

Black Book

ixia

Edition 10

Voice over IP

Your feedback is welcome

Our goal in the preparation of this Black Book was to create high-value, high-quality content. Your feedback is an important ingredient that will help guide our future books.

If you have any comments regarding how we can improve the quality of this book, or suggestions for topics to include in future Black Books, please contact us at ProductMgmtBooklets@ixiacom.com.

Your feedback is greatly appreciated!

Copyright © 2014 Ixia. All rights reserved.

This publication may not be copied, in whole or in part, without Ixia's consent.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

Ixia, the Ixia logo, and all Ixia brand names and product names in this document are either trademarks or registered trademarks of Ixia in the United States and/or other countries. All other trademarks belong to their respective owners. The information herein is furnished for informational use only, is subject to change by Ixia without notice, and should not be construed as a commitment by Ixia. Ixia assumes no responsibility or liability for any errors or inaccuracies contained in this publication.

Contents

How to Read this Book.....	vii
Dear Reader	viii
VoIP – Voice over IP	1
Test Case: Determining the Maximum Call Setup Rate (CPS)	7
Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems	11
Test Case: Determining the Maximum Number of Concurrent Calls.....	47
Test Case: VoIP Quality of Service in Converged Networks.....	75
Test Case: Subjective Quality of Voice	87
Test Case: Determining the Maximum Transaction Rate for VoIP Protocols	105
Test Case: Using VoIP to Measure NAT/PAT Performance	119
Test Case: Determining the Capacity of a VoIP to PSTN Gateway	155
Test Case: Determining the Performance of a Session Border Controller	173
Test Case: Measuring Quality of Experience for Voice Calls in LTE.....	189
Test Case: Measuring Quality of Experience for Multimedia VoIP Calls	213
Test Case: Telephony Denial of Service	235
Contact Ixia	257

How to Read this Book

The book is structured as several standalone sections that discuss test methodologies by type. Every section starts by introducing the reader to relevant information from a technology and testing perspective.

Each test case has the following organization structure:

Overview	Provides background information specific to the test case.
Objective	Describes the goal of the test.
Setup	An illustration of the test configuration highlighting the test ports, simulated elements and other details.
Step-by-Step Instructions	Detailed configuration procedures using Ixia test equipment and applications.
Test Variables	A summary of the key test parameters that affect the test's performance and scale. These can be modified to construct other tests.
Results Analysis	Provides the background useful for test result analysis, explaining the metrics and providing examples of expected results.
Troubleshooting and Diagnostics	Provides guidance on how to troubleshoot common issues.
Conclusions	Summarizes the result of the test.

Typographic Conventions

In this document, the following conventions are used to indicate items that are selected or typed by you:

- **Bold** items are those that you select or click on. It is also used to indicate text found on the current GUI screen.
- *Italicized* items are those that you type.

Dear Reader

Ixia's Black Books include a number of IP and wireless test methodologies that will help you become familiar with new technologies and the key testing issues associated with them.

The Black Books can be considered primers on technology and testing. They include test methodologies that can be used to verify device and system functionality and performance. The methodologies are universally applicable to any test equipment. Step-by-step instructions using Ixia's test platform and applications are used to demonstrate the test methodology.

This tenth edition of the black books includes twenty two volumes covering key technologies and test methodologies:

Volume 1 – Higher Speed Ethernet

Volume 12 – IPv6 Transition Technologies

Volume 2 – QoS Validation

Volume 13 – Video over IP

Volume 3 – Advanced MPLS

Volume 14 – Network Security

Volume 4 – LTE Evolved Packet Core

Volume 15 – MPLS-TP

Volume 5 – Application Delivery

Volume 16 – Ultra Low Latency (ULL) Testing

Volume 6 – Voice over IP

Volume 17 – Impairments

Volume 7 – Converged Data Center

Volume 18 – LTE Access

Volume 8 – Test Automation

Volume 19 – 802.11ac Wi-Fi Benchmarking

Volume 9 – Converged Network Adapters

Volume 20 – SDN/OpenFlow

Volume 10 – Carrier Ethernet

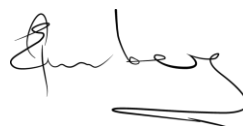
Volume 21 – Network Convergence Testing

Volume 11 – Ethernet Synchronization

Volume 22 – Testing Contact Centers

A soft copy of each of the chapters of the books and the associated test configurations are available on Ixia's Black Book website at <http://www.ixiacom.com/blackbook>. Registration is required to access this section of the Web site.

At Ixia, we know that the networking industry is constantly moving; we aim to be your technology partner through these ebbs and flows. We hope this Black Book series provides valuable insight into the evolution of our industry as it applies to test and measurement. Keep testing hard.



Errol Ginsberg, Acting CEO

Voice over IP

Test Methodologies

This booklet provides baseline test execution plans for running common VoIP testing scenarios. It provides a common set of recommendations and guidelines for running the test cases against various VoIP implementations.

VoIP – Voice over IP

VoIP refers to the transmission of voice data over IP-enabled networks, rather than using the traditional circuits of the PSTN (Public Switched Telephone Networks). Today, VoIP is associated with IP telephony, which refers to the family of protocols used to deliver the voice data – RTP/SRTP, H.323, SIP, MGCP, MEGACO/H248, SCCP (Skinny), and SIGTRAN.

VoIP Benefits

The most attractive benefit of VoIP is the cost savings, which can be substantial in large enterprises and for service providers. The savings can come from a number of different sources. For example, calls made over private data networks bypass PSTN network toll circuits, avoiding regulatory and per-call charges.

Many benefits are available using a common network for voice, video, and data applications. The unification of voice, text, video, and data on the same network allows quicker integration of multiple services, improving the way users collaborate and leading to enhanced productivity. Unified communications can transform a desktop application such as Microsoft® Outlook™ into a communication center in which e-mail, voice and video calls, screen sharing, instant messaging, voice mails, presence, and availability are easily accessible, resulting in enhanced productivity. As an example, a meeting (screen sharing, audio) can be recorded with a simple click of the mouse and then shared with team members who could not attend the meeting. Voice mail notifications can pop-up directly on the screen or be received via email.

User mobility and portability are advantages provided by IP networks. Using software applications such as soft phones, anyone can transform a laptop or PC into a mobile IP phone. Hence, employees can work from anywhere and home subscribers can gain additional mobility when they travel. Regardless of whether they are traveling or working from home, the mobility aspect adds a simple, yet powerful way to maintain communication.

User mobility allows IT administrators to provide the same communication services without installing a local IP-PBX in each remote location.

Another benefit comes from simplified scalability – with IP telephony, businesses can control and avoid underutilized equipment. New users may be added to existing locations one at a time, rather than buying or leasing equipment that may remain underutilized.

VoIP Challenges

To offer a good alternative to PSTN services, VoIP must provide the same speech quality and service reliability as customers are accustomed to receiving from legacy circuit-switched networks. PSTN is well-known for its five nines in reliability (99.999 percent uptime, which corresponds with 5 minutes of downtime a year). The quality of voice in PSTN networks is

referred to as toll quality – this corresponds to a mean opinion score (MOS) score of 4.00 measured on a scale from 1 (fair) to 5 (excellent), as specified by the ITU P.800 standard.

Network Requirements for Toll-Quality

The network itself plays a critical role in delivering toll-quality voice. IP telephony requires the network to be designed to deliver enough bandwidth and to meet specific latency, jitter, and packet loss requirements. While data traffic is less impacted in networks with high delay, jitter, and packet loss, voice communication is significantly impacted because it requires an end-to-end communication and has real-time constraints.

Bandwidth Requirements and CODECs

Bandwidth is directly affected by the phone's encoder and the expected number of simultaneously active calls. A higher number of active conversations (calls) corresponds to higher bandwidth requirements.

Audio CODECs, which implement speech compression, can differ significantly in bandwidth requirements, speech quality, encoding delay, loss resiliency, and computational requirements. Hence, the selection of a CODEC can directly affect the number of active calls that a link can support and the voice quality. In general, reducing the nominal bandwidth (that is, payload throughput) degrades speech quality. Because the relationship between bandwidth and speech quality is not a linear one, some CODECs can significantly reduce the nominal bandwidth required while preserving a good quality. For example, G.729 can reduce the nominal bandwidth 8 times compared to G.711, while maintaining good speech quality.

Many CODECs support silence suppression mechanisms, which can reduce the required bandwidth up to 50 percent.

Effect of Delay on Conversation

It is well known that network delay leads to two-way conversation difficulties. This effect can be better described using an example. Let us assume that Alice and Bob have a phone conversation using an IP network with a high round-trip delay (for example, 500 ms). Say that Alice decides to interrupt Bob while he is talking. Due to the delay, Alice will continue to hear old information from Bob, while Bob will continue to speak until the interruption is heard. For Alice, the effect of the delay is perceived as ignorance from Bob's side. Hence, she may stop talking. On the other side, Bob finally hears Alice's interruption and stops as well. Both Bob and Alice remain silent, and then both may start talking or stop talking. To meet the toll-quality requirement, the one-way delay must be kept below 100 ms. A one-way delay value of 150 ms makes its presence noticeable, but they can still have an acceptable conversation.

In networks where VoIP interacts with PSTN, the VoIP user may hear an echo if the PSTN network is not correctly tuned. The round-trip delay directly affects the effect of echo. Hence, the higher the roundtrip value, the more annoying the effect of the echo.

ITU G.114 provides recommendations for network latency.

Jitter

Jitter represents the variation of delay as measured by the receiver. Users perceive jitter as speech degradation. Many devices implement jitter buffers that can remove the jitter effect. A jitter buffer temporarily stores arriving packets to minimize delay variations and discards the packets that arrive too late. If a jitter buffer is too small, then an excessive number of packets may be discarded, which can lead to call quality degradation. If the jitter buffer is too large, then an additional delay is added to the conversation, which can lead to difficulties in conversation (see prior section on the effects of delay).

The most common causes for jitter are network congestion and router/switch queuing methods.

Packet Loss

Packet loss is probably the most common factor affecting speech quality. All voice CODECs can accept some packet loss without dramatically degrading speech quality. Many CODECs supports as much as 5 percent of random packet loss while maintaining acceptable voice quality. However, loss of many consecutive packets dramatically affects voice quality, even if the total percentage loss is low. When designing networks and applications the target should be zero packet loss.

In converged networks where voice and data use the same resources, voice traffic must be configured to have priority treatment over data traffic, which is not as affected by delay, jitter, and packet loss.

High Availability, Reliability, Performance, and Capacity Tests

IP telephony networks satisfying the high **availability** requirement need to address consumers' needs to register, place calls, and receive calls even at peak call rates. In a similar way, IP telephony systems must handle traffic even when devices are in maintenance or have failed. High availability networks are assured by adding redundant systems for both signaling and media components. Hence, when a primary node in a voice network is down for maintenance or because of a failure, a redundant device (secondary or tertiary) can take over the processing of the voice traffic.

Redundancy measures the ability to continue operation in the event of a failure.

Performance measures the maximum rate that a system or device under test (DUT) can sustain. For IP telephony systems, the key performance measurements are the maximum call setup rate and the maximum registration rate. The performance of systems without session intelligence (for example, a stateless proxy server) can be additionally measured using metrics such as maximum transaction rates or maximum messages per second.

In addition to performance measurements, **capacity** tests help in determining the maximum number of active calls that can simultaneously be active on a DUT. While the presence of media has an immediate impact on the performance and capacity of a DUT that forwards media packets between two hops, another system may not be impacted. For example, because only

session internet protocol (SIP) messages can traverse the DUT. While it is recommended that media traffic be included with any call testing, media may or may not be generate during the Call Hold Time as part of the test. The decision needs to be taken based on the logical components included in the system under test (SUT). As an example, a session border controller (SBC) must be tested by generating media after calls are established, while testing of a SIP proxy server will not necessarily require media.

While media traffic may not affect a specific logical component of the DUT, testing with media enabled can help discover additional issues, which may not be detected without media. An example is testing a device that implements a proxy element for SIP-only signal forwarding.

Other capacity tests measure the maximum number of registered users that can be concurrently maintained. The overhead added by registration or other keep-alive messages exchanged between servers and clients is frequently overlooked when testing. For example, a SIP registration that requires authorization includes four messages versus nine messages for a call session, which also requires authentication. On systems that need to handle thousands of users, the overhead is quite significant due the fact that online users will re-register every hour or less. On some systems, the registration can recur as often as every 60 seconds. This overhead is important for systems that implement registrar and proxy components on the same server.

In general, the DUT needs to be monitored during testing for the following:

- Memory leaks
- Stability
- Ability to sustain performance for an extended period
- Signaling quality of service
- Media quality of service

Testing Challenges in VoIP Networks

Convergence

Due to voice traffic's high sensitivity to delays, packet loss, and jitter, a converged network can have a negative impact on the voice quality when is not properly configured.

Quality of service (QoS) configuration helps to guarantee required bandwidth while minimizing latency, jitter, and packet loss. Minimizing those effects is assured by detecting and prioritizing voice traffic, thus controlling available bandwidth.

Data traffic may be impacted when voice traffic is prioritized, because it is restricted from accessing bandwidth when VoIP needs it. However, data applications are less sensitive to packet loss and delay, and user experience is less affected.

Proper testing of device handling of both data and voice traffic requires generating a mix of data and voice traffic that is common in the environment(s) where the DUT will be deployed. Baseline performance tests must be conducted for each individual protocol passing through the network.

Complexity

In a distributed architecture in which devices from multiple vendors are used, VoIP networks add an additional layer of complexity. Interoperability between different protocols (for example, SIP to H.323, H248 to SIP) and the interaction between different media (for example, PSTN, wireless, and pure VoIP networks) can lead to additional issues related to protocol translation (for example, SIP to H.323) and multiple media encoding-decoding operations that lead to degradation of speech quality.

Complex Features

When VoIP shares a common network with data applications, unified services and extended feature sets can be provided beyond those available with regular PSTN networks. The performance impact of common, complex features, such as Class 5 telephony features (for example, call transfer, call conference, call parking, call waiting), is underestimated. In practice, the complexity and message overhead added by those features may lead to system stability issues if minimal testing was performed at low scale using just a few phones.

Security

A VoIP system is susceptible to a number of attacks and threats, including denial of service (DoS) attacks, eavesdropping, registration and session hijacking, and server impersonation. Multiple levels of security are implemented at the transport and application levels to prevent attacks. Protocols such as IP security (IPsec), TLS, and SRTP add additional complexity, and system performance can decrease.

Test Case: Determining the Maximum Call Setup Rate (CPS)

It is important to determine the maximum call rate for devices that handle VoIP call setup and teardown phases. A few examples of such devices are:

- Soft switches
- IP PBXs
- Call managers
- Session border controllers
- VoIP application layer gateways
- Voice mail and interactive voice response servers (IVRs)
- Application servers
- Conference bridges

The primary performance metric assesses whether the attempted call rate is sustainable for a long time without call failures or affecting users' experience.

The following standard metrics can help assessing the QoS for signaling protocols:

- **Call setup time** – measures the time required to setup a call, including the call acknowledgement. This metric is significant if the called party answers immediately. In practice, someone may answer after a variable time.
- **Post dial delay** – measures the time required to receive the first ring back notification after the last digit of the destination phone number was dialed.

The post dial delay is significant for the caller. A large post dial delay is perceived as a call that did not go through because the caller does not receive any indication of the call progress in the ring back tone. A post dial delay value of less than four seconds is recommended for IP telephony systems. When post dial delay exceeds four seconds, the caller may hang up and attempt a new call.

- **Post pickup delay** – measures the time after the call was answered until the first media packet is received.

The post pickup delay has a direct impact on the user experience. Normally, a caller starts to be affected with delays higher than four seconds.

- **Media delay** – measures the time to receive the first media packet after a call setup message was sent.

While the call may be connected via a delayed conversational path, media delay can still be annoying to the caller.

Test Case: Determining the Maximum Call Setup Rate (CPS)

- **End call time** – measures the time to tear down a call.
- **Message retransmissions** – counts the number of retransmissions required to deliver control plane messages across the network

A high number of retransmissions received or transmitted by a DUT can drop the overall performance of the DUT and cause stability issues. The number of messages retransmitted must be maintained below 5 percent in a deployed IP telephony system, and should be as close as possible to zero when testing a DUT in isolation.

- **Call establishment ratio** – measures the ability of a called party to successfully connect a call and establish a conversation.

$$\text{Call Establishment Ratio} = \frac{\text{\# of Calls Connected}}{\text{\# of Calls Attempted}} * 100 \text{ [\%]}$$

- **Call completion ratio** – measures the ability of a called party to successfully connect the call and to successfully complete the call by initiating or receiving the appropriate disconnect request.

$$\text{Call Completion Ratio} = \frac{\text{\# of End Calls completed}}{\text{\# of Calls Attempted}} * 100 \text{ [\%]}$$

When the DUT simultaneously handles signaling and media it is important that media traffic is enabled for the calls. Media traffic must be tested to ensure that it meets the QoS requirements. The following standard metrics can be used to help assess the QoS for RTP-based voice traffic:

- Assessing media transport:
 - Packet loss
 - Jitter
 - One-way delay
 - RTP packets sent versus RTP packets received
 - Inbound and outbound throughput
 - Maximum consecutive loss
 - Packet loss distribution
 - Packet loss correlation
 - Packet errors
 - Packet duplicates
 - Calls without media versus calls with media
- Audio quality metrics
 - R-Factor

Test Case: Determining the Maximum Call Setup Rate (CPS)

- MOS
- PESQ (obsoletes PAMS and PSQM)
- Video quality metrics
 - V-factor
 - Absolute MOS-V
 - Relative MOS-V
 - Video loss degradation
 - Video jitter degradation
 - Video CODEC degradation

The following test methodology provides a structural approach to determining the maximum call rate supported by a particular DUT.

- Set performance goals for signaling and media traffic.
 - Targeted calls per second.
 - Use a binary search to determine the maximum call rate. Start by using a shorter test trials of 15-30 minutes, which can minimize the time required to discover the maximum supported peak rate.

