

**Oracle® Communications Session Border
Controller**

TSCF SDK Guide
Release 1.4.0

March 2016

Notices

Copyright© 2016, 2016, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

1 Overview.....	7
TSM Tunnel.....	8
SDK Host Operating System Relationship.....	8
Provided Functionality.....	10
2 Compiling the TSM Library and Documentation.....	13
SDK Directories.....	13
Generating the API Documentation.....	14
3 Accessing and Using the TSM SDK APIs.....	15
Sample TSM SDK-based Applications.....	15
Using The SDK To Create A TSM Tunnel.....	16
Enabling Redundancy.....	18
Error Codes.....	18

About this Guide

The Oracle® Communications Tunneled Session Controller SDK Guide describes the client-side SDK (software development kit) that facilitates the creation of secure tunnels between a client application and an Oracle Communications Tunneled Session Controller (TSC) server. A client is typically a softphone application that utilizes the SDK software libraries and source code to create TLS tunnels to a TSC service, thus achieving secure real time communications and ubiquitous firewall traversal.

This document specifically describes the SDK, functional libraries, and source code supplied with the SDK Version 1.4.0.

Related Documentation

The following table describes the documentation set for this release.

Document Name	Document Description
Oracle® Communications Tunneled Session Controller Essentials Guide	Contains information for setting up Tunneled Service Control Function Server that provides tunneled Session Initiation Protocol (SIP) and Message Session Relay Protocol (MSRP) secure transport services.

Revision History

Date	Description
March 2016	<ul style="list-style-type: none">• Initial release• Corrections to SDK directory structure

Overview

Tunnel Session Management (TSM) is a new feature on the Session Border Controller. Tunnel Session Management dramatically improves firewall traversal for real time communications for OTT VoIP applications and reduces the dependency on SIP/TLS and SRTP by encrypting access-side VoIP within standardized VPN tunnels. As calls or sessions traverse a TSM tunnel, the SBC will route all SIP and RTP traffic from within the TSM tunnel to the core (or appropriate destination).

Oracle Communications is working with other telecom providers and vendors to standardize TSM. Within the 3GPP, TSM is called a Tunneled Services Control Function (TSCF). Currently the 3GPP Technical Requirement draft is TR 33.8de V0.1.3 (2012-05) as a standardized approach for overcoming non-IMS aware firewall issues with supporting companies including China Mobile, Ericsson, Huawei, Intel, RIM, Vodafone, and ZTE. Beyond the standard, we provide exceptional tunnel performance & capacity within the Oracle Communications Session Border Controller as well as high availability, DDoS protection and our patented TSM Tunnel Redundancy to improve audio quality in lossy networks such as the Internet.

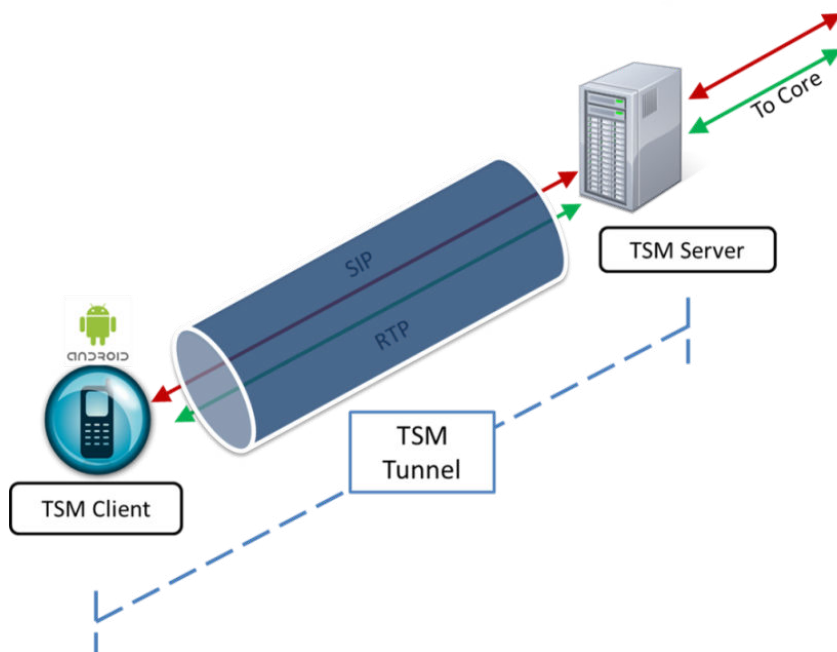


Figure 1: Basic TSM Setup

Overview

TSM consists of two parts:

