, then the call will not be treated as a retrieval call, it will instead be parked. Thus endpoints that do not support the above procedure, can still park calls using an unattended or blind transfer feature of the device. In this case the parker must choose an orbit to park on. If an orbit is selected that already contains a parked call, the newly parked call will be “queued” behind the existing one.

Retrieve Procedure

Call the orbit number where a call is parked. The To header of INVITE request can take one of the following two forms:

1. sip:<orbit>@<hostname> ie. sip:6000@server.com
2. sip:<configuredparkUri>;orbit=<orbit> ie. sip:park@server.com;orbit=6000

The park server will process the INVITE as follow:

- If there is a user currently parked in this orbit, then the park server will answer the inbound call then immediately REFER the retriever to the parked party to “connect” them.

- If no call is parked at the called orbit, then a 404 response will be returned.

Note: RFC5359 describes that a retriever should subscribe to the dialog event package on the park server to retrieve the dialog information of the parked calls. It can then use this information to send an INVITE request with a Replaces header directly to the parked party for retrieval. This mechanism of retrieval is currently not implemented.

Execution Requirements

The sipXtapi codec plug-in dlls/libraries required, must be present in the application directory. After compilation these can be found in the sipXmediaLib\bin directory. For example: codec_pcmapcmu.dll.

A properly provisioned configuration must be specified in a file called MOHParkServer.config in the application directory.

If TLS is used, then ca.pem must be present in the application directory and contain valid domain keys.

Configuration

Command Line Options

The following command line options are available.

-a <IP Address> - bind SIP transports to this IP address
-d <DNS servers> - comma separated list of DNS servers, overrides OS detected list
-up <port num> - local port number to use for UDP SIP messaging
-tp <port num> - local port number to use for TCP SIP messaging
-sp <port num> - local port number to use for TLS SIP messaging
-td <domain name> - domain name to use for TLS server connections
-nk - no keepalives, set this to disable sending of keepalives
-op <SIP URI> - URI of a proxy server to use a SIP outbound proxy
-l <NONE|ERR|WARNING|INFO|DEBUG|STACK> - logging level

Additionally, any of the settings from the configuration can be overridden using long format command line options. For example:

-MohRegistrationTime 3600

Configuration File

Note: The configuration is only read once at startup. The server must be restarted to apply new settings.

MOH Settings

MOHUri – The URI that SIP user agents use to communicate with the MOH server. If a display name is desired on calls placed by the MOH server then it can be provided here using the following format: “{displayName}”<sip:{user}@{domain}> Example: "Music-On-Hold"<sip:moh@server.com>

MOHRegistrationTime – The registration period, in seconds, to use when registering. Specify this setting as 0 to disable registration of the MOHUri.

MOHPassword – The password that is used if registration is enabled (ie. MOHRegistrationTime is greater than 0), and the registrar requires a password for the user specified in the MOHUri.

MOHFilename – The audio filename to play for music-on-hold. Audio files may be AU, WAV or RAW formats. Audiofiles should be 16bit mono, 8khz, PCM to avoid runtime conversion.

Park Settings

ParkUri – The URI that SIP user agents use to communicate with the Park server. If a display name is desired on calls placed by the Park server then it can be provided here using the following format: “{displayName}”<sip:{user}@{domain}> Example: "Parked"<sip:park@server.com>

ParkRegistrationTime – The registration period, in seconds, to use when registering. Specify this setting as 0 to disable registration of the ParkUri.

ParkPassword – The password that is used if registration is enabled (ie. ParkRegistrationTime is greater than 0), and the registrar requires a password for the user specified in the ParkUri.

ParkMOHFilename – The audio filename to play for music-on-hold. Audio files may be AU, WAV or RAW formats. Audiofiles should be 16bit mono, 8khz, PCM to avoid runtime conversion.

ParkOrbitRangeStart – A number specifying the starting number of the park orbit range. Orbit numbers are treated as integers, thus this number must be 9 digits or less.