During the test verify that:

- The call rate is sustained without call failures.
- The SIP QoS is in an acceptable threshold range.
- The RTP QoS is in an acceptable threshold range.
- Using the measured maximum call setup rate, the test should be repeated for a minimum of 72 hours to confirm that the DUT is free of memory leaks and maintains acceptable QoS.
- After the peak rate is determined, short tests can be executed to gather additional data points that can characterize the DUT's QoS. The following charts may be used to describe DUT performance:
 - CPS versus signaling QoS charts
 - CPS rate versus post dial delay, post pickup delay, call setup time, and end call time
 - CPS rate versus call establishment ratio and call completion ratio
 - CPS rate versus retransmissions
 - CPS versus media QoS charts (if applicable)
 - CPS versus MOS
 - CPS versus PESQ-LQ and PESQ-LE
 - CPS versus packet loss percentage and packet loss correlation

Test Case: Determining the Maximum Call Setup Rate (CPS)

- CPS rate versus jitter and one-way delay
- To plot the performance characteristic of the DUT, at least 10 data points should be included on the chart (5 percent, 10 percent, 25 percent, 40 percent, 50 percent, 65 percent, 75 percent, 85 percent, 95 percent, 100 percent are suggested) based on the maximum rate supported. Using data points that are higher than 100 percent of the determined peak rate can be also useful, because they can depict the behavior of the DUT when overloaded.

Some VoIP devices only handle call signaling aspect while others handle both signaling and media simultaneously.

Further DUT performance characterization should be pursued by changing the test variables that influence the performance of the DUT. Those test variables include:

- IP version (IPv4/IPv6)
- Transport layer (UDP/TCP/TLS)
- IP and port mapping for signaling and media (1:1, 1:n)
- Call duration
- Number of CODECs per user

If media traverses the DUT, the negotiated CODEC type must be considered. The CODEC that is used is even more important when the DUT acts as a transcoder (for example, converts the voice from G.711 to G.729).

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

Overview

The goal of this test methodology is to determine the maximum peak load, measured in calls per second, supported by a DUT that implements at least one of the following SIP logical components defined by the SIP protocol:

- SIP proxy server
- SIP redirect server
- SIP back-to-back user agent

The primary test metric is the number of calls that the DUT sustains for a long time while providing the desired user experience. In addition, the QoS metrics discussed in the **Max Call Setup Rate** section are required to determine pass/fail criteria. Those pass/fail criteria may be different for a DUT that is isolated by the test tool, as opposed to a complete IP telephony system that is passing calls over a WAN. For example, the post dial delay should not exceed 250 ms if the test tool isolates the DUT, while a post dial delay of 2 seconds is acceptable when the DUT is a complete IP telephony system where the post dial delay is measured across the WAN. Obviously, the QoS constraints are more restrictive when running with the DUT in isolation because practical deployments include a chain of DUTs that help the call reach the desired destination. Also, for calls transported over a WAN, additional delay, jitter, packet loss, and retransmissions can be induced by the network. The sum of these delays can exceed the end-to-end accepted thresholds (for example, 2 seconds for SIP post dial delay, 150 ms for one-way delay).

The standard metrics below can help assess the SIP QoS.

- **Call setup time** – for the calling part, the call setup time measures the time from when an INVITE message is sent until an ACK message is sent.

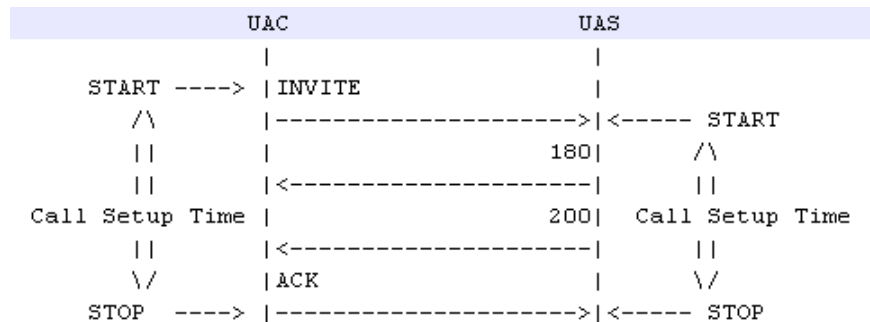


Figure 1. SIP call setup time

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

For call destinations, call setup time measures the time from when the INVITE message is received until the ACK message is received. When the call is authenticated, the call setup time includes the authentication time as well.

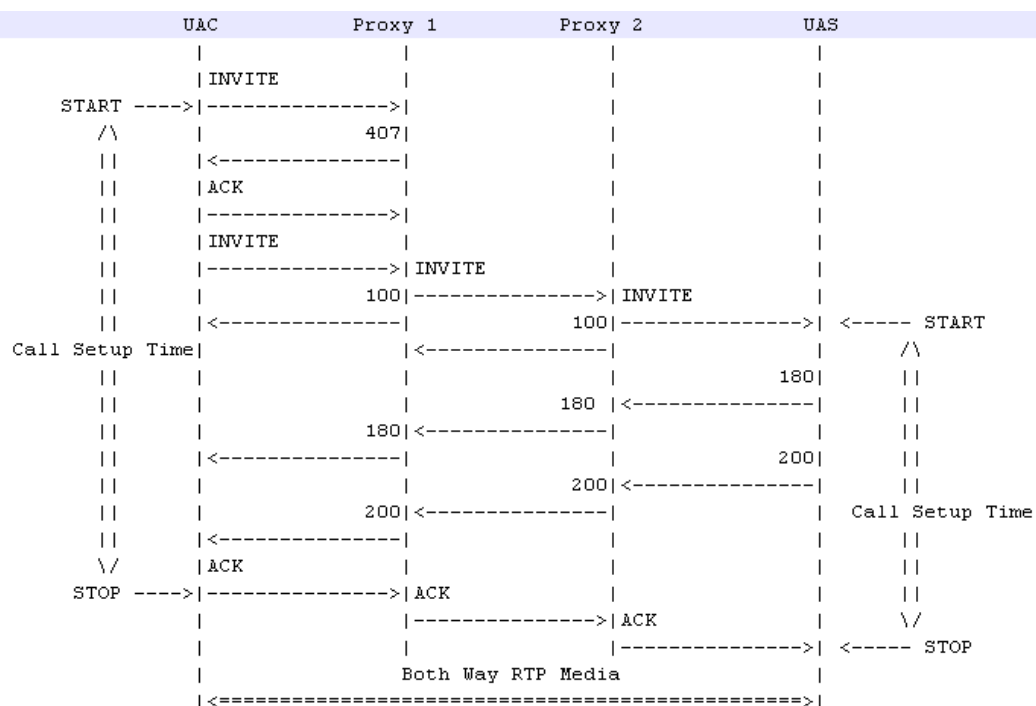


Figure 2. Call setup time for a call with authentication required

- **Post dial delay** – measures the time required to get the first ring back notification after the last digit of the destination phone number was dialed.

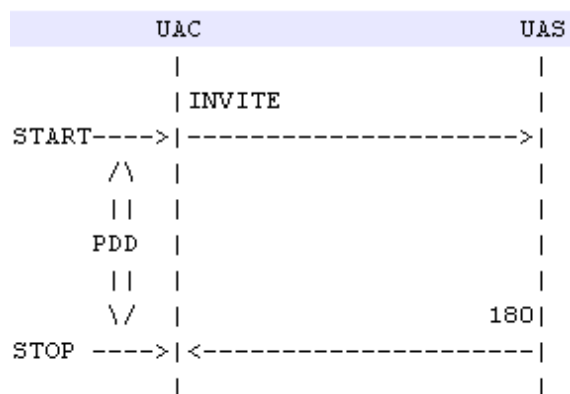


Figure 3. Post dial delay (PDD) in SIP

For SIP calls, post dial delay measures the time from when the INVITE is sent until the 180 RINGING message is received.

The post dial delay has an important significance for the caller. A large post dial delay is perceived as a call that does not go through, because the caller does not receive any

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

indication of call progress (ring back tone). A post dial delay value below four seconds is recommended for IP telephony systems. When the post dial delay values exceed four seconds, the caller may hang up and attempt a new call.

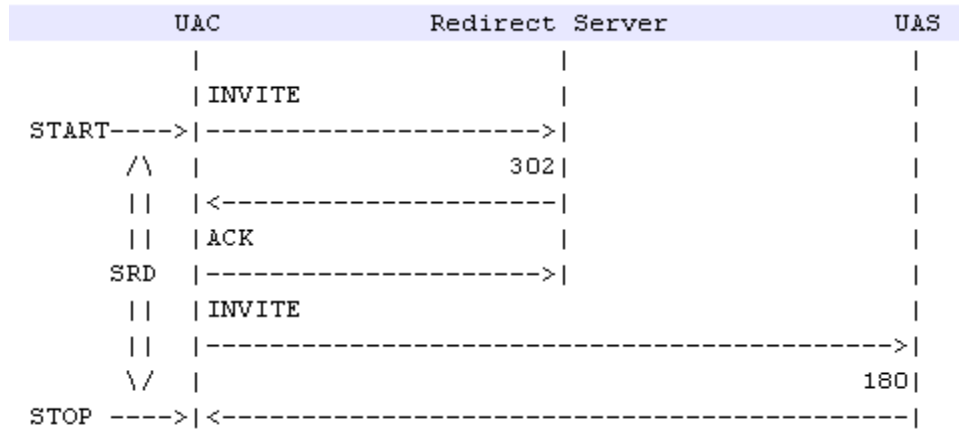


Figure 4. Post dial delay measured on a redirected call

- **Post pickup delay** – measures the time from when the call was answered until the first media packet is received. In SIP, this corresponds to the time elapsed from when the 200 OK response for the INVITE was sent until the first RTP packet is received.

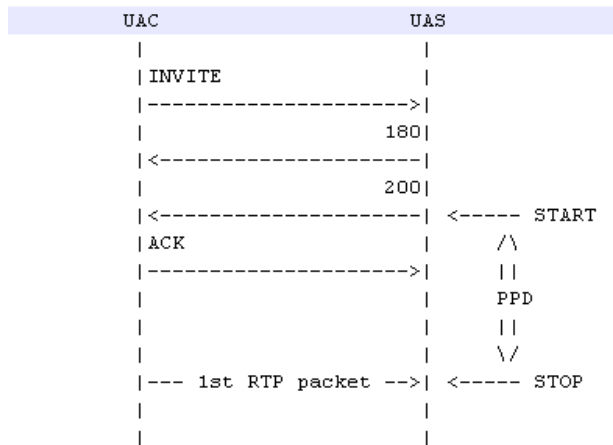


Figure 5. Post pickup delay

The post pickup delay has a direct impact on the user experience. Normally, a caller starts to be affected with delays higher than four seconds.

- **Media delay** – measures the time to receive the first media packet after the call setup message was sent. This measurement can be applied for both the calling party and the called party.

When using SIP, the media delay for the calling party is calculated as the time between the INVITE message and the first RTP packet is received. For the called party, the

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

media delay measures the time after the INVITE message is received and until the first RTP packet is sent.

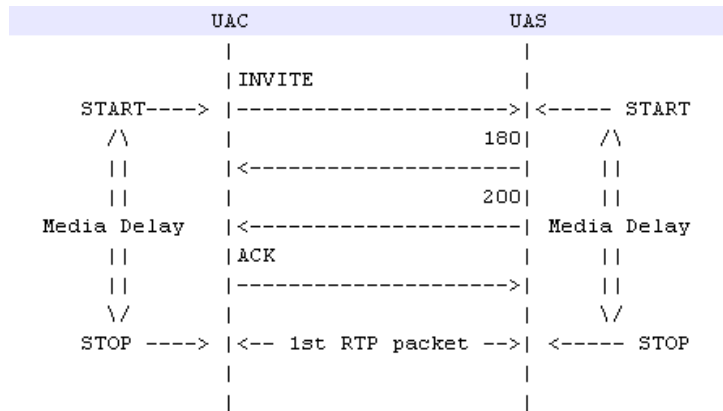


Figure 6. Media delay

- **End call time** – measures the time to tear down the call. When using SIP, end call time is measured as the time elapsed between the BYE message and until a 200 OK successful response is received.

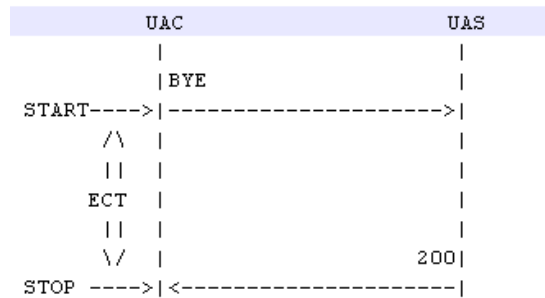


Figure 7. End call time

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

- Message retransmissions – counts the number of retransmissions required to deliver control plane messages across the network.

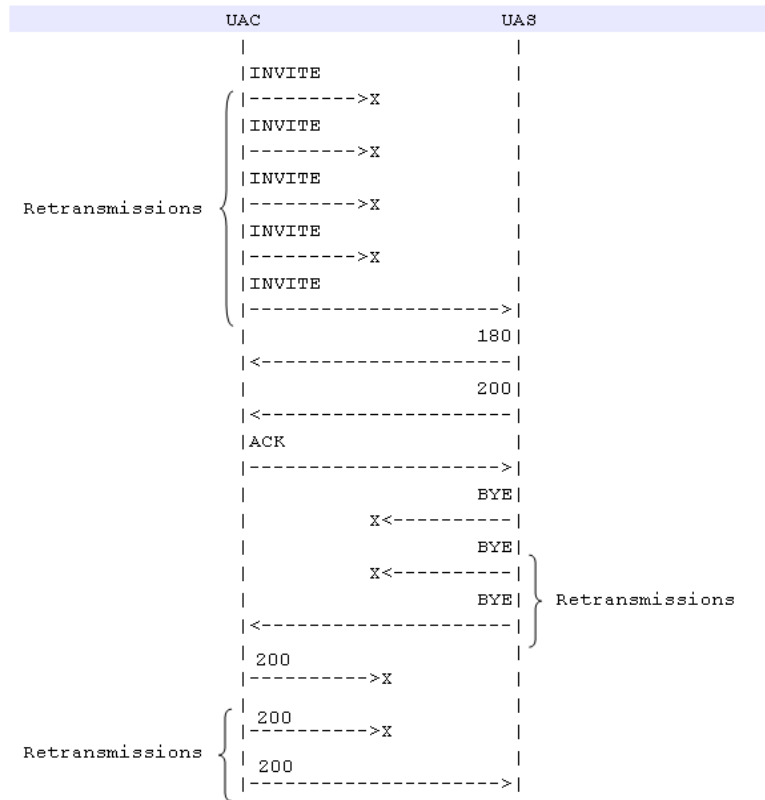


Figure 8. SIP message retransmission

A large number of retransmissions received or transmitted by a DUT can decrease the overall DUT performance and cause instability. The number of messages retransmitted must be maintained below 5 percent in a deployed IP telephony system and should be kept close to zero when testing a DUT in isolation.

- **Call establishment ratio** – measures the ability of a called party to successfully connect a call and establish a conversation.

$$\text{Call Establishment Ratio} = \frac{\# \text{ of Calls Connected}}{\# \text{ of Calls Attempted}} * 100 [\%]$$

- **Call completion ratio** – measures the ability of a called party to successfully connect the call and complete the call by initiating or receiving the appropriate disconnect request.

$$\text{Call Completion Ratio} = \frac{\# \text{ of End Calls Completed}}{\# \text{ of Calls Attempted}} * 100 [\%]$$

Objective

The objective of this test is to attempt a steady call rate of 100 calls per second (cps) while applying a constraint of 8,000 simulated users. The test will also highlight how to apply a constraint of call hold time or a dual constraint of simulated users and call hold time.

The instructions provided for this test describe how to configure SIP activities to simulate SIP IP phones, SIP trunks, and SIP networks with users behind them. They will also helping the reader to understand how Ixia's IxLoad traffic generator can be configured to achieve a constant call rate.

Setup

In this example, a pair of Acceleron-NP ports are used. Calls will send voice traffic bi-directionally and will have a call hold time (conversation time) determined by the test objective. The test will run for five minutes.

The configuration simulates a SIP network (**Network1**) with 8,000 IP phones and a secondary SIP network (**Network2**), which includes one SIP Proxy Server and one Media Gateway with 8,000 phones behind it.

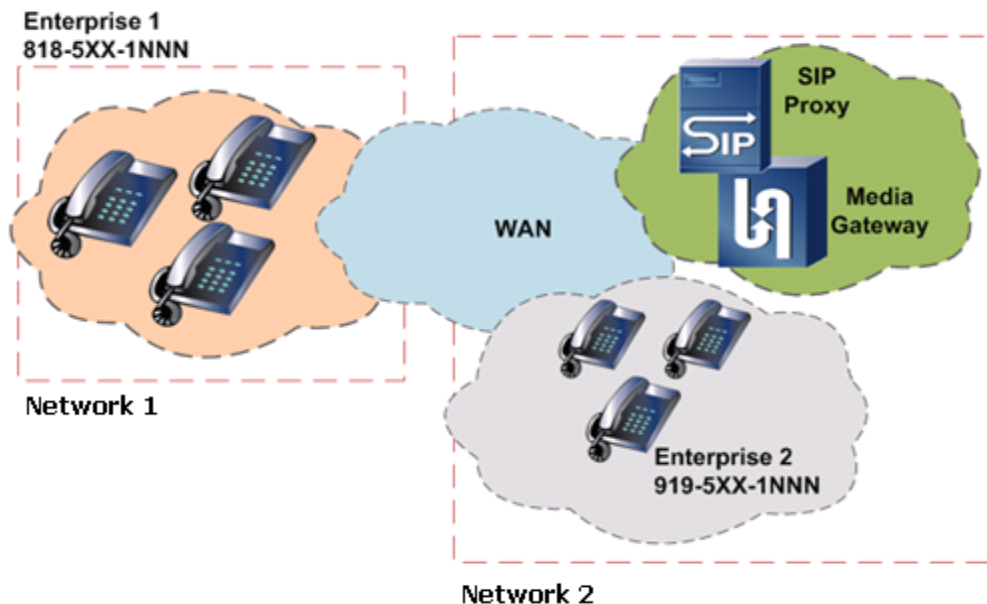


Figure 9. Test Topology – Ixia generates and receives the traffic passing through the WAN

Step-by-Step Instructions

The final IxLoad configuration as a result of these steps is provided on the blackbook.ixiacom.com Web site: *IxLoad 5.10 Voice - SIP Call Setup Rate.crf*. To import a Compressed Repository File (crf) in IxLoad, use the command **Import** under the **File** menu.

