Click-To-Call Server
User Documentation

Author

Scott Godin – SIP Spectrum Inc.

Revision History

July 19, 2009 – Created Original Version
August 30, 2009 – Added XML interface documentation

Table of Contents

Introduction

This server offers an HTTP interface for click to call functionality.

Building the Source Code on Unix/Linux

Note: <basedir> represents a base working directory for the instructions below.

Prerequisites: The following standard packages should be installed on the system prior to the following:

- GNU C++ compiler and tools
- SVN client - http://subversion.tigris.org/
- OpenSSL 0.9.8 libraries - http://www.openssl.org/
- GNU Perfect Hash function (gperf) generator - http://www.gnu.org/software/gperf/
- Perl Compatible Regular Expressions (PCRE) library - http://www.pcre.org/

Building Resiprocate SIP Libraries

1. Go to <basedir>.
2. mkdir resip - create a project directory that will contain all the files required for the gateway project

   Note: If the following does not work - please download root certificates from www.cacerts.org or just use http instead of https. Https is only required for SVN commits.
3. svn checkout https://svn.resiprocate.org/rep/resiprocate/main resip
4. cd resip - Note: steps 4-6 can be skipped if building on Windows.
5. ./configure

   Choose all defaults, except:

   - Build Repro proxy server: no <- saves build time
   - Compile in IPv6 support: yes
6. make

Building clicktocall binary

Windows

1. Open <basedir>/resip/apps/clicktocall_9_0.sln from Visual Studio 2008.
2. Build project.

Linux

1. cd <basedir>/resip/apps/clicktocall
2. make

Configuration

Configuration File

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

SIP settings

IPAddress - Local IP Address to bind SIP transports to. If left blank the clicktocall 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).

ClickToCallIdentity - Used in From header of SIP calls, when calling initiator. Value must be a valid formatted SIP URI. Ie. sip:<user>@<domain>. Note: The Click to Call server will automatically populate the From display name with “Click-To-Call: <destuser>”.

UDPTCPPort - Local port to listen on for SIP messages over UDP or TCP

TLSPort - Local port to listen on for SIP messages over TLS

TLSDomainName - TLS domain name for this server (note: a SIP domain cert and private key for this domain must be present in the current directory)

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.

General settings

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

LogFilename – Filename of log file

LogFileMaxLines – The maximum number of log lines in the log file. After the log contains this many lines it will start a new log file. Note: 50000 lines is approximately a file size of 5Mb.

XmlRpcPort - Port to listen for Xml Rpc requests on. Set to 0 to disable. Note: clicktocall will listen on this port for all interfaces, for both IPv4 and IPv6 (if available).

HttpPort - Port to listen on for HTTP call control messaging (ie. Click-to-call). Set to 0 to disable. Note: clicktocall will listen on this port for all interfaces, for both IPv4 and IPv6 (if available).

HttpAuth - Enable/Disable HTTP Digest authentication: 1 to enable, 0 to disable

HttpAuthPwd - Password required if HttpAuth is enabled: Note: auth user is always "admin".

Address Translation Rules

Address translation rules are used for all click to call originator and destination addresses. The rules can be used to create a valid SIP URI or remap an existing URI to something else. The address is passed through these rules in order to be translated into the proper URI for SIP dialing.

TranslationPattern - Address translation patterns use Perl compatible regular expression (PCRE) syntax. See http://perldoc.perl.org/perlre.html.

TranslationOutput – If a string matches the corresponding translation pattern then it is translated using this output rule. Bracketed expressions from the pattern can be used in the output string by using a $<bracket-number> notation. For example to output the first bracketed expression from the translation patter, use $1 in the output string.

Note: TranslationPattern and TranslationOutput settings can be repeated as many times as required, and MUST always be listed in pairs. Rules are evaluated in a top down manner, and the translation output for the first matching pattern is applied.

Examples:

TranslationPattern=^(.*)$
TranslationOutput=sip:$1@sipdomain.com
Ie. 9055551234 -> sip:9055551234@sipdomain.com

Configuring ClickToCall for TLS

TODO

Certificate Configuration

Using the Click To Call Server

HTTP Interface

The ClickToCall server will listen for HTTP traffic on all interfaces (IPv4 and IPv6) on the configured port (HttpPort setting). If HTTP authentication is turned on, then a digest challenge for username and password is presented to the client. The username is hardcoded to be admin, and the password is configurable via the configuration file.

Commands available are:

Click To Call

This HTTP command causes a call to be placed first to the initiator. Once the initiator answers, the destination party is called. The HTTP interface does not provide call progress status to the caller, as long as the parameters were pass correctly it will just return a brief confirmation page that the click to call command is executing. If call progress status is required, see the XML RPC interface.

http://<server-address>:<httpPort>/clicktocall.html?initiator=*{initiator}*&destination=*{destination}*[&anchor=*{true/false}*]

The initiator and destination provided will be passed through the configured translation rules to form a valid SIP URI.

If the anchor setting is not present then "false" is the default. With anchor=false the click to call server will first call the originator, then "REFER" the answering phone to the destination. In this case, after a successful REFER, the ClickToCall server is no longer involved in the call. If the REFER request fails then the ClickToCall server will revert to using an anchored call instead. With anchor=true the click to call server will first call the originator. Once the originator answers, the server will form a new call to the destination and will direct RTP to go directly between the two endpoints. In this case the ClickToCall server is behaving as a back to back user agent.

Example:

http://192.168.1.2:5090/clicktocall.html?initiator=sip:scott@sipdomain.com&destination=9055551234

XML Interface

The ClickToCall server will listen for TCP socket connections on all interfaces (IPv4 and IPv6) on the configured port (XmlRpcPort setting). Requests are then passed to the server in XML format over the socket connection. The server will respond with an XML reply to the request. It is legal to send multiple requests on the same connection, if desired.