ParkNumOrbits – The number of park orbits to allocate and potentially register. For example, if ParkOrbitRangeStart is specified as 6000 and ParkNumOrbits is 10, then the number range 6000-6009 will be used for park orbits.

ParkOrbitRegistrationTime - The registration period, in seconds, to use when registering the orbit retrieval identities. Specify this setting as 0 to disable registration of the park orbit URIs. Note: Park

orbit URI's are formed by taking the park orbit number as the SIP user and the domain from the ParkUri setting. For example, if the ParkUri is specified as sip:park@server.com and the ParkOrbitRangeStart is 6000, then the first park orbit Uri will be: sip:6000@server.com

ParkOrbitPassword - The password that is used if orbit registration is enabled (ie. ParkOrbitRegistrationTime is greater than 0), and the registrar requires a password for the park orbit user.

MaxParkTime – The number of seconds the park server will allow a call to stay parked. If a call is not retrieved within this time, it will ring back the party that originally parked the call. Set to 0 to disable.

SIP Settings

IPAddress - Local IP Address to bind SIP transports to. If left blank the MOHParkServer will bind to all adapters. Note: If you specify an IP address here it will not be possible to support both IPv4 and IPv6 at the same time.

DNSServers - Comma separated list of DNS servers, overrides the default OS detected list (leave blank for default DNS servers).

UDPPort - Local port to listen on for SIP messages over UDP. Specify as 0 to disable UDP.

TCPPort - Local port to listen on for SIP messages over TCP. Specify as 0 to disable TCP.

TLSPort - Local port to listen on for SIP messages over TLS. Specify as 0 to disable TLS. Note: TLS must be properly enabled at compile time if required.

TLSDomainName - TLS domain name for this server. A SIP domain cert and private key for this domain must be present in the current directory. Note: TLS must be properly enabled at compile time if required.

KeepAlives - Enable/Disable TCP/UDP CRLFCRLF keepalive packets for SIP endpoints: 1 to enable, 0 to disable

OutboundProxy - URI of a proxy server to use a SIP outbound proxy. This setting should not be required if proper DNS based SIP routing is operational.

Media Settings

MediaRelayPortRangeStart – Specifies the start of the range of UDP port numbers to be used to send RTP traffic.

MediaRelayPortRangeSize - Specifies the size of the range of UDP port numbers to be used to send RTP traffic. Calls will be rejected when the media ports are depleted. A SIP call requires 2 media ports for each call (RTP and RTCP). For example: a setting of 800 will limit the server to a maximum of 400 calls.

sipXLogFilename – the filename to use for any sipXtapi (ie. media engine) error logging

General Settings

LogLevel – Controls the amount of logs sent to the sip log file. Valid values are:
NONE|CRIT|ERR|WARNING|INFO|DEBUG|STACK

LogFilename – Filename of log file

LogFileMaxBytes – The maximum number of log bytes in the log file. After the log contains this many bytes it will start a new log file.

Performance

The MOHParkServer was performance tested under Windows XP on a Quad Core Q6600 (2.4Ghz, 4GbRAM) box, using a 32-bit Release build.

The following maximum call limitations were observed:
400 Maximum Simultaneous Music on Hold calls – CPU use ~25%, using ~400mb of memory
200 Maximum Simultaneous Park calls (each one on it's own orbit) – CPU use ~25%, using 1 Gb of memory

CPU use tops out at 25%, since the media engine runs in one thread and can only utilize 1 of the 4 cores on the quad core CPU.

Performance on the park server is lower and memory is higher, because a sipXtapi flow graph is used for every call, whereas the MOH server uses a sipXtapi flow graph for every 7th call.

Note: The define DISABLE_FLOWMANAGER_IF_NO_NAT_TRAVERSAL was enabled in recon/RemoteParticipantDialogSet.cxx in order to get improved performance results.