The step-by-step instructions highlight how to set the essential parameters of this configuration and explain additional options, which may be used to change the behavior of the test.

Open the Configuration Template

1. Open the IxLoad GUI.
2. Open the **VS_002_B2B_SIPv4 MakeCall - ReceiveCall - EndCall with RTP - 33s.rxf** configuration template included in **Getting Started | Templates | VoIPsip**.

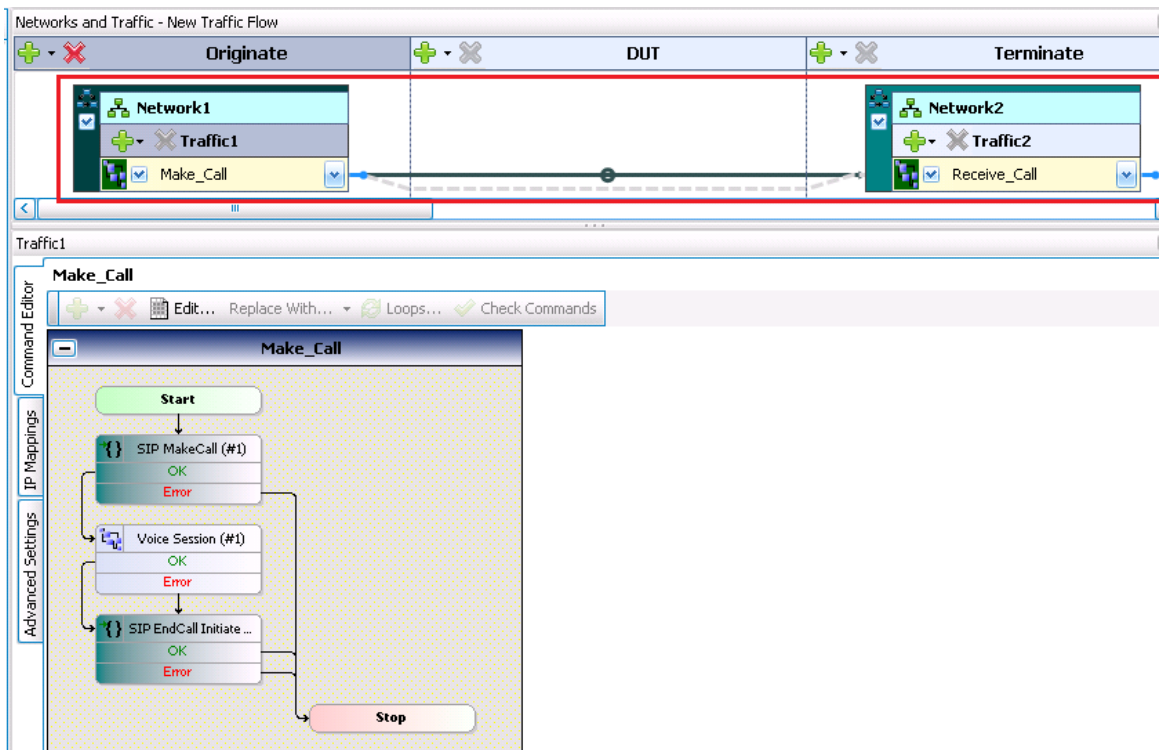


Figure 10. Overview of the IxLoad configuration sample

Configuring the Network Parameters for Network1

Note: This network hosts 8,000 SIP IP phones (SIP clients). Each simulated phone will use a distinct IPv4 IP address and a unique MAC address.

Select **Network1** to display the IP network ranges.

1. Set the following parameters:
 - a. **Stack: IP over MAC/VLAN**

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

- b. **IP Type = IPv4**
- c. **Count = 8,000**
- d. **Address = 20.1.1.1**
- e. **Mask = 8**
- f. **Gateway = 0.0.0.0**

Table 1 – Summary of Network1 parameters

IP Type	Count	Address	Mask	Increment	Gateway	Gateway Increment	MSS (RX)
IPv4	8,000	20.1.1.1	8	0.0.0.1	0.0.0.0	0.0.0.0	1460

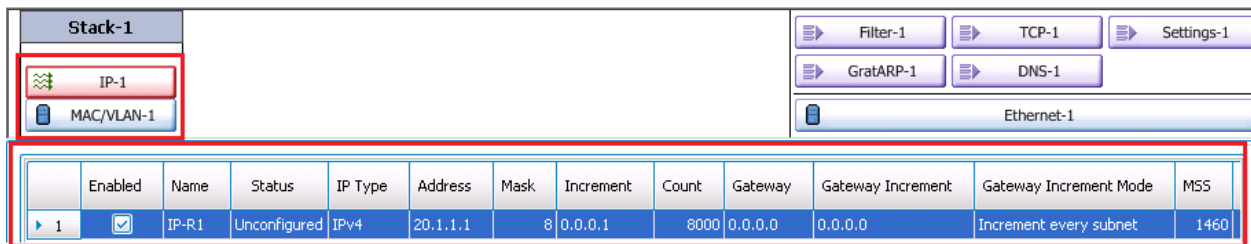


Figure 11. Network1: network configuration example for SIP IP Phones simulation

Notes: If the number of MAC addresses is higher than the maximum supported by the DUT, add an emulated router (right click the IP stack/Insert below/Emulated Router).

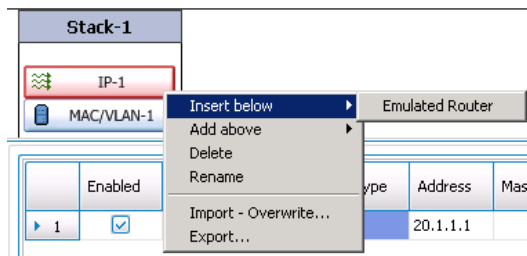


Figure 12. Network1: Add an emulated router

This will put the simulated phones behind an emulated router, thus exposing a single MAC address for all the traffic. The emulated router performs the same functions for the simulated subnet(s) that a real router performs for real subnets; it forwards the packets between networks. To configure the settings of the emulated router, select the **Emulated Router** stack. Each Ixia port you use in the test requires its own Emulated Router.

Configuring the Network Parameters for Network2

Note: This network will host one SIP proxy server and one large media gateway (RTP proxy), which routes the traffic to a group of 8,000 phones that they serve. The SIP proxy server will handle the SIP traffic and the media gateway will handle the RTP traffic. To separate the signaling IP address of the SIP proxy from the media gateway address, two separate network

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

ranges are required (one for SIP and one for RTP). The traffic configuration step will show how to associate network ranges with the SIP traffic or with the RTP traffic.

1. Select **Network2**; this will display the **IP Network Ranges**.
2. Add a secondary IP network range by clicking **Add Row**.
3. Set the network parameters as shown in the following table:

Table 1. Summary of Network2 parameters

Parameter	IP Range 1 (will be used for SIP)	IP Range 2 (will be used for RTP)
IP Type	IPv4	IPv4
Count	1	1
Address	20.1.100.1	20.1.200.1
Mask	8	8
Increment	0.0.0.1	0.0.0.1
Gateway	0.0.0.0	0.0.0.0
Gateway Increment	0.0.0.0	0.0.0.0
MSS (RX)	1460	1460

Note: To simulate a SIP trunk, which receives and generates both SIP and RTP traffic using a common IP address, set a single network range instead of two (that is, skip the preceding step 2).

Enabled	Name	Status	IP Type	Address	Mask	Increment	Count	Gateway	Gateway Increment	Gateway Increment Mode	MSS
<input checked="" type="checkbox"/>	IP-R2	Unconfigured	IPv4	20.1.100.1	8	0.0.0.1	1	0.0.0.0	0.0.0.0	Increment every subnet	1460
<input checked="" type="checkbox"/>	IP-R3	Unconfigured	IPv4	20.1.200.1	8	0.0.0.1	1	0.0.0.0	0.0.0.0	Increment every subnet	1460

Figure 13. Configuration example for Network2

Configuring Traffic 1 for Network1

The traffic element includes options that allow one or more IP ranges to be associated with one or more activities. Similarly, every VoIP activity allows signaling and media to use distinct ranges. The default settings associate the same IP address with SIP and RTP traffic.





Because **Network1** simulates SIP IP phones that generate SIP and RTP traffic using the same IP source, we can use the default settings.

Select **Traffic1** from **Network1**.

1. For **Network Range IP-R1** included in **Network1** (**Error! Reference source not found.**), verify that:
 - a. **Group1** uses the **Consecutive IPs** distribution rule.

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

b. In the **Activities** column, the **SIP** and **RTP** check boxes are selected.

Notes: Did you notice the  **Create New Group**,  **Remove Selected Group**,  **Edit/Modify Selected Distribution Group** and  **Move Up/Move Down** buttons? Try them and see how they work!

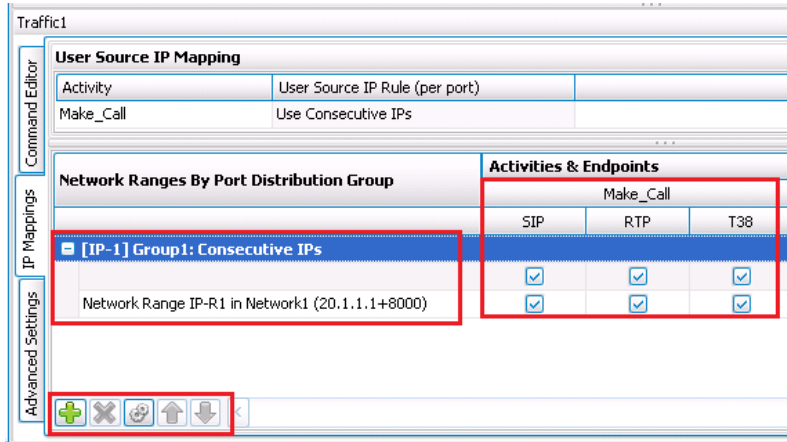


Figure 14. Traffic configuration example for SIP IP phone simulation

Configuring Traffic 2 for Network 2

This section describes how to map distinct network ranges for SIP traffic and RTP traffic when simulating a network where SIP and RTP traffic have separate IP sources. In our example, the SIP server and the media gateway components are separate entities and they use separate IPs. The steps highlight how the IP range 20.1.100.1+1 is associated with the SIP server component and 20.1.200.1+1 is associated with the media gateway component.

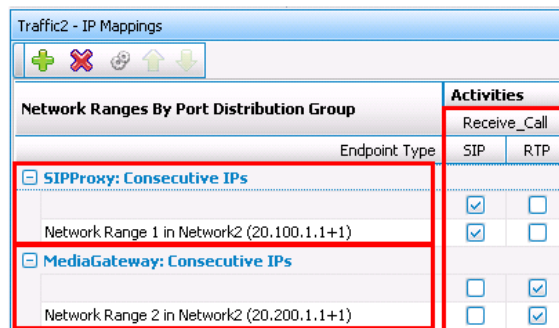


Figure 15. Configuration example for Traffic2 - IP mappings

1. Select **Traffic2** from **Network2**.
2. Click the **IP Mappings** tab.
3. For **Group1**:
 - a. Edit the name of the IP group from Group1 to **SIPProxy** and check that the group has **Consecutive IPs** rule set (right click the IP Group/Edit Group).

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

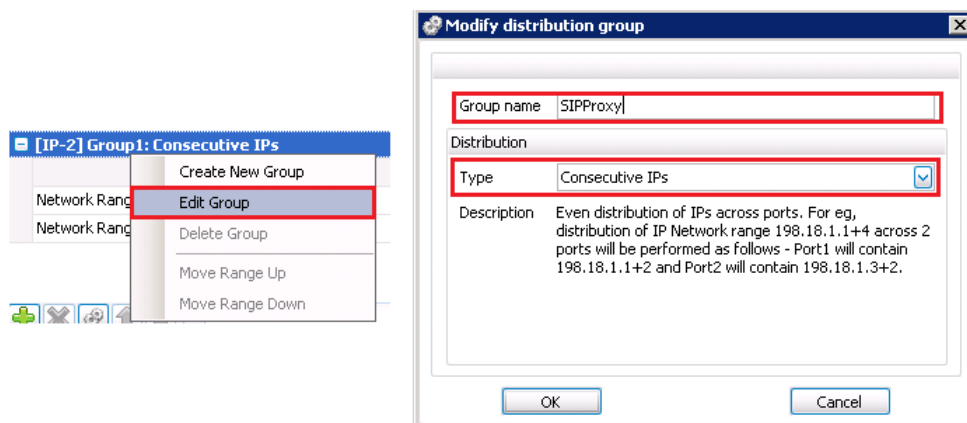


Figure 16. Modify the IP distribution group parameters

- b. Close the **Modify distribution group** by clicking **OK**.
- 4. Create a new Distribution Group.

Network Ranges By Port Distribution Group	Activities & Endpoints		
	Receive_Call		
	SIP	RTP	T38
[IP-2] SIPProxy: Consecutive IPs			
Network Range IP-R2 in Network2 (20.1.100.1+1)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Network Range IP-R3 in Network2 (20.1.200.1+1)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 17. Create a new distribution list

- 5. For the new created distribution group **DistGroup1**:
 - a. Edit the name of the IP group from DistGroup1 to **MediaGateway** and check that the group has **Consecutive IPs** rule set (right click the IP Group/Edit Group).
 - b. Close the **Modify distribution group** by clicking OK.
- 6. Move the **Network Range IP-R3** under the **MediaGateway** distribution group (select the Network Range and click the arrows to move it up or down).


Network Ranges By Port Distribution Group	Activities & Endpoints		
	Receive_Call		
	SIP	RTP	T38
[IP-2] SIPProxy: Consecutive IPs			
Network Range IP-R2 in Network2 (20.1.100.1+1)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Network Range IP-R3 in Network2 (20.1.200.1+1)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MediaGateway: Consecutive IPs			
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 18. Network Range

- 7. The **MediaGateway** group has the **Consecutive IPs** rule set.

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

- a. The network range included under **MediaGateway** has only the **RTP** check box selected.

Note: The traffic toolbar  allows you to create and delete **Port Distribution** groups and to select the IP distribution rule within a group (**Consecutive Ranges**, **Consecutive IPs** and **Round Robin**). You can move a **Network Range** from one group to another using the **Up/Down** buttons available in the same toolbar. In this example, the group labels are set to *SIPProxy* and *MediaGateway* for clarity.

Configuring a Test Scenario for the Calls Attempted per Second Objective

The media duration configured inside Scenario Editor plays a critical role in achieving the desired objective. To achieve the desired call rate for a given number of simulated users, the call hold time must not exceed certain limits.

IxLoad provides two working modes:

- **Auto control** of the media duration using the **TALK TIME** option that is available on all the RTP script functions (for example, RTP TALK, RTP Listen, RTP Voice Session).
- **Script overrides** of the TALK TIME duration. This is a programmed duration (constant across all calls), which can be configured on the media script objects; when set, it overrides the TALK TIME value calculated by the test objective

The **Calls per Second** objective can work in three different modes, explained below.

The first mode allows the media duration to take assume the values calculated by the **Calls Initiated per Second** test objective. In this mode, the call rate is kept constant for the entire test duration.

Requirement examples:

- Attempt the calls using a steady call rate of 100 cps for 8,000 users.
- Attempt the calls using a steady call rate of 100 cps with a talk time of 3 minutes.

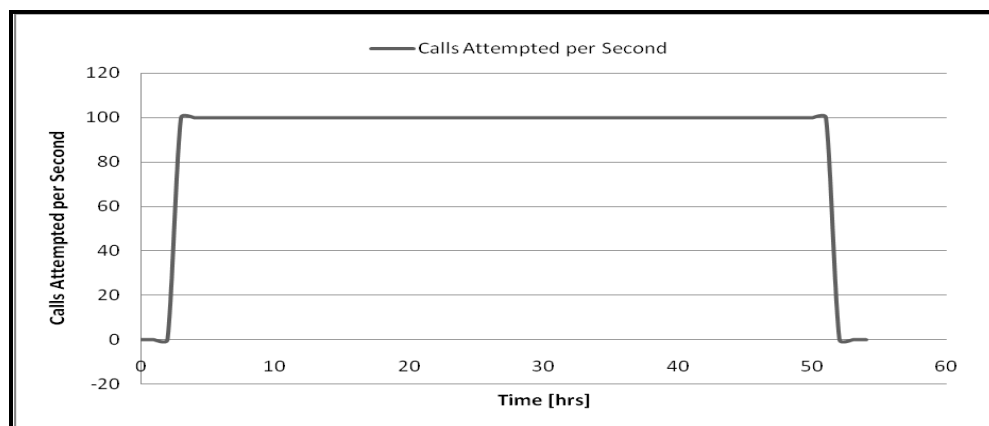


Figure 19. Example of constant call rate using Auto Control

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

The second mode is useful in some use cases when a very long conversation time is required (for example, 1 hour) and the available number of channels (phones) cannot achieve the desired rate. However, when the calls are generated they should be attempted at the constant call rate.

Requirement Example: using a pool of 8,000 phones and calls with a talk time of 1 hour, generate a steady call rate of 100 calls per second. In this case, the steady rate will be achieved only once every hour for a duration of 80 seconds (8,000 phones/100 cps = 80 sec). If the requirement imposes 100 cps with 1 hour talk time, then at least $3,600 \text{ sec} \times 100 \text{ cps} = 360,000$ phones will be required.