Note: The XML formatting used is specific to this implementation and is NOT intended to be compatible with the XML-RPC protocol defined by the following: http://www.xmlrpc.com/

Commands available are:

Click To Call

This XML interface command causes a call to be placed first to the initiator. Once the initiator answers, the destination party is called.

```
<ClickToCall>
  <Request>
    <Initiator>{initiator}</Initiator>
    <Destination>{destination}</Destination>
    <AnchorCall>{true|false}</AnchorCall>
  </Request>
</ClickToCall>
```

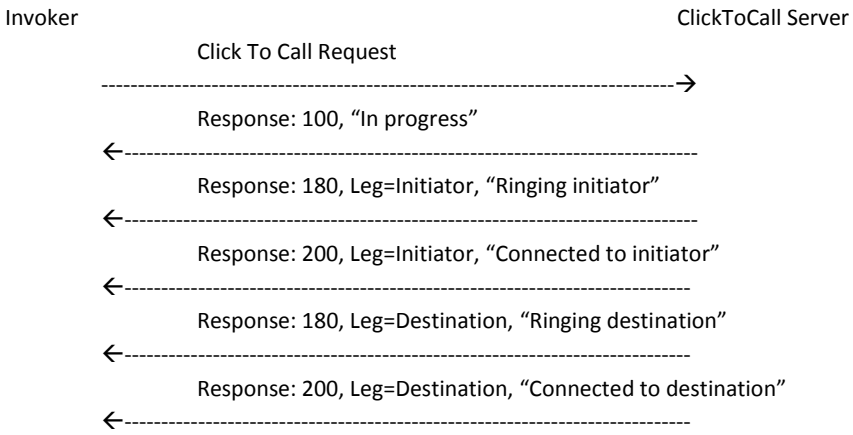
The initiator and destination provided will be passed through the configured translation rules to form a valid SIP URI.

If the anchor setting is not present then “false” is the default. With anchor=false the click to call server will first call the originator, then “REFER” the answering phone to the destination. In this case, after a successful REFER, the ClickToCall server is no longer involved in the call. If the REFER request fails then the ClickToCall server will revert to using an anchored call instead. With anchor=true the click to call server will first call the originator. Once the originator answers, the server will form a new call to the destination and will direct RTP to go directly between the two endpoints. In this case the ClickToCall server is behaving as a back to back user agent.

An XML response or set of responses are provided. The XML response will contain the original request in order to help the invoker correlated the response to the request. It will also contain a result code, result text and other potential data. Result codes follow the HTTP and SIP standards, whereby:

- codes in the range of 100-199 represent provisional responses, and indicate that a final response is pending
- codes in the range 200-299 represent a success response
- codes in the range of 300-699 represent a failure response

Since there are two legs to each click to call request, there is also a Leg attribute of the Result, this indicates which leg of the call the result code applies to (ie. Initiator or Destination). A typical successful click to call response chain would be:



The invoker can consider the request failed if it receives a result code greater than or equal to 300.

The invoker can consider the request successfully completed if it receives a response code in the range of 200-299 where Leg=Destination.

The initial 100 "In Progress" is unique in that it will also return the translated initiator and destination URIs for the invoker to use for diagnostics or debugging information. These are the URI's formed after passing the initiator and destination parameters through the configure translation rules. This response also does not contain a Leg attribute.

Example:

Request:

```
<ClickToCall>
  <Request>
    <Initiator>sip:scott@sipdomain.com</Initiator>
    <Destination>9055551234</Destination>
    <AnchorCall>false</AnchorCall>
  </Request>
</ClickToCall>
```

Responses:

```
<ClickToCall>
  <Request>
    <Initiator>sip:scott@sipdomain.com</Initiator>
    <Destination>9055551234</Destination>
    <AnchorCall>false</AnchorCall>
  </Request>
 <Response>
    <TranslatedInitiator>sip:scott@sipdomain.com</TranslatedInitiator>
```

```
    <TranslatedDestination>sip:9055551234@gateway.com</TranslatedDestination>
    <Result Code="100">In progress</Result>
  </Response>
</ClickToCall>

<ClickToCall>
  <Request>
    <Initiator>sip:scott@sipdomain.com</Initiator>
    <Destination>9055551234</Destination>
    <AnchorCall>false</AnchorCall>
  </Request>
  <Response>
    <Result Code="180" Leg=”Initiator”>Ringing initiator</Result>
  </Response>
</ClickToCall>

<ClickToCall>
  <Request>
    <Initiator>sip:scott@sipdomain.com</Initiator>
    <Destination>9055551234</Destination>
    <AnchorCall>false</AnchorCall>
  </Request>
  <Response>
    <Result Code="200" Leg=”Initiator”>Connected to initiator</Result>
  </Response>
</ClickToCall>

<ClickToCall>
  <Request>
    <Initiator>sip:scott@sipdomain.com</Initiator>
    <Destination>9055551234</Destination>
    <AnchorCall>false</AnchorCall>
  </Request>
  <Response>
    <Result Code="180" Leg=”Destination”>Ringing destination</Result>
  </Response>
</ClickToCall>

<ClickToCall>
  <Request>
    <Initiator>sip:scott@sipdomain.com</Initiator>
    <Destination>9055551234</Destination>
    <AnchorCall>false</AnchorCall>
  </Request>
  <Response>
    <Result Code="200" Leg=”Destination”>Connected to destination</Result>
  </Response>
</ClickToCall>
```