- the TSM server (often referred to as a TSCF or Tunneled Services Control Function)
- the TSM client

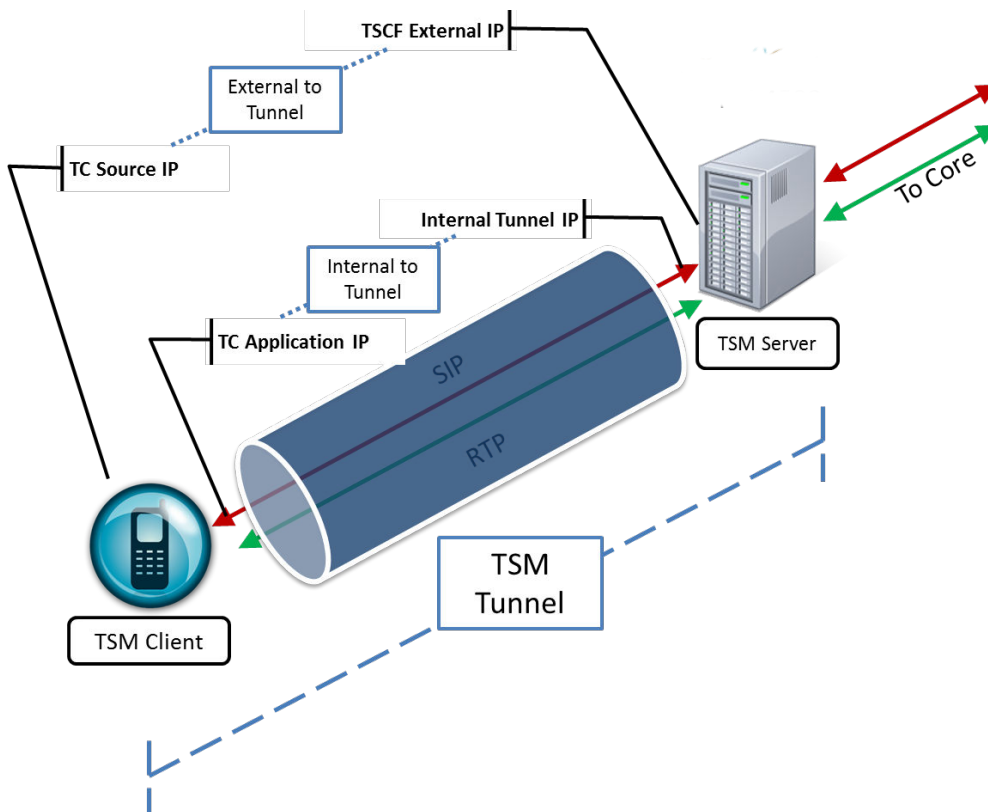
The TSM server resides and runs on the SBC and the TSM client runs within applications that reside on workstations, laptops, tablets and mobile devices (ex. Android, iPhone or iPad) and even network elements.

To deploy TSM enabled-clients such as softphones, SIP-enabled iOS/Android applications or contact center agent applications, customers and 3rd party ISVs will need to incorporate the open source TSM software libraries into their applications which will establish tunnels to the TSM server.