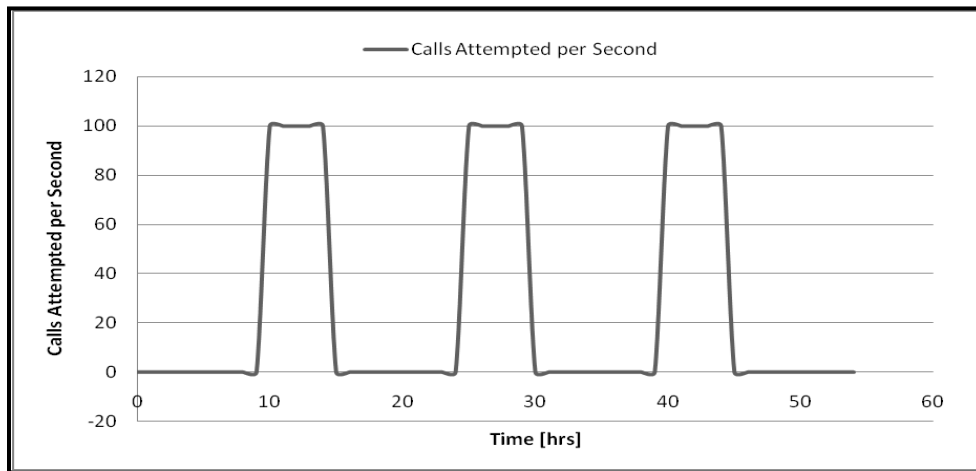
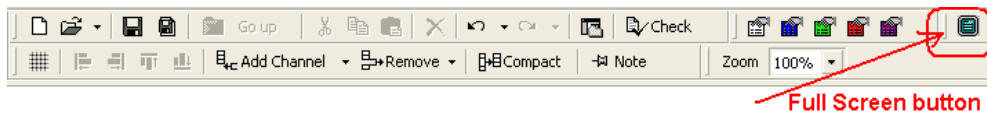


Figure 20. Example of constant call rate with limited number of channels available and long Talk Time

The duration of media functions can be specified at the activity level or at the function level (in the test scenario). We recommend you to use the activity level settings because these are exposed to TCL API; in automation environment, the control of call duration will be easily done without opening the test Scenario Editor. The following steps show how to set the parameters of Voice Session functions in the Scenario Editor to use the setting at the activity level.

1. In **Network1**, select the **SIP Peer** activity labeled **Make_Call**.
2. The configuration page displays **Scenario Editor**.
3. Click **Full Screen** in the **Scenario Editor** toolbar.



Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

4. Locate the **RTP Voice Session** script objects used by **Scenario Editor**. The scenario has two functions, one on the caller side and one on the called side.

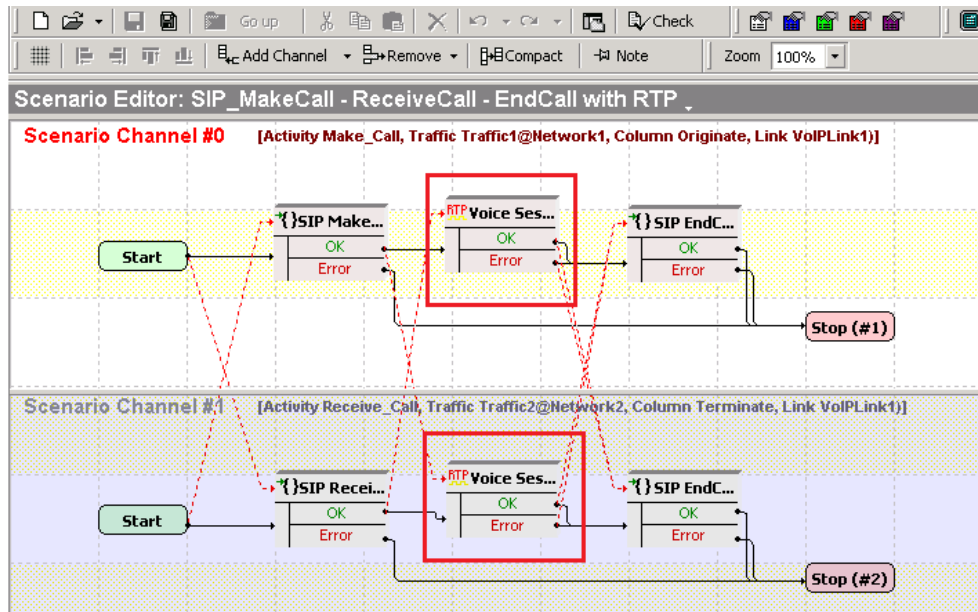


Figure 21. Scenario Editor in full screen mode - VOICE SESSION objects

5. For both **Voice Session** script objects included in the script, do the following:
 - a. Double-click the **Voice Session** script object, opening its properties.
 - b. In the **Talk Parameters** tab, clear the **Overwrite playback activity settings** check box.
 - c. Click the **Listen Parameters** tab, and then clear the **Overwrite playback activity settings** check box.
 - d. Click **OK** to close the properties page.
 - e. Remember to repeat these steps for the second voice session script object.

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

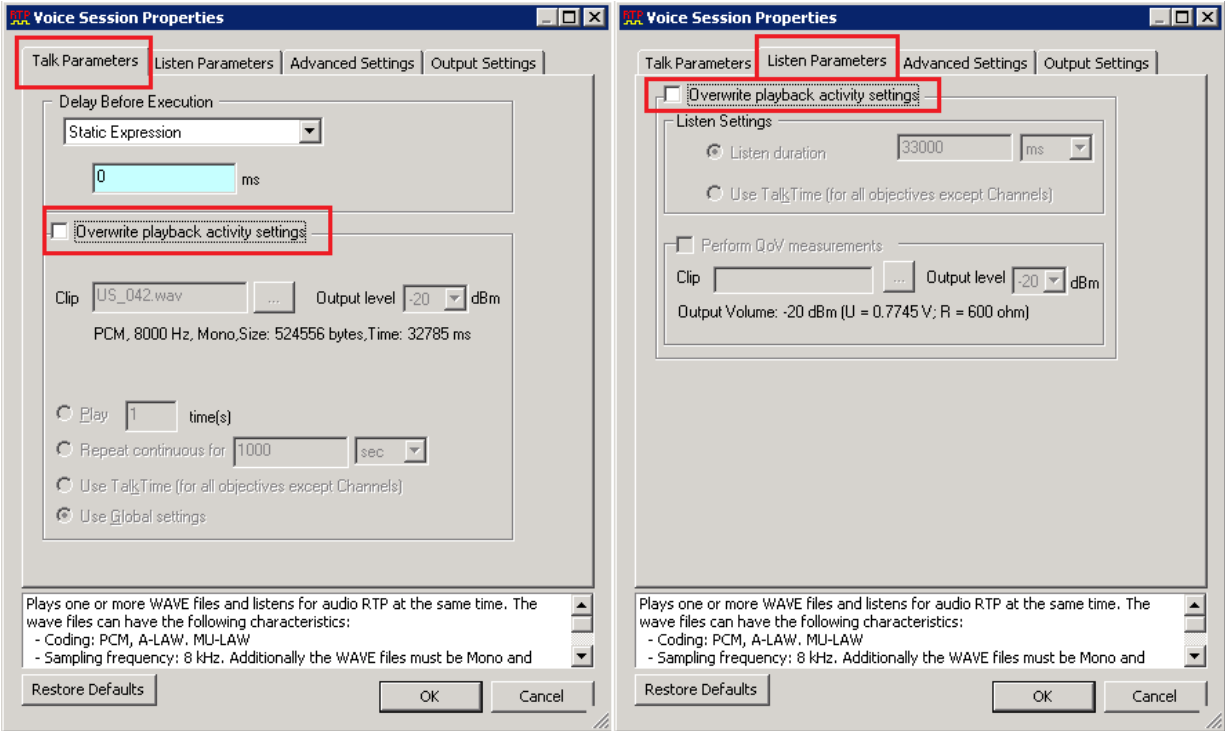
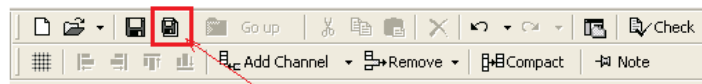


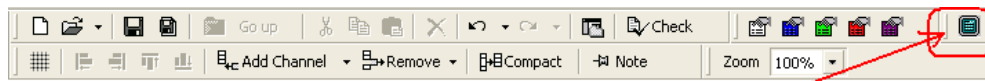
Figure 22. Voice Session configuration example for the Calls Attempted per Second objective

6. Save the test scenario flow as a **.TST** file by clicking **Save As** available in the **Scenario Editor** toolbar; change the location and the name of the **.TST** file (for example, 'SIP_MakeCall - ReceiveCall - EndCall with RTP.tst').



Save As button

7. Click **Full Screen** in the **Scenario Editor** toolbar. **Scenario Editor** exits the full screen mode.



Full Screen button

8. Save the IxLoad repository file as an **.RXF** file using the **File > Save As** menu option; change the location and the name of the **.RXF** file (for example, 'IxLoad Voice - SIP Call Setup Rate.rxf').

Configuring Execution Settings for the Make_Call Activity

1. Select the **Make_Call** activity.
2. Click the **Execution Settings** tab.

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

3. Set the script to continuously repeat during the **Sustain Time** (the **Sustain Time** is the test duration, to be set later in this example) using the **Run For the entire test duration** option.
4. Select the **Graceful Ramp-down** check box (this is the default setting).

Note: If the ramp-down event in the timeline occurs while the phone is in the middle of the call, the **Graceful Ramp-Down** option will stop the RTP traffic and will continue executing the next actions by skipping any media action and executing only signaling actions. In this way, the signaling flow will end by sending a BYE request that will gracefully disconnect the call. Calls without media may be reported by the test tool in the ramp down phase for calls initiated but not connected.

Traffic1 - Make_Call (VoIPsip Peer)

Scenario **Execution** Dial Plan SIP Automatic TLS Cloud Codecs RTP Audio Video Fax (T.38) Fax (T.30) SRTP Other

Run for
 the entire test duration
 a number of loops

Loop delays
Before 1st loop: ms
Between loops: ms

Aliases
Number of aliases (phone numbers) per channel:
NOTE: If more than one, aliases will cycle.

Channel mapping rules for SIP UA
IP address: Use consecutive values (per port)
UDP/TCP/TLS port: Use same value
Phone no: Use consecutive values (per port)
 Accept multiple channels sharing the same IP:Port

Channel mapping rules for media
IP address: Use consecutive values (per port)
Port: Use same value

Graceful Ramp-down

Figure 23. Configuration example for the Execution Settings tab of the SIP Make_Call activity

5. Channel mapping settings for SIP when simulating SIP IP Phones:
 - a. **IP Address = Use Consecutive Values (per port).**
 - b. **UDP/TCP/TLS port = Use same value.**
 - c. **Phone Number = Use Consecutive Values (per activity).**
6. Channel mapping rules for RTP when simulating SIP IP Phones:
 - a. **IP Address = Use Consecutive Values (per port).**
 - b. **UDP/TCP/TLS port = Use same value.**

Configuring the Dial Plan for the Make_Call Activity

1. Click the **Dial Plan** tab.
2. Set **Source Phone numbers** by selecting the **Specify Number/URI** check box.
3. Set the source phone number by clicking **Specify** and entering the sequence *818501[0001-
J*.
4. Set **Destination** to **Traffic2_Receive_Call:5060** by selecting the corresponding symbolic link from the list available for destination **IPs**.

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

Note: When the number dialed must be different than the numbers specified in the remote activity, the **Override phone numbers from destination activity** check box must be selected and a destination phone range must be specified.

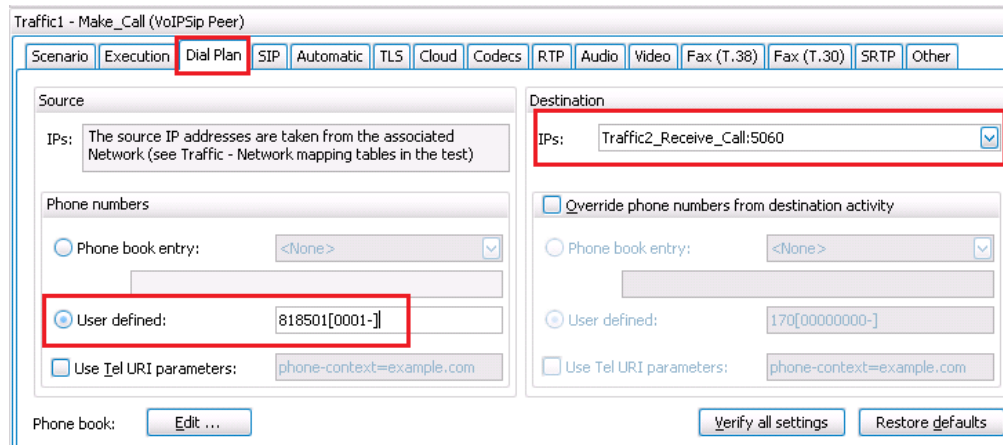


Figure 24. Dial plan settings

Configuring SIP Parameters for the Make_Call Activity

1. Click the **SIP** tab.
2. Set the **SIP Port** number to **5060**.

Note: This parameter will also accept a range of ports (for example, [5060-6060,2]).

3. For back to back tests, verify that the **Use External Server** check box is cleared.

If running against an external server use these settings instead (at the bottom in the following figure):

- Select the **Use External Server** check box.
- Set the **Server Address** option with the IP address of the SIP Server.
- Set the **Server Port** option with the port number used by the SIP Server (typically 5060). IxLoad will send the messages to the Server Address on port Server Port.
- Set the **Domain Name or Local IP** option to match the one set on the server (for example, *voice.ixiacom.com*).

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

Traffic1 - Make_Call (VoIPSip Peer)

Scenario Execution Dial Plan **SIP** Automatic TLS Cloud Codecs RTP Audio Video Fax (T.38) Fax (T.30) SRT

Enable signaling on this activity
(if unchecked, all SIP script functions will be SKIPPED)

SIP Port: 5060

Transport settings

Maximum message size on UDP: 1024

Override transport specified in scenario: UDP Only

TCP send immediate

Enable FQDN resolution

Authentication UAC

User name: Anonymous

Password:

AKA authentication settings
Select configuration: <None>

Edit configurations...

Type Of Service

TOS/DSCP: Best Effort (0x00)

Use external server

Server address:

Server port: 5060

Domain name or local IP:

Outbound proxy

Registrar server

Auto register simulated user agents

Override registrar IP:PORT

Construction of SIP messages

Override default contact settings Edit Contact ...

Override default destination domain name or host:port
Domain name or Host:Port:

Use Tel URI scheme for Source

Use Tel URI scheme for Destination

Transfer address: Edit ...

Verify all settings Restore defaults

Transport settings

Maximum message size on UDP: 1024

Override transport specified in scenario: UDP Only

TCP send immediate

Enable FQDN resolution

Use external server

Server address: 61.15.1.1

Server port: 5060

Domain name or local IP: voice.ixiacom.com

Outbound proxy

Registrar server

Figure 25. SIP Settings for back to back configuration (Top)
Sample configuration with a SIP Server (61.15.1.1) acting as an Outbound Proxy (Bottom)

Configuring Automatic Behavior for the Make_Call Activity

1. Click the **Automatic** tab.
2. Select the **Enable Retransmission**, **Ignore received retransmissions**, and **Retransmit ACK** check boxes and leave the default values for the **T1** and **T2** timers.
 - a. **T1** = 500 ms (default).
 - b. **T2** = 4000 ms (default).

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

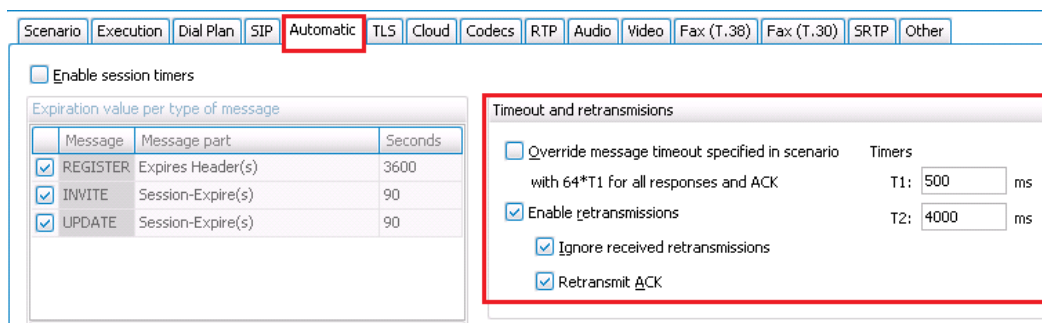


Figure 26. SIP Retransmissions settings

Configuring CODECs Settings for the Make_Call Activity

1. Click the **Codecs** tab.
2. Make **G.711 uLaw** the preferred CODEC by setting its position in the CODEC list to **1**.
3. Set RTP **Packet time** to **20 ms**, corresponding with a payload of 160 bytes per frame.

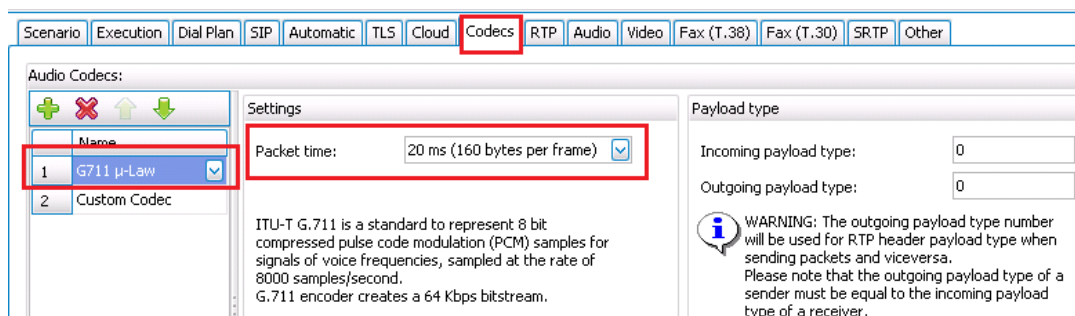


Figure 27. Configuring CODECs settings for the Make_Call activity

Note: If the required codec is not natively supported by IxLoad, the **Custom Codec** can be used; this feature allows playing of any RTP stream provided as a traffic capture.

Configuring RTP Settings for the Make_Call Activity

1. Click the **RTP** tab.
2. Ensure that you have selected the **Enable Hw Acceleration** check box to allow 8,000 RTP streams per Acceleron-XP port.