TSM Tunnel

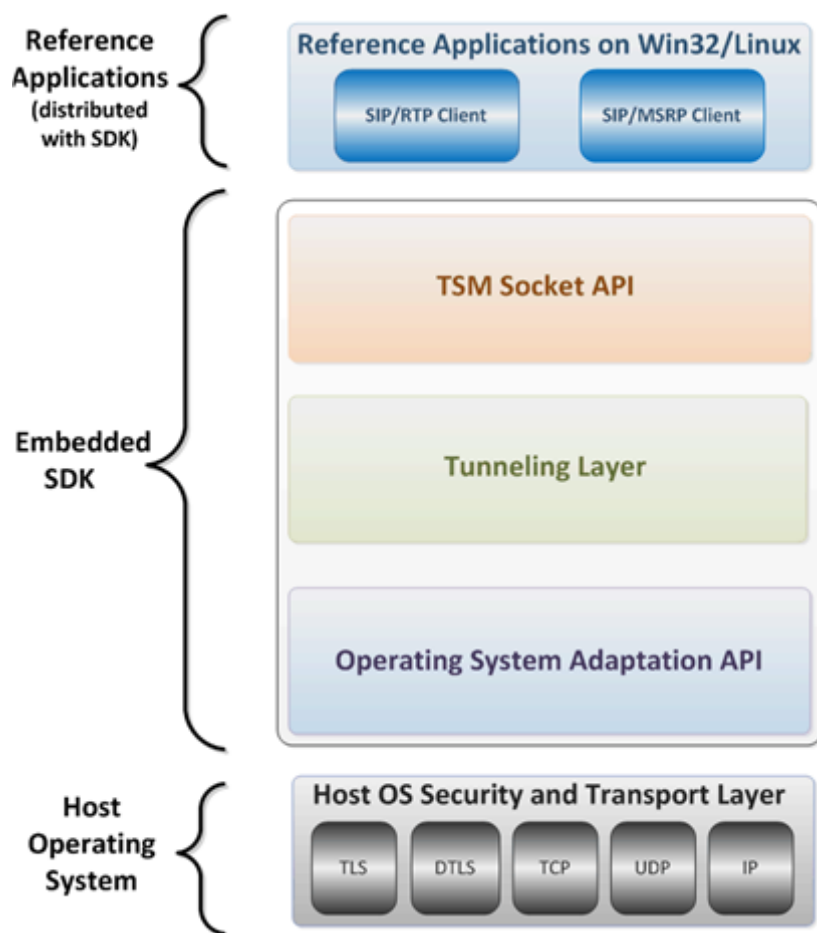
The following diagram briefly explains the various IP addresses utilized during the TSM session.

- TSCF External IP—This IP address is visible to any endpoint on the Internet and is used to initiate the TSM session between the TC and the TSCF. This may be configured under **security > tscf > tscf-interface**. See the TSCF Essentials Guide to configure the TSCF function on the server.
- TC Source IP—This IP address corresponds to the source address of the TC in its respective access network or it could be the IP of the Proxy behind which it is located.
- Internal Tunnel IP—This IP address will be assigned to the TC (once TLS authentication is successful) from a configured pool of IP addresses on the TSCF. It will be used to facilitate communication with the core (P-CSCF). The address pool can be configured under **security > tscf > tscf-address-pool**.
- TC Application IP—This is the IP address associated with the respective application (SIP / RTP / other) at the TC.