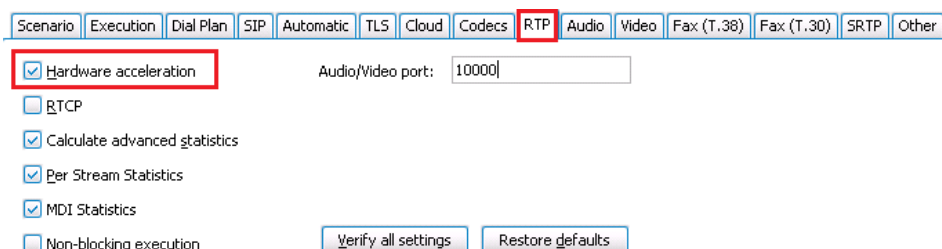


Figure 28. Configuring RTP settings for the Make_Call activity

Configuring Audio Settings for the Make_Call Activity

1. Click the **Audio** tab.
2. Ensure that you have selected the **Enable audio on this activity** check box.
3. Specify the clip to be played.
4. Specify the duration of the play—this will determine the call duration.
5. Select the **MOS** and **One Way Delay** check boxes if these metrics are of interest for the test that you want to configure.

Figure 29. Audio Settings

Note: The **Clip** and **Play Duration** settings may be overwritten by the Voice Session functions. In this example, the settings at the activity level (from this tab) are used because the Voice Session functions have the **Overwrite** check box cleared.

Configuring Execution Settings for the Receive_Call Activity

1. Click the **Receive_Call** activity.
2. Click the **Execution Settings** page.
3. Set the activity to run the script continuously during the **Sustain Time** using the **Run for the entire test duration** option. This will ensure that the CPS rate will be constant for the entire Sustain Time.

Note: The **Run for** parameter accepts a fixed number of loops as a parameter. This is particularly useful when you wish to force a number of loops per channel, forcing deterministic results. The Sustain Time must be set to **number of loops x Talk Time + 10 seconds** or more.

4. Select the **Graceful Ramp-down** check box (this is the default setting).

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

Traffic2 - Receive_Call (VoIPSip Peer)

Scenario Execution Dial Plan SIP Automatic TLS Cloud Codecs RTP Audio Video Fax (T.38) Fax (T.30) SRTP Other

Run for

the entire test duration

a number of loops 1

Loop delays

Before 1st loop: 0 ms

Between loops: 0 ms

Aliases

Number of aliases (phone numbers) per channel: 1

NOTE: If more than one, aliases will cycle.

Channel mapping rules for SIP UA

IP address: Use same value (per port)

UDP/TCP/TLS port: Use same value

Phone no: Use consecutive values (per port)

Accept multiple channels sharing the same IP:Port

Graceful Ramp-down

Channel mapping rules for media

IP address: Use same value (per port)

Port: Use consecutive values (per port)

Verify all settings Restore defaults

Figure 30. Configuration Example for the “Execution Setting” page of the SIP Receive_Call activity

5. Set the **Channel mapping rules for SIP** (when sharing the same SIP IP address and SIP port number for all the channels).
 - a. **IP Address = Use same value (per port).**

Note: This mode requires a unique RTP port for each simulated channel. The RTP port is specified as a range using a sequence generator [10000-65534,4] and can be configured using the RTP settings page (see Step #7 below).

 - b. **UDP/TCP/TLS port = Use same value.**
 - c. **Phone no = Use consecutive values (per activity).**
 - d. Enable **Accept multiple channels sharing the same IP:port.**
6. Set the **Channel mapping rules for RTP** (when using the same IP address for all streams).
 - a. **IP Address = Use same value (per port).**
 - b. **UDP port = Use consecutive values (per port).**

Configuring RTP Settings for the Receive_Call Activity

Because all of the media streams will originate from the same IP address, a range of RTP port numbers must be used. Use the **RTP** tab to set a unique RTP port number for each one. Remember that 8,000 RTP streams require 8,000 distinct port numbers with even values. The odd port value is for RTCP traffic.

1. Click the **RTP** tab.
2. Set the RTP Port to [10000-65534,2]. The range starts from port 10,000 and increments by 4 until it reaches 65,534. A step of 2 will work for audio sessions. For audio and video sessions, a step of 4 is required.

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

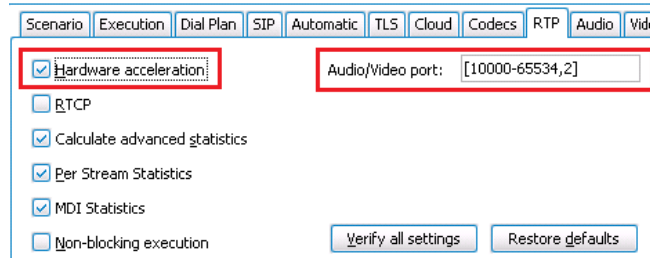


Figure 31. Activating the hardware RTP acceleration mode and RTP port configuration when all RTP streams are generated from a single IP address

3. Ensure that you have selected the **Enable Hw Acceleration** check box to allow 8,000 RTP streams per Acceleron-XP port.

Configuring the Dial Plan for the Receive_Call Activity

1. Click the **Dial Plan** tab.
2. Set the **Source Phone numbers** by clicking **Specify** and using the sequence *919501[0001-]*.
3. For **Destination IPs**, click **None** (no calls are generated from this activity with the test scenario used).

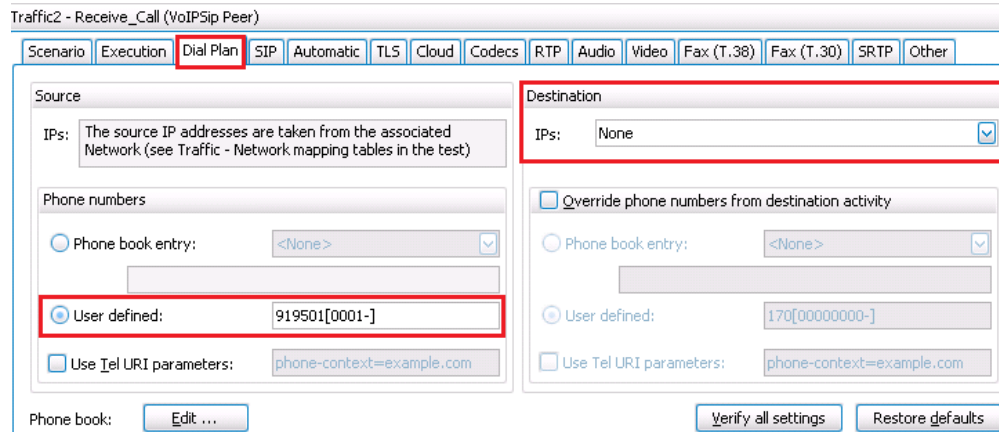


Figure 32. : Dial Plan settings for Receive_Call activity

Configuring the SIP Parameters for the Receive_Call Activity

1. Click the **SIP** tab.
2. Set the **SIP Port** number to **5060**.
3. For back to back tests, verify that the **Use External Server** check box is cleared.

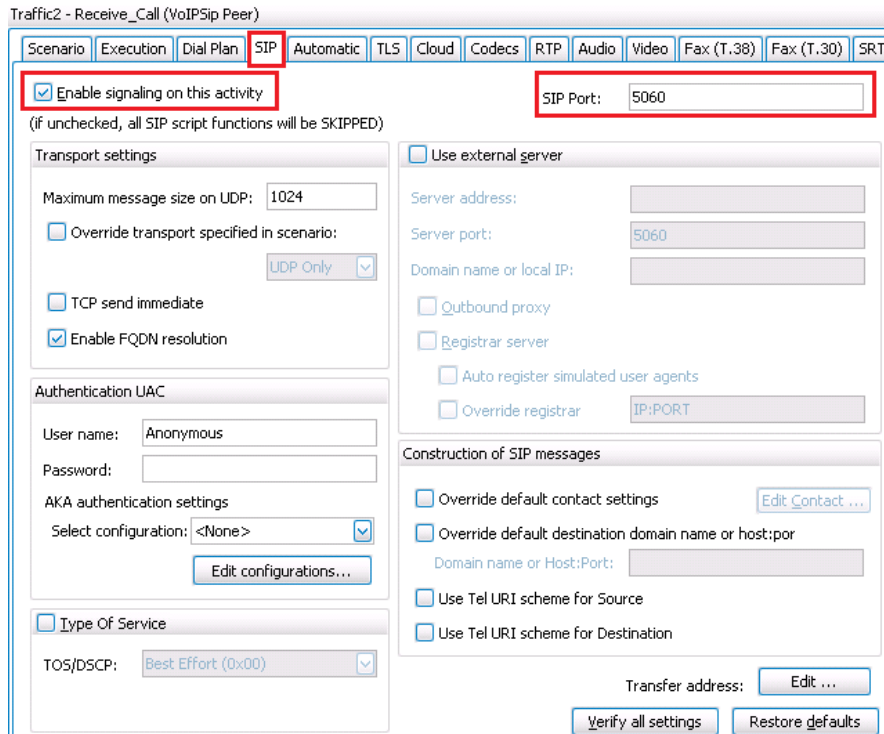


Figure 33. SIP Settings for back to back configuration

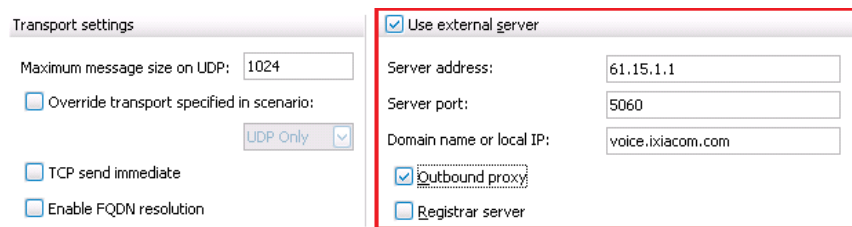


Figure 34. Sample configuration with a SIP Server (61.15.1.1) acting as an Outbound Proxy

Configuring Automatic Behavior for the Receive_Call Activity

1. Click the **Automatic** tab.
2. Select the **Enable Retransmission**, **Ignore received retransmissions**, and **Retransmit ACK** check boxes and leave the default values for the **T1** and **T2** timers.
 - a. **T1** = 500 ms (default).
 - b. **T2** = 4000 ms (default).

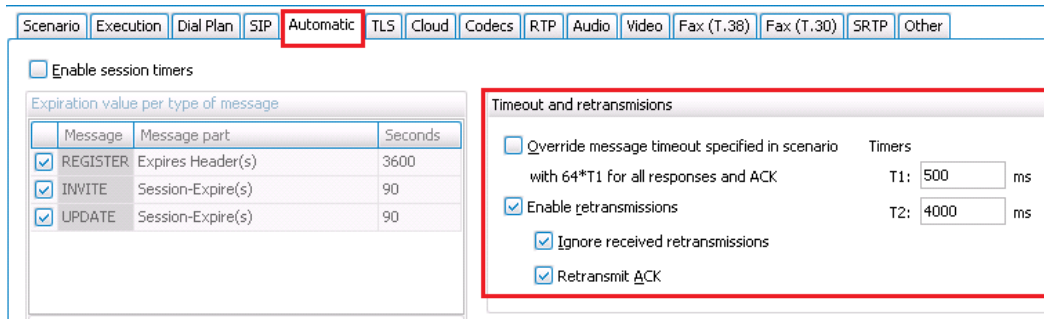


Figure 35. SIP Retransmissions settings

Configuring the Codecs Settings for the Receive_Call activity

1. Click the **Codecs** tab.
2. Make **G.711 uLaw** the preferred CODEC by moving it in the top of the Audio Codecs list its position in the CODEC list to 1.
3. Set the RTP **Packet time** to **20 ms (160 bytes per frame)**.

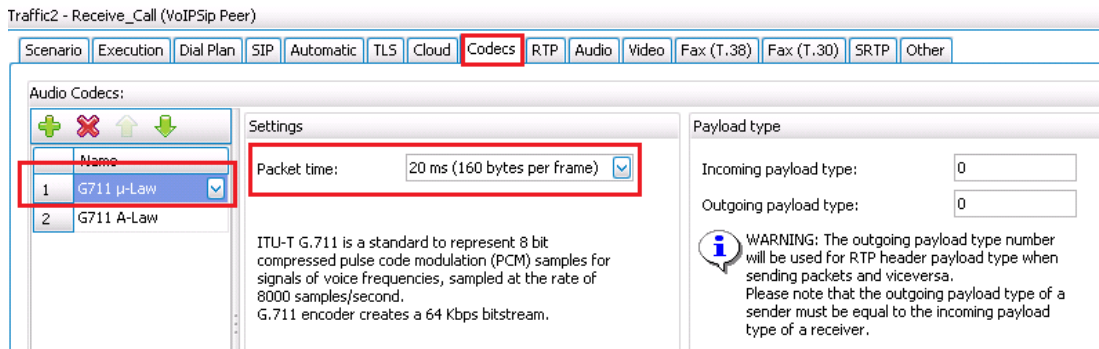


Figure 36. Configuring CODECS settings for the Receive_Call activity

Configuring Audio Settings for the Receive_Call Activity

1. Click the **Audio** tab.
2. Ensure that you have selected the **Enable audio on this activity** check box.
3. Specify the clip to be played.
4. Specify the duration of the play—this will determine the call duration.
5. Select the **MOS** and **One Way Delay** check boxes if these metrics are of interest for the test you want to configure.

Enable audio on this activity (if unchecked, all audio script functions will be SKIPPED)

Play Settings

Clip: US_042.wav

Format: PCM, Duration: 32785 ms, Size: 524556 bytes

Output level: -20 dBm

Play for clip duration or TalkTime (all objectives except Channels)

Play for: 10 Seconds

Type Of Service

TOS/DSCP: Class 1 (0x20)

Perform MOS

Calculate One Way Delay

Enable jitter buffer

Buffer size: 20 ms

Use compensation

Max. size: 1000 ms

Max. dropped consecutive packets: 7

Perform QoS

Units: # of Channels

Value: 100

Channel Selection: First Channels

Generate silence

Null data encoded Comfort noise

Figure 37. Audio Settings

Configuring the Timeline and Objective

1. Select **Timeline & Objective** from the test configuration panel.
2. Set the test **Objective Type** to **Calls Initiated per Second**.
3. Set the test **Objective Value** to **100**.
4. On the **Timeline** tab, set the **Ramp Up Value** to **100**.
5. Set the **Ramp Up Interval** to **1 second**.
6. Set the **Sustain Time** to **5 minutes**.
7. Set the **Ramp Down Time** to **40 seconds**.

Network Traffic Mapping	Objective Type	Objective Value	Timeline	Iteration Time	Total T
<ul style="list-style-type: none"> [-] New Traffic Flow <ul style="list-style-type: none"> [-] Activity Links <ul style="list-style-type: none"> VoIPLink1 <ul style="list-style-type: none"> Make_Call@Network1 Receive_Call@Network2 	Calls Initiated Per Second	100	Timeline1	000:05:45	
	Calls Initiated Per Second	100	Timeline1	000:05:45	
	Calls Initiated Per Second	100	Timeline1	000:05:45	

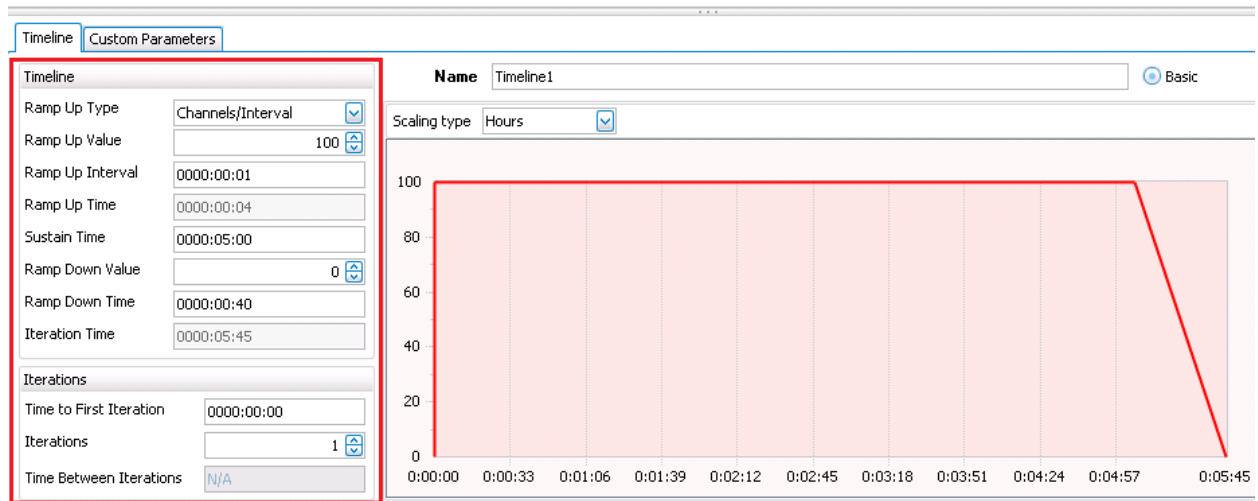


Figure 38. Configuration example for 100 calls per second

8. Under **Network Traffic Mapping**, select **VoIPLink1**, and then click the **Custom Parameters** tab.

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

9. Within the **CPS Objective** parameters group, select:
 - a. **Specify Number of channels = 8,000.**
 - b. **Estimated Overhead Time = 10,000 ms.**
 - c. **Minimum Channel Inter-Call Duration = 5,000 ms.**

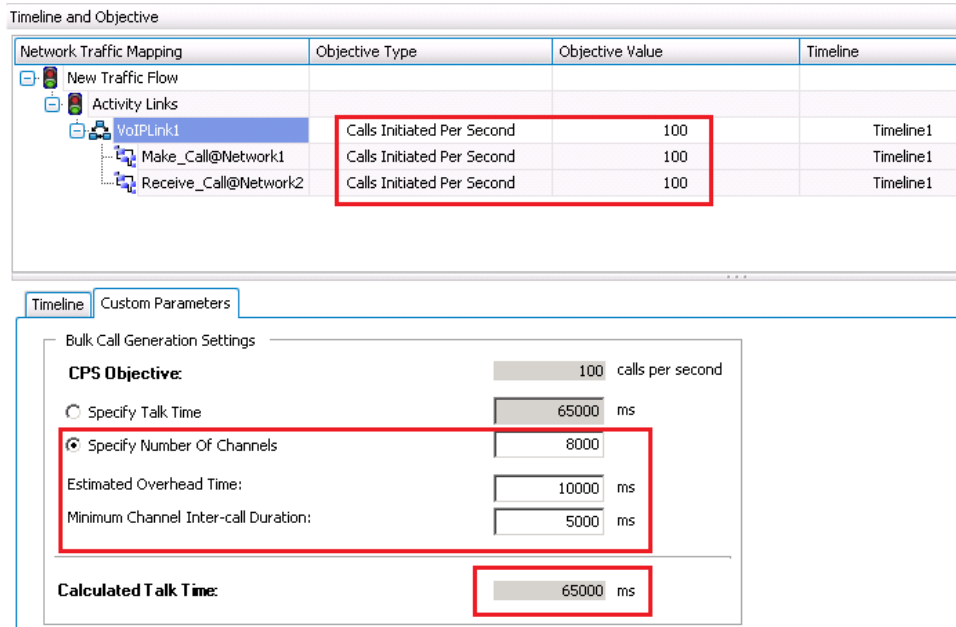


Figure 39. Configuration example for 100 cps with a test objective constraint of 8,000 channels

Notes:

- Based on the values defined in the **Custom Parameters** tab and the test **Objective Value**, IxLoad determines the call duration to be used at run time.
- The **Estimated Overhead Time** includes an estimation of the **Call Setup Time** and **Call Teardown Time**.
- The **Minimum Channel Inter-Call Duration** enforces a minimum time between two calls that will be generated by the same phone; this is required by some applications.
- In addition to the Talk Time option available as a configuration parameter on the RTP script objects, the calculated talk time provided by the test objective is also available as a variable named **\$TalkTime**, which is read-only. This variable can be used with the **SLEEP** script object, which can accept it as a parameter (for example, SLEEP (\$TalkTime)) allowing control of the call rate using configurations without RTP. In this case, the SLEEP script object replaces the Voice Session script objects used in this configuration.