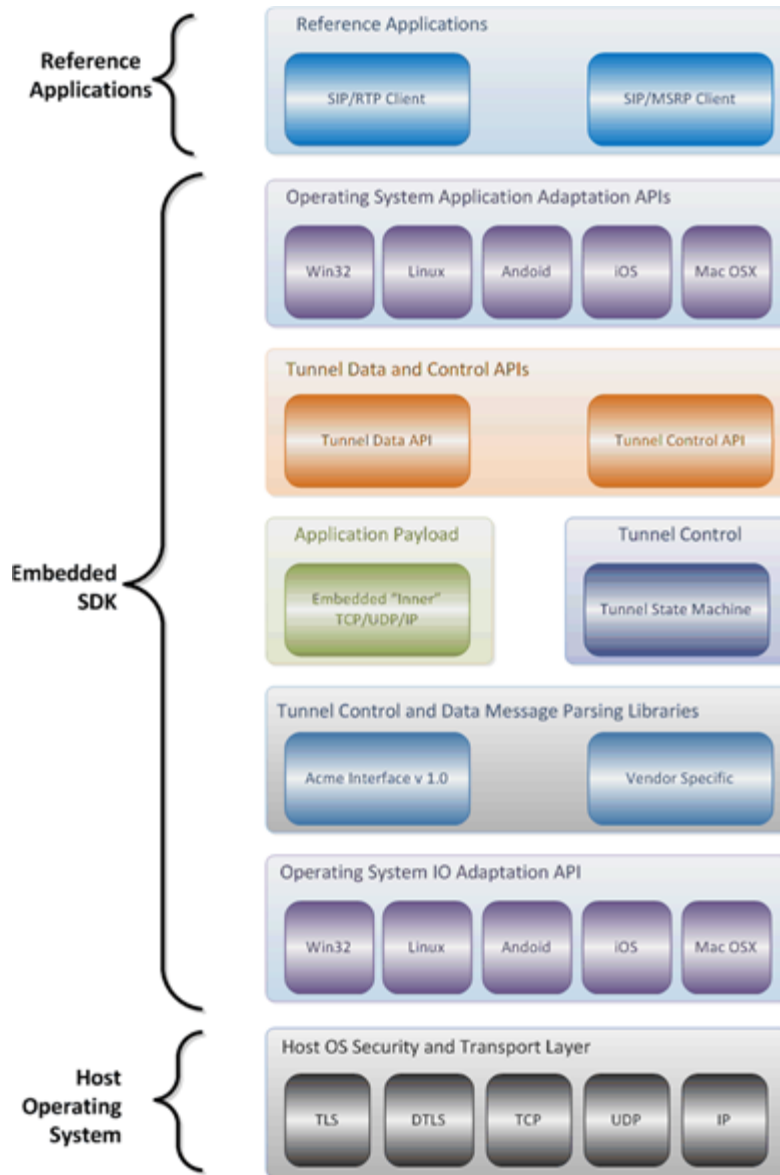
SDK Host Operating System Relationship

The following illustrations depict the relationship between the SDK and the host operating system



SDK/Host OS Relationship (Simplified View)

Overview



SDK/Host OS Relationship (Detailed View)

Provided Functionality

This SDK release supports the following platforms:

- Linux
- OS X
- iOS
 - 9.2
- Android
 - 2.3.x (Gingerbread)
 - 3.2.x (Honey Comb)
 - 4.0.X (Ice Cream Sandwich)
 - 4.4 (KitKat)
 - 5.X (Lollipop)

- 6.X (Marshmallow)
- Windows
 - Win32

This SDK release supports:

- Server Assigned Configuration mode
- Tunnel Transport
 - TLS
 - DTLS
- IP version
 - IPv4
 - IPv6







Note: When used in Decoupled Mode, the TSC also supports mixing IPv4 and IPv6. For example, you can use an IPv6 external address outside the tunnel and an IPv4 address inside the tunnel, or vice versa.

- Payload multiplexing within a tunnel
- Each SDK instance can support:
 - Up to 3 concurrent voice calls
 - Up to 10 MSRP chat sessions
 - 1 MSRP file transfer session
- Security Traversing Gateway (STG)

Compiling the TSM Library and Documentation

Read the documentation that corresponds to your application's target operating system.

Operating System	Description	Location
	This file provides information on how to compile the TSM SDK for the Android OS	/tsm/sdk/lib/android-ndk/README.tsc
	This file provides information on how to compile the TSM SDK for the iOS	/tsm/sdk/lib/README.ios
	This file provides information on how to compile the TSM SDK for the Linux OS	/tsm/sdk/lib/README
	This file provides information on how to compile the TSM SDK for the Windows OS	/tsm/sdk/lib/README.WIN



Warning: The OpenSSL and TSM libraries must be compiled before proceeding with development.

SDK Directories

SDK directories are shown below. Note that not all listed directories may be present (or supported) in the current release.

Compiling the TSM Library and Documentation

```
SDK:
+---apps
|   +---linphone
|
|   +---sipp
|
|   +---tsc_sip
+---docs
|   +---html
+---extlib
|   +---openssl-1.0.1g
+---lib
|   +---android-ndk
|   +---CSM
|   +---EIP
|   +---include
|   +---OSAA
|   +---TAPI
|   +---TPL
+---tools
|   +---wireshark
```

SDK based applications
Open source VoIP application utilizing TCP/TLS/DTLS/UDP for tunnel transport that has been ported to the SDK. Platform support limited to Windows, Android, and iOS. Refer to www.linphone.org for additional details.
Open source SIP traffic generator application utilizing TCP tunnel transport that has been ported to the SDK. Platform support limited to Linux. Refer to www.sourceforge.net for additional details.
Reference demonstration/development guide app (tsc_sip_client.c)
SDK Documentation
Authoritative API HTML-based documentation. Access via `"../html/index.html"` after running ``make doxygen``.
External, optional libraries
OpenSSL library
SDK Library source - to be linked with the target application
Android Specific library instructions and precompiled libs
tunneling Client State Machine
Embedded TCP/UDP/IP Stack
SDK API definitions
Operating System Application Adaptation APIs
Tunnel Data and Control APIs
Tunnel Control and Data Message Parsing Libraries
Development Tools
TSCF protocol dissector

Generating the API Documentation

You can generate the API documentation using the following steps:

1. Go to the SDK library folder.

```
cd tsm/sdk/lib
```

2. Auto generate the API documentation

```
make doxygen
```

3. Open the documentation using a browser pointing to `../docs/html/index.html`



Note: Functions and data structures have been removed from this document as of version 1.3 in favor of publishing the latest version and updates of these SDK elements directly from the code.

Accessing and Using the TSM SDK APIs

Sample TSM SDK-based Applications

A number of small application templates are found throughout the apps directory. Each of these files contains a small, well-defined set of functionality that enables a software developer to easily understand its implementation via the TSCF client-side SDK.

```

SDK:
+---apps/tsc_sip/tsc_sip_client
|   +---tsc_sip_client.c           Provides a basic SIP client utilizing
                                   TCP/TLS/UDP/DTLS for tunnel transport
                                   of inner UDP sockets. Supports Linux,
                                   Windows, and Android operating systems.

+---apps/tsc_sip/tsc_sip_server
|   +---tsc_sip_server.c         Provides a basic SIP server utilizing
                                   TCP/TLS/UDP/DTLS for tunnel transport
                                   of inner UDP sockets. Supports Linux,
                                   Windows, and Android operating systems.

+---apps/tsc_sip/tsc_sip_inner_tcp
|   +---tsc_sip_inner_tcp.c     Provides a basic SIP client utilizing
                                   TCP for tunnel transport and
                                   demonstrating usage of TCP sockets
                                   for applications such as HTTP --
                                   supports Linux operating systems.

```

All of the above files contain extensive comments making it an easy task to navigate through the code. Using `tsc_sip_client.c` as an example, you can readily proceed through the file.