Configuring the Test Options

1. From the **Test Configuration** panel, click **Test Options**.
2. Select the **Forcefully Take Ownership** check box.
3. Select the **Reboot Ports before Configuring** check box.
4. Set **CSV Polling Interval** to 1 second.

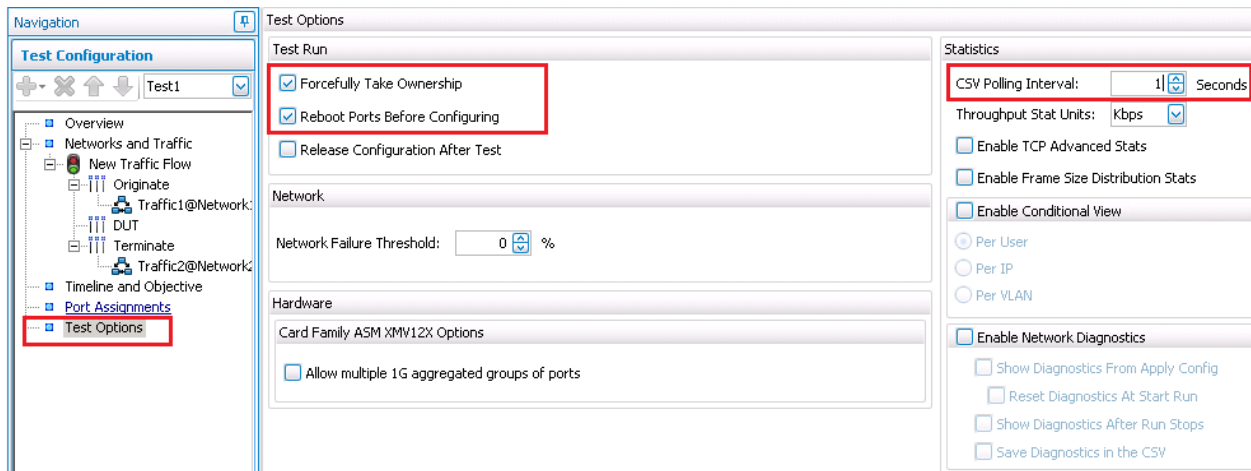


Figure 40. Configuration example for Test Options

Configuring Port Assignments

1. From the **Test Configuration** panel, click **Port Assignments**.
2. Add your chassis.
3. Assign the port(s) for **Network1** hosting the **Make_Call** activity

- Assign the port(s) for **Network1** hosting the **Receive_Call** activity.

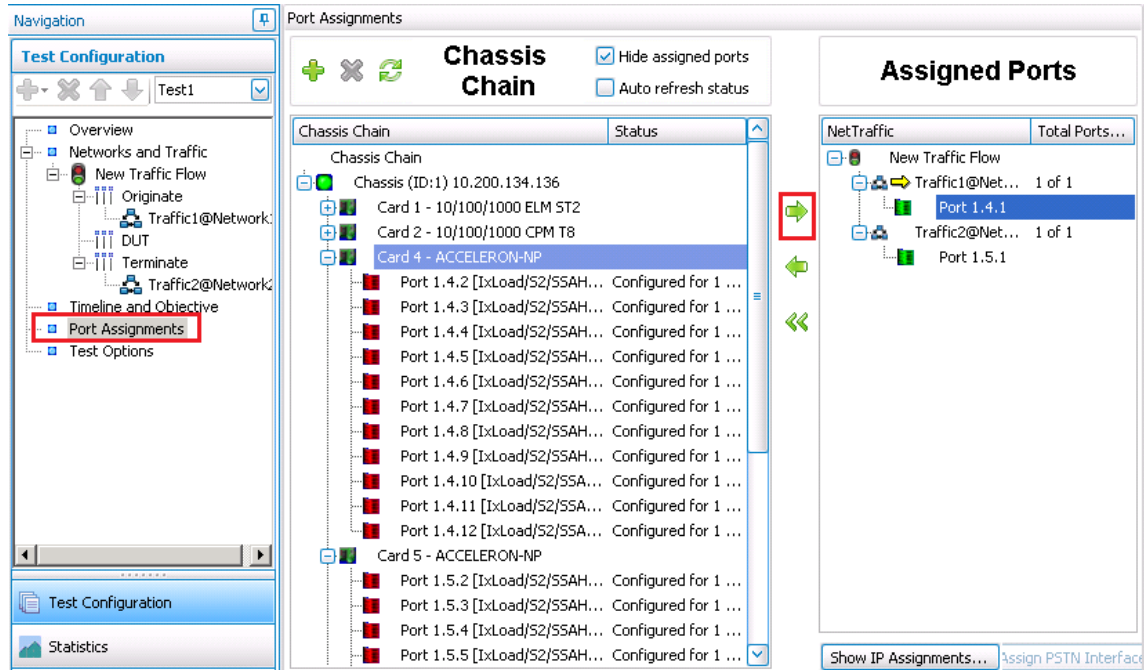


Figure 41. Port assignments page

Running the Test

- Click **Run** to start the test execution.
- IxLoad will automatically display **Statistic Views** after execution starts.

Results Analysis

The following questions provide guidelines on how to recognize specific problems during or at the end of the test execution:

- Has the test objective been achieved? Check the **Call Rates** view.

Table 2. Call Rate statistics

Statistic Name	Value	Questions
Calls Attempted per Second		1. Have the calls been attempted continuously at a constant call rate during the Sustain Time?
Calls Connected per Second		2. How do the Calls Attempted rate and the Calls Connected rate compare to each other?

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

2. Have any call failures been reported? Check the **Calls** view.

Table 3. Call statistics

Statistic Name	Value	Questions
Calls Attempted		1. Have any call attempts failed? Compare: a. Calls Attempted and Calls Received, with b. Calls Attempted and Calls Connected.
Calls Connected		
Calls Received		
Calls Answered		
End Calls Initiated		1. Have any attempts to hang up the call failed? Compare: a. End Calls Initiated and End Calls Received, with b. Calls Attempted and End Calls Received.
End Calls Received		
End Calls Completed		
		2. Have all the calls attempted ended successfully? Compare: [2 * Calls Connected] with [End Calls Completed]
		3. Why do we need to compare End Calls Completed with twice the number of calls connected?

3. Have any scenario loop failures been reported? Check the **Loops** statistics view.

Table 4. Statistics highlighting the pass/fail result based on call flow execution

Statistic Name	Value	Questions
Total Loops		1. Are the Successful Loops and Total Loops values equal?
Successful Loops		
Failed Loops		2. Have any Failed Loops, Aborted Loops or Warning Loops been reported? Note: failed/aborted and warning loops highlights failures at the scenario level.
Aborted Loops		
Warning Loops		

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

4. Has the QoS for signaling met the expected quality? Check the **Call Times** and **Delays** statistic views. Use the maximum value reported.

Table 5. Statistics used to determine the QoS for the SIP signaling

Statistic Name	Value (max /avg/min)	Questions
Call Setup Time		1. Is the maximum Call Setup Time less than 4 seconds? 2. Is the maximum End Call Time less than 2 seconds? 3. Is the maximum Media Delay (Tx or Rx) less than 4 seconds? 4. Is the maximum Post Dial Delay less than 2 seconds? 5. Is the maximum Post Pickup Delay less than 2 seconds? 6. For all the stats listed in this table, compare their value distribution in time.
End Call Time		
Talk Time		
Media Delay TX/RX		
Post Dial Delay		
Post Pickup Delay		
Note: Another important factor in establishing the quality of the signaling is the number of retransmissions. IxLoad counts those using the SIP Retransmitted Msgs statistic, located under the SIP Messages view.		

5. Has the QoS for media met the expected quality? Check the **RTP MOS RTP QoS, RTP Advanced QoS, RTP Jitter Distribution, RTP Consecutive Lost Datagram Distribution,** and **RTP Streams** statistic views.

Table 6. MOS statistics

Statistic Name	Value(s)	Questions
RTP MOS Best RTP MOS Worst	Max = _____ Min = _____	1. How do the last values reported by the RTP MOS Best and RTP MOS Worst compare with each other? 2. How does the RTP MOS Worst score compare with the max theoretical score for the CODEC used?
RTP MOS Instant (Best/Avg/Worst)	Max = _____ Min = _____ Avg = _____	1. Are any times without an instantaneous MOS value? 2. How frequent are the changes in the instantaneous MOS values?
RTP MOS Per Call (Best/Avg/Worst)	Max = _____ Min = _____ Avg = _____	1. How do the MOS per Call statistics compare with the RTP MOS Best and RTP MOS Worst statistics?

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

Table 7. Basic RTP QoS statistics (see RTP QoS and RTP Advanced QoS statistics views)

Statistic Name	Value(s)	Questions
RTP Packets Sent RTP Packets Received RTP Packets Lost		<ol style="list-style-type: none"> Are there any differences between RTP Packets Sent and RTP Packets Received? Does the difference match the value of RTP Lost Packets?
RTP One Way Delay [us]		<ol style="list-style-type: none"> Is the One Way Delay higher than 100 ms?
RTP Delay Variation Jitter [us] RTP Interarrival Jitter [us]		<ol style="list-style-type: none"> What is the max Delay Variation Jitter? What is the max Interarrival Jitter?

Table 8. RTP Jitter distribution statistics

Statistic Name	Value(s)	Questions
Packets with Delay Variation Jitter up to 1 ms		<ol style="list-style-type: none"> Assuming Jitter was reported, what is the distribution of the Delay Variation Jitter values?
Packets with Delay Variation Jitter up to 3 ms		
Packets with Delay Variation Jitter up to 5 ms		
Packets with Delay Variation Jitter up to 10 ms		
Packets with Delay Variation Jitter up to 20 ms		
Packets with Delay Variation Jitter up to 40 ms		
Packets with Delay Variation Jitter over 40 ms		

Table 9. Distribution of RTP Consecutive Lost Packets

Statistic Name	Value(s)	Questions
Consecutive Loss of One Packet Sequence		<ol style="list-style-type: none"> Assuming that packet loss was reported, what is the distribution of the lost RTP packets?
Consecutive Loss of Two or Three Packet Sequences		
Consecutive Loss of Four or Five Packet Sequences		
Consecutive Loss of Six to Ten Packet Sequences		
Consecutive Loss of Eleven or More Packet Sequence		

Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems

Table 10. RTP Streams

Statistic Name	Value(s)	Questions
Concurrent RTP Streams		<ol style="list-style-type: none"> 1. Assuming that packet loss was reported, what is the distribution of the lost RTP Packets? 2. Are any calls without RTP? 3. Are any calls with RTP? 4. Does this number match the number of Calls Connected * 2?
Concurrent RTP Streams (max)		
Number of calls with incoming RTP packets		
Number of calls without incoming RTP packets		

Troubleshooting and Diagnostics

The following table summarizes some of the common issues that may be encountered when running a call rate test.

Table 11.

Issue	Troubleshooting Solution
The attempted call rate is not sustained for the entire test duration	<p>Check the SIP Retransmissions counter, Call Setup Time, and End Call Time measurements. A high number of retransmissions is an indication that the SUT cannot maintain the load generated. This leads to larger call setup and teardown times, which affects the number of users available to place new calls.</p> <p>Increasing the Estimated Overhead Time parameter of the Calls Initiated per Second objective can force the tool to maintain a higher number of users available to place calls by decreasing the value of the estimated Call Hold Time (Talk Time). Hence, the call duration will be shorter and the user will become available faster.</p>
Calls are attempted at a constant call rate only from time to time followed by intervals without any calls attempted	<p>This issue is common when the Talk Time set in the script is hardcoded to a value larger than the estimated Talk Time provided by the test objective.</p> <p>Correct the problem by configuring the media script objects to play for the "Talk Time" duration calculated by the test objective, or use a hardcoded value lower than the one estimated by the test objective.</p>
The attempted call rate is not constant (has large variations)	<p>Verify that the Call Rate Attempted does not exceed the published performance. Ixia provides several cards that may have a lower or higher performance.</p> <p>Increase the Estimated Overhead Time value exposed by the Calls Initiated per Second objective.</p>
Call rate is constant but the calls are attempted at a lower rate than I configured in the test objective	<p>Verify that the Ramp Up Value and Ramp Up Interval parameters available in the timeline configuration of the test objective creates users at a rate equal to or higher than the CPS rate configured by the objective.</p>

Test Variables

Test Tool Variables

Table 12.

Parameter Name	Current Value	Additional Options
IP Version	IPv4	IPv6
Calls Attempted Rate	100	Up to 3,000 cps per XMV12-RTP card
User Constraint	8,000	Up to 96000 per XMV12-RTP card
Objective constraint for the Calls Attempted per Second	User Constraint	Talk Time constraint (call hold time) Scenario Call Hold constraint
Network Configuration	IP Only (Static IPs)	IP/DHCP, IP/IPsec, IP/PPPoE
SIP Transport	SIP over UDP	SIP over TCP, SIP over TLS/TCP
Codec parameters for the negotiated audio CODEC (type, packet size & frequency)	G.711u, 160 bpf every 20ms	G.711, G.729, G.723, G.726, iLBC, AMR
Mix with data protocols (for example, FTP, HTTP, Telnet)	Not included	Any combination of data protocols supported by IxLoad
Voice Call Flow	RFC 3261 standard call flow	IMS Call Flows (PRACK) Custom Call Flows (specific to a particular DUT)
Secure RTP	Off	On
IP Mapping rules	N to 1	1 to 1, 1 to N, N to 1 and N to N
TOS/DSCP marking for SIP & RTP	Off	On (individual settings for SIP & RTP)

DUT Test Variables

Table 13.

Parameter Name	Current Value	Additional Options
IP version	IPv4	IPv6
Transport Protocol	SIP/UDP	SIP/TCP, SIP/TLS
Dial Plan size	-	-
Media Transcoding		

Conclusions

This configuration covered the main parameters of the SIP Peer activity using a practical example allowing the user to control the call rate while running performance tests. The results section covered the main statistics that may highlight an issue.

Test Case: Determining the Maximum Number of Concurrent Calls

Determining the maximum number of active calls with media is necessary for devices and networks that handle media traffic, where media may include text, audio, video or fax. A few examples of such devices are:

- Session border controllers
- Application layer gateways
- Media gateways
- Voice mail servers
- Interactive voice response systems
- Media transponders
- Media proxy servers

The primary performance metric is the number of active calls that can be maintained by the DUT for a long period without having call failures or affecting the user experience.

The following standard metrics can help assess the QoS for signaling protocols:

- **Call setup time** – measures the time required to setup the call, including the acknowledgement from the called party. This metric has significance if the called party answers immediately. In practice, someone may answer after a variable time.
- **Post dial delay** – measures the time required to receive the first ring back notification after the last digit of the destination phone number was dialed.
- **Post pickup delay** – measures the time between when the call is answered until the first media packet is received.
- **Media delay** – measures the time to receive the first media packet after the call setup message was sent.
- **End call time** – measures the time to tear down a call.
- **Message retransmissions** – a count of the number of retransmissions under ideal conditions, that is, no packet loss or delay.

Test Case: Determining the Maximum Number of Concurrent Calls

When the DUT simultaneously handles signaling and media (for example, SBCs), it is important that media traffic be enabled for all calls. In these test cases, you must check that media traffic meets QoS requirements by inspecting the following measurements:

- Assessing the transport of media
 - Packet loss, jitter, and one-way delay
 - Packets sent versus packets received
 - Inbound/outbound throughput
 - Maximum consecutive loss, packet loss distribution, packet loss correlation
 - Packet errors, packet duplicates
 - Calls without media
- Audio quality metrics
 - R-Factor
 - MOS
 - PESQ (obsoletes PAMS & PSQM)
- Video quality metrics (when applicable)
 - V Factor
 - Absolute MOS-V
 - Relative MOS-V
 - Video loss degradation
 - Video jitter degradation
 - Video CODEC degradation

Determining the Maximum Number of Concurrent Calls for H.323-Based Devices and Systems

Overview

The goal of this test methodology is to help you determine the peak capacity, measured in active calls, supported by a DUT that uses the H.323 protocol. The DUT may include one or more of H.323 application layer gateways, SBCs, H.323 gateways, signaling gateways, IP softswitches, and IP-PBXs.

While the primary metric is the number of calls that can be sustained by the DUT, you must keep the user experience in the desired range.