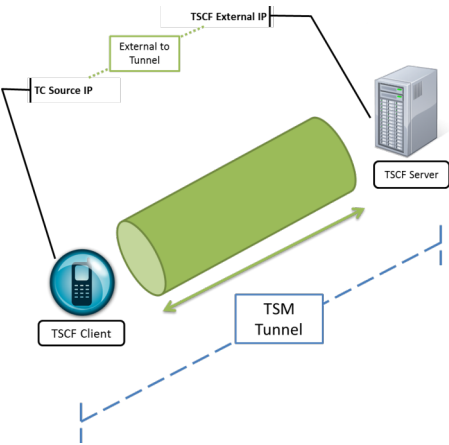
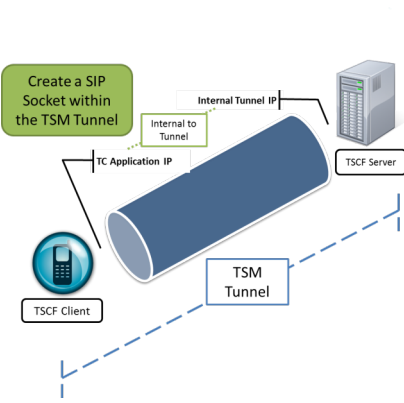
1. Search for `tsc_ctrl_init ()` and examine the code immediately following this function for the details of tunnel initialization.
2. Search for `Create a Tunnel` and examine the code immediately following for the details of tunnel creation and the configuration exchange between the TSCF server and client.
3. Search for `SIP SOCKET CREATION` and examine the code that creates and binds TSCF sockets.
4. Search for `REGISTER TRANSACTION` and examine the code that builds a SIP REGISTRAR request and processes the REGISTRAR response.
5. Search for `INVITE TRANSACTION` and examine the code that builds a SIP INVITE request and processes the REGISTRAR response.
6. Search for `Build and send ACK` to examine ACK creation and processing code.
7. Search for `RTP Exchange` to examine RTP code.

Accessing and Using the TSM SDK APIs

8. Search for BYE TRANSACTION to find code that terminates a SIP connection.
9. Search for TEST DONE to find code that terminates a tunnel.

Using The SDK To Create A TSM Tunnel

The following steps provide an outline on integrating a SIP client (in this example tsc_sip_client) with the TSM SDK. Please refer to the file tsc_sip_client.c (located at tsm/sdk/apps/tsc_sip/tsc_sip_client) which contains working code references on establishing a TSM tunnel and making a SIP/RTP based call.

Initialize the TSCF-Client side API	
<p>Use the TSCF Server IP address, port, transport type, wireshark tracing and certificate parameters (if using TLS/DTLS) to initialize the client. Populate the required information in a tsc_tunnel_params type structure.</p> <p>Search for "tsc_ctrl_init()" in the reference file for actual code implementation.</p>	
Create a TSM Tunnel	
<p>Once initialized, create a TSM tunnel between the TSCF client and server, register for callbacks and obtain the SIP server IP address.</p> <p>Search for the text "Create a tunnel" in the reference file for actual code implementation.</p>	
Create a SIP Socket within the TSM Tunnel	
<p>With the TSM tunnel established, create a SIP socket and bind the local address assigned by the TSCF server to it.</p> <p>Search for the text "SIP SOCKET CREATION" in the reference file for actual code implementation.</p>	
Send SIP messages over the TSM Tunnel	

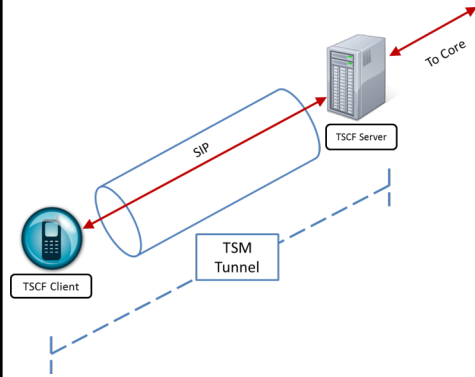
You can now send SIP messages over the TSM tunnel. Based on the kind of SIP applications, you may need to send a REGISTER message or directly initiate a peer-to-peer call through an INVITE message.

You can find examples of both messages being sent by looking for the following text. Since this is only sample code, please use the same as reference for sending and receiving SIP messages via the TSM tunnel.

REGISTER TRANSACTION: Constructs and sends a REGISTER message to the tsc_sip_server. The code currently doesn't check for a 200 OK message.

INVITE TRANSACTION: Constructs and sends an INVITE message and handles a 200 OK.

Build and send ACK (to INVITE transaction) : This code builds and sends an ACK to the INVITE received.

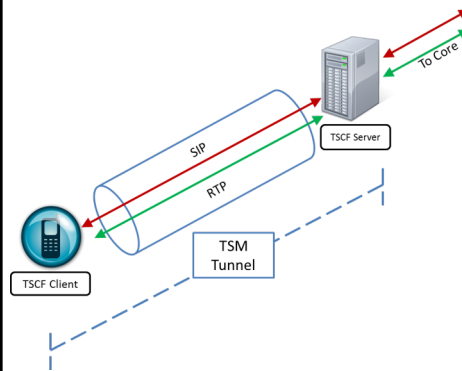


Send RTP message over the TSM Tunnel

As part of the call, there is also sample code to send and receive RTP packets over the TSM tunnel. Search for the text "RTP EXCHANGE" within the reference file for a code implementation.

The code builds RTP packets purely for the purpose of simulation.

At this point you should be able to make calls through the TSM tunnel.



Terminate a SIP call over the TSM Tunnel

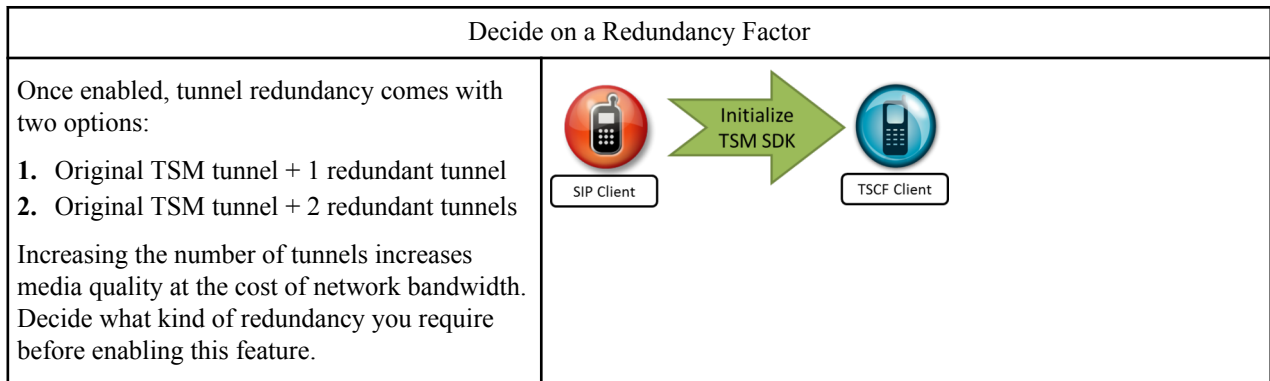
After you are done with the call, you may terminate it using sample code in the reference file. Search for the text "BYE TRANSACTION".

Terminate a TSM Tunnel

If your application has determined that it doesn't require the use of a TSM tunnel, you may terminate the tunnel using the sample code. Search for "Finished, Socket, Tunnel Cleanup".

Enabling Redundancy

TSM enables an application to improve media quality under adverse network packet loss through the tunnel redundancy feature.



Tunnel redundancy can be enabled on a per-socket basis.

1. Create a notification handler function. Once requested, the TSCF notifies the application whether the redundancy was enabled successfully.
2. Set a socket option with the type of redundancy factor (1 or 2).

Search for the text “RTP socket created” and "TSC_REDUNDANCY" in the reference file for actual code implementation that creates redundant tunnels for RTP packets.

Error Codes

The following are error codes you can check against when calling the TSC layer to ensure you can handle all success and failure scenarios:

Return Value	TSC_ERROR_CODE
0	tsc_error_code_ok
1	tsc_error_code_error
2	tsc_error_code_not_logged
3	tsc_error_code_cannot_connect
4	tsc_error_code_cannot_configure
5	tsc_error_code_keepalive_failure
6	tsc_error_code_service_failure
7	tsc_error_code_cannot_rcv_data
8	tsc_error_code_no_data
9	tsc_error_code_cannot_send_data
10	tsc_error_code_authenticate
11	tsc_error_code_cannot_release
12	tsc_error_code_queue_overflow