Test Methodology

- Set the performance goals: max active calls, calls completion percentage, post dial delay, post pickup delay, TCP retransmissions, media delay, MOS, PESQ, packet loss percentage, jitter, and one-way delay.
- Use a binary search to determine the maximum call rate using short calls with duration of at least 15 min (30 min recommended).
 - Verify whether:
 - The calls connect without call failures.
 - The H.323 QoS meets expectations.
 - The RTP QoS meets expectations.
 - Plot a chart of active calls versus QoS.
 - Plot a chart of active calls versus DUT CPU utilization.
 - Plot a chart of active Calls versus DUT RAM utilization.
- Repeat the test for every supported audio CODEC.
- For every CODEC, repeat the test for all packet times supported by the DUT. The performance can differ dramatically between a ptime of 5 ms and a ptime of 30 ms.
- After the maximum capacity is determined, a system characterization can be made using longer execution time (for example, minimum 72 hours) to confirm that the system maintains the QoS in the expected range.
- To plot the performance characteristic of the DUT, at least 10 data points are recommended for the chart: 5 percent, 10 percent, 25 percent, 40 percent, 50 percent, 65 percent, 75 percent, 85 percent, 95 percent, and 100 percent of the maximum supported rate.

Test Case: Determining the Maximum Number of Concurrent Calls

The performance characterization of the DUT should continue by changing test variables that influence the performance of the DUT. Those test variables include:

- H.323 call setup mode: normal call, fast start, tunneling, H.245 in parallel with fast start
- Negotiated CODEC
- IP version (IPv4/IPv6)
- IP and port mapping for signaling and media (1:1, 1:n)
- Call duration
- Number of CODECs supported per user.

If the media traverses the DUT, the negotiated CODEC type must also be considered. The CODEC used is even more important when the DUT acts as a transcoder (for example, converts the voice from G.711 to G.729).

Objective

The test objective is to connect 8,000 calls, keeping them active for 5 minutes. This simulation will use 8,000 users, with every phone generating a single call. Using a call rate of 100 cps results in all the calls being active after 80 seconds (= 8,000 calls/100 cps).

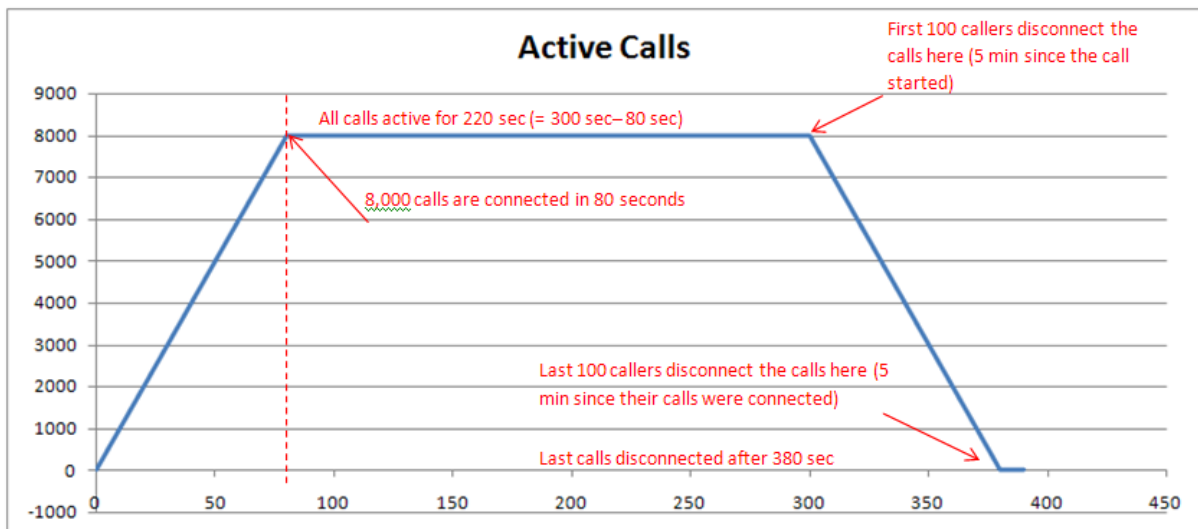


Figure 42. Max active calls in time

Test Case: Determining the Maximum Number of Concurrent Calls

Setup

The configuration simulates two H.323 networks, each hosting 8,000 IP phones. The calls will send voice traffic bi-directionally and will have a constant call hold time of 5 minutes. This example uses a single pair of Acceleron-XP ports.

In this example, we will assume that the phones in **Network1** originate the calls and the phones in **Network2** receive the calls and then immediately answer them.

The next figure highlights the test setup:

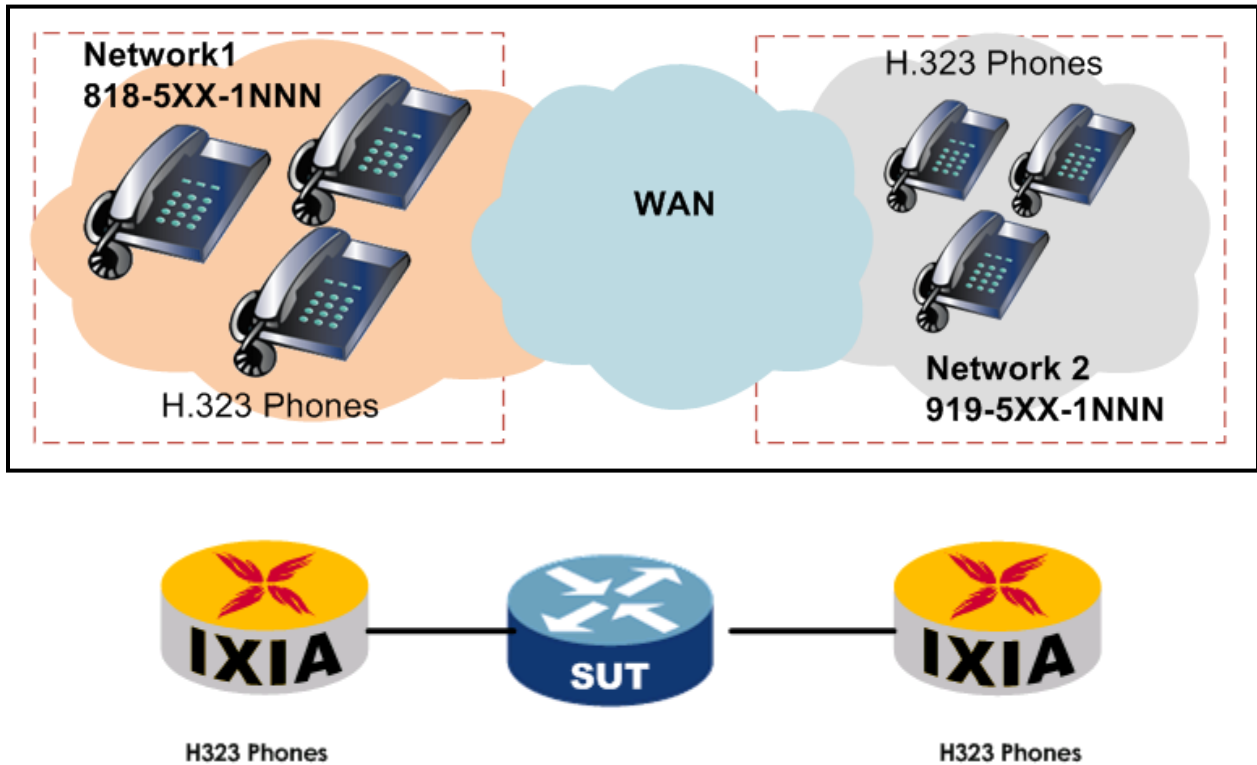


Figure 43. : Test setup

Step-by-Step Instructions

The final IxLoad configuration as a result of these steps is provided on the blackbook.ixiacom.com Web site - see *IxLoad 5.10 Voice - H323 Concurrent Calls.crf*. To import a Compressed Repository File (crf) in IxLoad, use the command **Import** under the **File** menu.

The step-by-step instructions include further details for configuring the primary parameters of the test, and explain additional options that can change the behavior of the test.

Open the Configuration Template

1. Open the IxLoad GUI.
2. Open the **VH_002_B2B_H323v4_NC_Basic_Call_with_RTP.rxf** configuration template included in **Getting Started | Templates | VoIPH323**

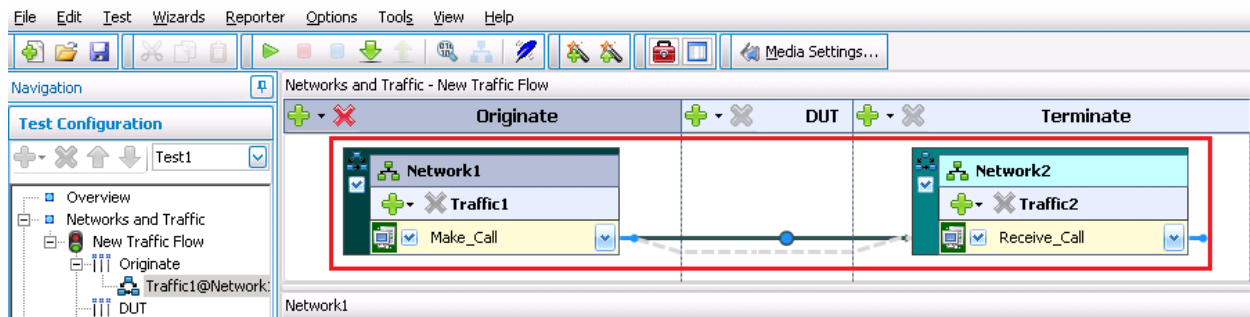


Figure 44. Overview of the IxLoad configuration sample

Configuring the Network Parameters for Network1

Note: This network will host 8,000 H.323 IP Phones. Each simulated phone will use a unique IPv4 address and MAC for both H.323 and RTP traffic.

1. Click **Network1** to display the IP Network Ranges.

Test Case: Determining the Maximum Number of Concurrent Calls

2. Set the following parameters:
 - a. **IP Type = IPv4**
 - b. **# Hosts = 8,000**
 - c. **First IP/Subnet = 20.1.1.1**
 - d. **Mask = 8**
 - e. **Increment = 0.0.0.1**
 - f. **Gateway = 0.0.0.0**
 - g. **Gateway Step Size = None**

Table 14. – Summary of Network 1 parameters

IP Type	Count	Address	Mask	Increment	Gateway	Gateway Increment	MSS (RX)
IPv4	8,000	20.1.1.1	8	0.0.0.1	0.0.0.0	0.0.0.0	1460

	Enabled	Name	Status	IP Type	Address	Mask	Increment	Count	Gateway	Gateway Increment	Gateway Increment Mode	MSS
1	<input checked="" type="checkbox"/>	IP-R1	Unconfigured	IPv4	20.1.1.1	8	0.0.0.1	8000	0.0.0.0	0.0.0.0	Increment every subnet	1460

Figure 45. Configuration example for Network 1

Configuring the Network Parameters for Network2

Note: This network will host 8,000 H.323 IP Phones; every simulated phone will use a unique IPv4 address for both H.323 and RTP traffic.

1. Click **Network2** to display **IP Network Ranges**.

Test Case: Determining the Maximum Number of Concurrent Calls

2. Set the following parameters:
 - a. **IP Type** = IPv4
 - b. **# Hosts** = 8,000
 - c. **First IP/Subnet** = 20.2.1.1
 - d. **Mask** = 8
 - e. **Increment** = 0.0.0.1
 - f. **Gateway** = 0.0.0.0
 - g. **Gateway Step Size** = *None*

Table 15. Summary of Network 2 parameters

IP Type	Count	Address	Mask	Increment	Gateway	Gateway Increment	MSS (RX)
IPv4	8,000	20.2.1.1	8	0.0.0.1	0.0.0.0	0.0.0.0	1460

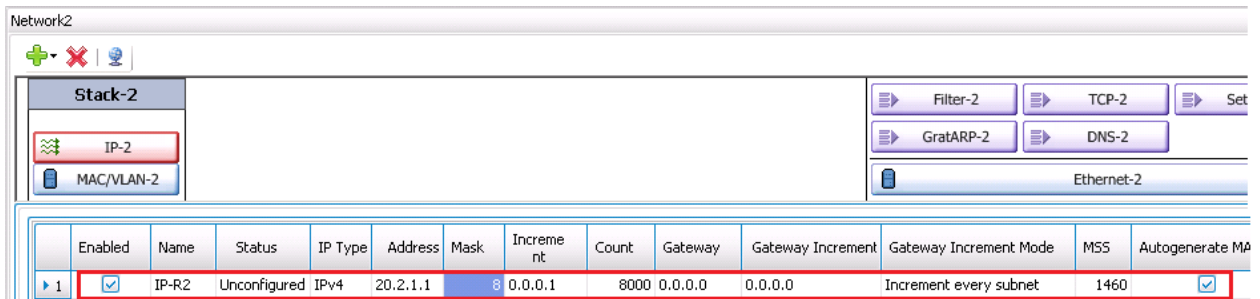


Figure 46. Network2 configuration example

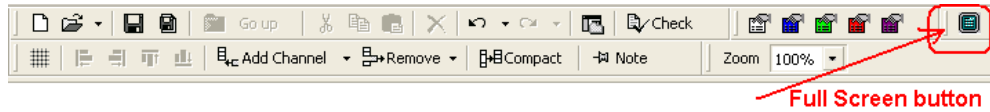
Configuring the Talk Time of 5 Minutes (300 sec)

The number of active calls depends directly on the call hold time (talk time), which is configurable within the test scenario or within the Activity settings (the **Audio** tab). We recommend you to use the settings at the activity level; in this way, the call duration can be changed from the automation tools because the parameters of the activity are exposed through TCL API (the parameters in **Scenario Editor** are not accessible from TCL API). To make the configuration using the parameters from activity:

1. From **Network1**, click the H.323 peer activity labeled **Make_Call**.
2. The configuration page shows **Scenario Editor**.

Test Case: Determining the Maximum Number of Concurrent Calls

3. Click **Full Screen** located on the **Scenario Editor** toolbar. **Scenario Editor** appears in full screen mode.



4. Locate the **Voice Session** script objects used by **Scenario Editor**. The scenario has two functions (one on the caller side and one on the callee side).

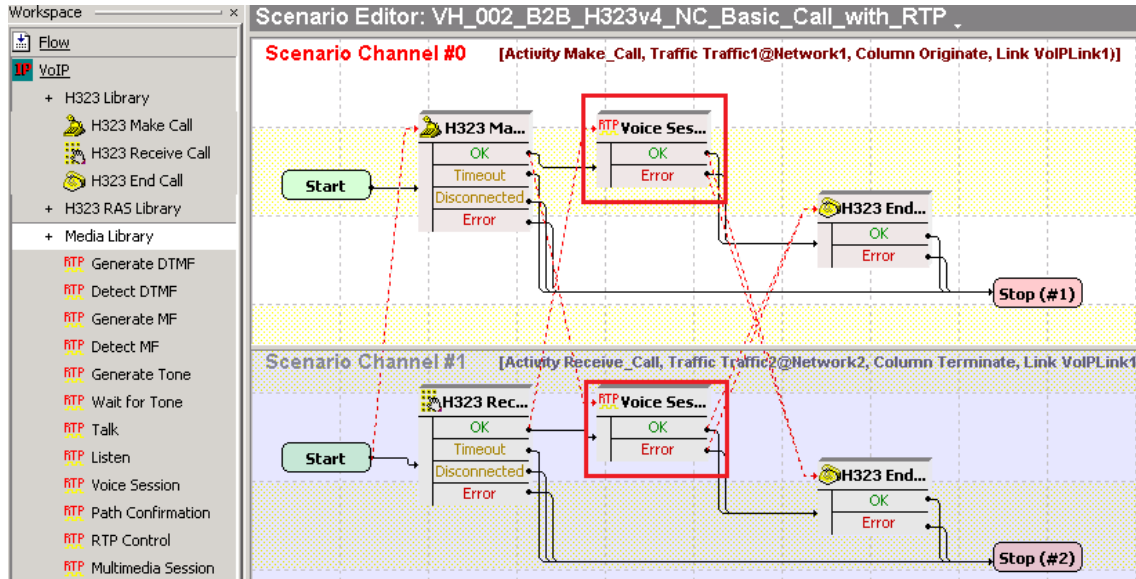


Figure 47. Scenario Editor in full screen mode - see highlighted Talk script objects

Test Case: Determining the Maximum Number of Concurrent Calls

5. For both **Voice Session** script objects included in the script, do the following:
 - a. Double-click the **Voice Session** script object. Its properties open; the **Talk Parameters** tab appears.
 - b. Clear the **Overwrite playback activity settings** check box.
 - c. Click the **Listen Parameters** tab.
 - d. Clear the **Overwrite playback activity settings** check box.
 - e. Click **OK** to close the properties page.
 - f. Repeat Steps a) to e) for the second RTP **Talk** script object.

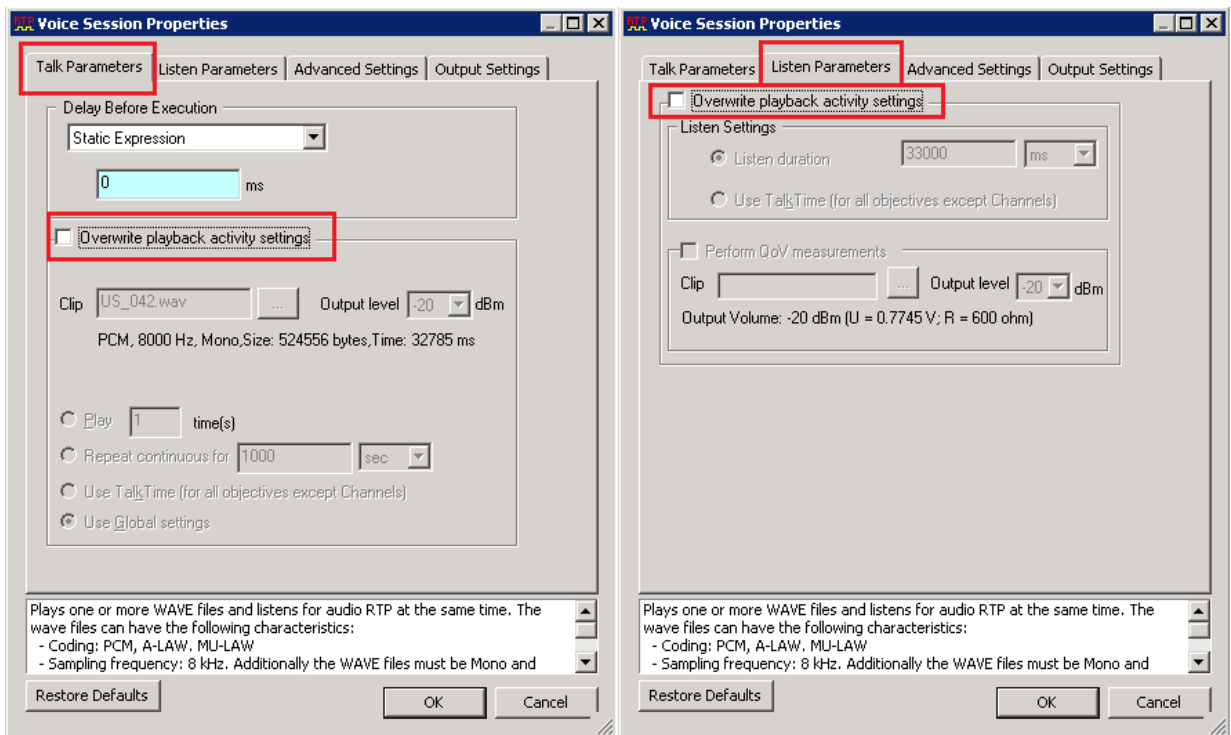
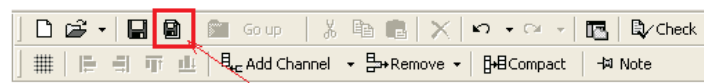


Figure 48. Voice Session parameters

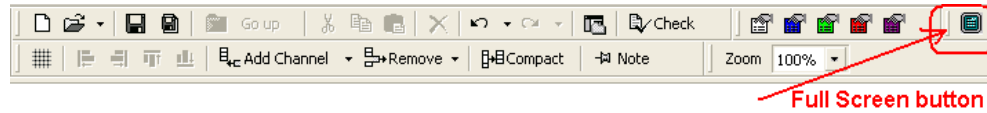
6. Save the test scenario flow by clicking **Save As** located in the **Scenario Editor** toolbar in a convenient location and with an appropriate name (for example, 'H323_MakeCall – ReceiveCall – EndCall with RTP.tst').



Save As button

Test Case: Determining the Maximum Number of Concurrent Calls

7. Click **Full Screen** located in the **Scenario Editor** toolbar. **Scenario Editor** exits full mode.



8. Save the IxLoad configuration file using **File > Save As ...**; save the configuration in a convenient location with a proper name (for example, 'IxLoad Voice – H323 Concurrent Calls.rxf').

Configuring the Execution Settings for the Make_Call Activity

1. Click the **Make_Call** activity.
2. Click the **Execution** page.
3. Set the script to be executed a single time during the Sustain Time by setting **Run for** to **A number of loops** and entering **1**.
4. Select the **Graceful Ramp-down** check box (this is the default setting).

Note: This option forces the users to hang up the call when the ramp-down request is received in the middle of the call.

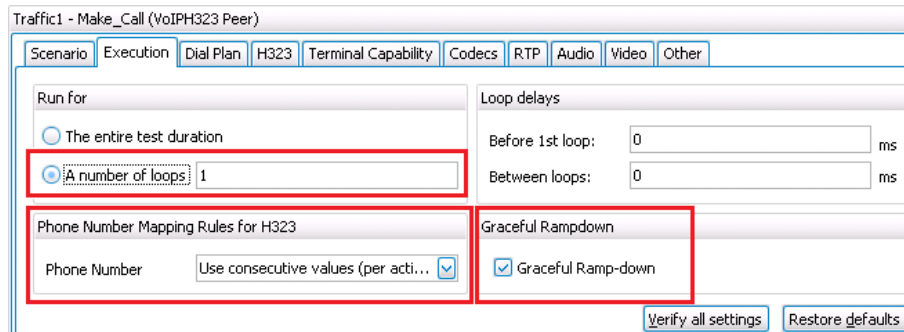


Figure 49. Configuration example for Execution Settings page of VoIPPeer1

5. Set the **Phone Number Mapping Rules for H.323**:
 - a. **Phone Number** = *Use consecutive values (per activity)*.

Configuring the Dial Plan for the Make_Call Activity in Network 1

1. Click the **Dial Plan** configuration page of **Make_Call**.
2. Set the **Source Phone numbers** by selecting the **User defined** check box.
3. Set the source phone number using the sequence *818501[1000-]*.
4. Keep the **Type of Alias**, **Type of number**, and the **Numbering plan** options to their defaults.

Test Case: Determining the Maximum Number of Concurrent Calls

5. Set the destination to **Traffic2_Receive_Call** by clicking the corresponding symbolic link from the list available for destination IPs.

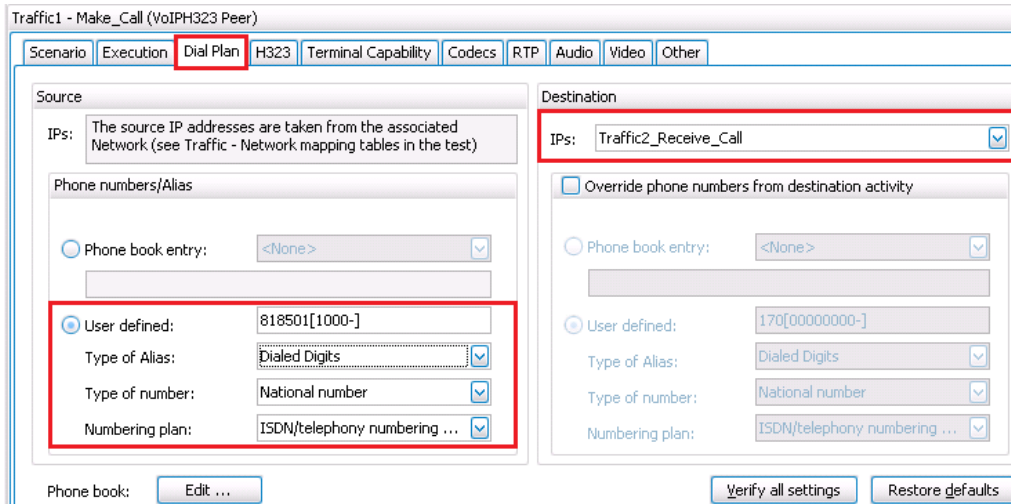


Figure 50. Make_Call, Dial Plan - configuration example for Network 1

Configuring the H.323 Settings for the Make_Call Activity in Network 1

1. Click the **H.323** configuration page of **Make_Call**.
2. Verify that the **Enable signaling on this activity** check box is selected.
3. Verify that the **Enable FastStart** check box is cleared
4. Verify that the **Enable Tunneling** check box is cleared.
5. Set **Q.931 User-User** to *818501[1000-]*.
6. Set **Q.931 Display** to *Ixia-818501[1000-]*.

Test Case: Determining the Maximum Number of Concurrent Calls

7. Verify that the **Enable RAS** check box is cleared.

Traffic1 - Make_Call (VoIPH323 Peer)

Scenario Execution Dial Plan **H323** Terminal Capability Codecs RTP Audio Video Other

Enable signaling on this activity (if unchecked, all H323 script functions will be SKIPPED)

H323 Specific Settings

Advanced Signalling Options

Enable FastStart

Enable Tunneling

Enable Parallel H245

Send Call Alerting

Send Call Proceeding

Bandwidth And Terminal Type

Bandwidth (in Kbps) 64

Terminal Type Terminal Entity without MC (50)

Versions

H.225 Version 0.0.8.2250.0.5

H.245 Version 0.0.8.245.0.9

Registration, Admission and Status (RAS)

Enable RAS

Use Registration Parameters

Auto Register to GateKeeper

Use Gatekeeper for Admission

Enable Disengage

Enable Keep-Alive Registration

Retry and Timeouts For RAS Messages

Maximum Retry Count 1

Timeout (in secs) 4

GateKeeper

Adresse(s) for GK Discovery 198.18.80.80

Q.931

User-User 818501[1000-] Hexadecimal Byte Stream

Display Ixia-818501[1000-]

Figure 51. Configuration example for H.323 settings of Make_Call activity (H.323 Normal Start)

Configuring the CODEC Settings Page for the VoIPH.323Peer1 Activity

The following figure highlights the default settings for this page. The **Codec** page controls the CODECs supported by the simulated phone, the preferential order of the CODEC, and additional CODEC options such as packet time.

Traffic1 - Make_Call (VoIPH323 Peer)

Scenario Execution Dial Plan H323 Terminal Capability **Codecs** RTP Audio Video Other

Audio Codecs:

Name
1 G711 μ-Law
2 G711 A-Law

Settings

Packet time: 20 ms (160 bytes per frame)

ITU-T G.711 is a standard to represent 8 bit compressed pulse code modulation (PCM) samples for signals of voice frequencies, sampled at the rate of 8000 samples/second. G.711 encoder creates a 64 Kbps bitstream.

Payload type

Incoming payload type: 0

Outgoing payload type: 0

WARNING: The outgoing payload type number will be used for RTP header payload type when sending packets and viceversa. Please note that the outgoing payload type of a sender must be equal to the incoming payload type of a receiver.

Figure 52. Configuration example for the Codec Settings of the Make_Call

The configuration shown uses the default settings that negotiate **G.711 uLaw** with a **Packet time of 20 ms (160 bytes per frame)**.

Test Case: Determining the Maximum Number of Concurrent Calls

Configuring the RTP Settings for the Make_Call Activity

1. Click the **RTP** tab.
2. Select the **Enable advanced stats calculation** check box.
3. Select the **Per Stream Statistics** check box.
4. Select the **Enable HW Acceleration** check box to allow 8,000 RTP streams per port.

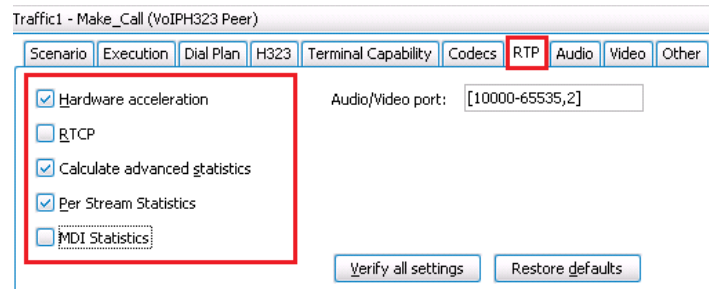


Figure 53. Configuration example for the RTP Settings page

Configuring the Audio Settings for the Make_Call Activity

1. Click the **Audio** tab.
2. Verify that the **Enable audio on this activity** check box is selected.
3. Set the duration of the **Play for** to *5 minutes*
4. Select the **Perform MOS** check box to also automatically select the **Calculate One Way Delay** check box.

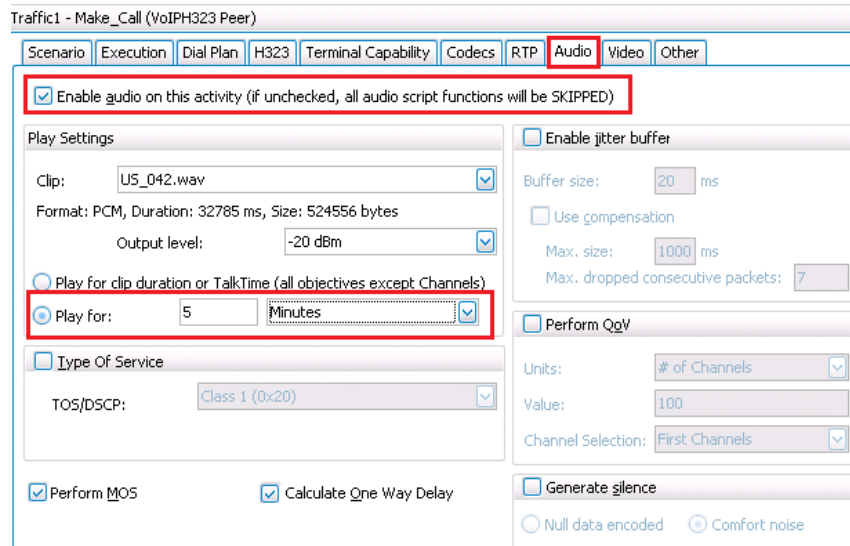


Figure 54. H323 Make_Call - Configuration example for the Audio Settings page

Configuring the Execution Settings for the Receive_Call Activity

1. Click the **Receive_Call** activity.
2. Click the **Execution** tab.
3. Verify that the script will be executed a single time during the Sustain Time; the **Run for to A number of loops** is set to **1**.
4. Verify that the **Graceful Ramp-down** check box is selected.

Note: This option forces the users to hang up the call when the ramp-down request is received in the middle of the call.

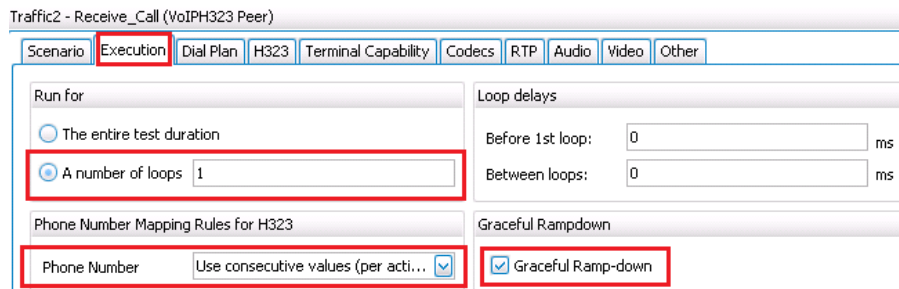


Figure 55. Configuration Example for Execution Settings page of VoIPPeer2

5. Under **Phone Number - Mapping Rules for H.323**, set **Phone Number** to **Use consecutive values (per port)**.

Configuring the Dial Plan for the Receive_Call Activity in Network2

1. Click the **Dial Plan** configuration page of **Receive_Call**.
2. Set the **Source Phone** numbers by selecting the **User defined** check box, and enter the sequence *919501[1000-]*.

Test Case: Determining the Maximum Number of Concurrent Calls

3. We do not recommend you to set the destination while this activity just receives calls and does not generate calls.

Traffic2 - Receive_Call (VoIPH323 Peer)

Scenario Execution **Dial Plan** H323 Terminal Capability Codecs RTP Audio Video Other

Source

IPs: The source IP addresses are taken from the associated Network (see Traffic - Network mapping tables in the test)

Phone numbers/Alias

Phone book entry: <None>

User defined: 919501[1000-]

Type of Alias: Dialed Digits

Type of number: National number

Numbering plan: ISDN/telephony numbering ...

Phone book: Edit ...

Destination

IPs: None

Override phone numbers from destination activity

Phone book entry: <None>

User defined: 170[00000000-]

Type of Alias: Dialed Digits

Type of number: National number

Numbering plan: ISDN/telephony numbering ...

Verify all settings Restore defaults

Figure 56. Receive_Call, Dial Plan - configuration example for Network 2

Configuring the H.323 Settings for the Receive_Call Activity

1. Click the **H.323** configuration page of **Receive_Call**.
2. Verify that the **Enable signaling on this activity** check box is selected.
3. Verify that the **Enable FastStart** check box is cleared.
4. Verify that the **Enable Tunneling** check box is cleared.
5. Set **Q.931 User-User** to *918501[1000-]*.
6. Set **Q.931 Display** to *Ixia-918501[1000-]*.

Test Case: Determining the Maximum Number of Concurrent Calls

7. Verify that the **Enable RAS** check box is cleared.

The screenshot shows the configuration page for 'Receive_Call (VoIPH323 Peer)'. The 'H323' tab is selected. A red box highlights the 'Enable signaling on this activity (if unchecked, all H323 script functions will be SKIPPED)' checkbox, which is checked. Another red box highlights the 'Enable FastStart' and 'Enable Tunneling' checkboxes, both of which are unchecked. A third red box highlights the 'Enable RAS' checkbox, which is also unchecked. Below this, the 'Registration, Admission and Status (RAS)' section contains several other unchecked checkboxes: 'Use Registration Parameters', 'Auto Register to GateKeeper', 'Use Gatekeeper for Admission', 'Enable Disengage', and 'Enable Keep-Alive Registration'. The 'Retry and Timeouts For RAS Messages' section shows 'Maximum Retry Count' set to 1 and 'Timeout (in secs)' set to 4. The 'GateKeeper' section shows 'Adresse(s) for GK Discovery' set to 198.18.80.80. The 'Q.931' section shows 'User-User' set to 918501[1000-] and 'Display' set to Ixia-918501[1000-]. A 'Hexadecimal Byte Stream' checkbox is also present and unchecked.

Figure 57. Configuration example for H.323 settings of Receive_Call activity (H.323 Normal Start)

Configuring the CODEC Settings Page for the Receive_Call Activity

The second group of H.323 phones is configured to use **G.711 uLaw**, **Packet time = 20 ms (160 bytes per frame)**.

The screenshot shows the 'Codec Settings' page for 'Receive_Call (VoIPH323 Peer)'. The 'Codecs' tab is selected. A red box highlights the 'Audio Codecs' table, which lists two codecs: 'G711 μ-Law' (index 1) and 'G711 A-Law' (index 2). Another red box highlights the 'Packet time' dropdown menu, which is set to '20 ms (160 bytes per frame)'. The 'Settings' section contains a text area with the following text: 'ITU-T G.711 is a standard to represent 8 bit compressed pulse code modulation (PCM) samples for signals of voice frequencies, sampled at the rate of 8000 samples/second. G.711 encoder creates a 64 Kbps bitstream.' The 'Payload type' section shows 'Incoming payload type' and 'Outgoing payload type' both set to 0. A warning message is displayed: 'WARNING: The outgoing payload type number will be used for RTP header payload type when sending packets and viceversa. Please note that the outgoing payload type of a sender must be equal to the incoming payload type of a receiver.'

Figure 58. H323 Receive_Call - Codec Settings page

Test Case: Determining the Maximum Number of Concurrent Calls

Configuring the RTP Settings for the Receive_Call Activity

1. Click the **RTP** tab.
2. Select the **Enable advanced stats calculation** check box.
3. Select the **Per Stream Statistics** check box.
4. Select the **Enable HW Acceleration** check box to allow 8,000 RTP streams per port.

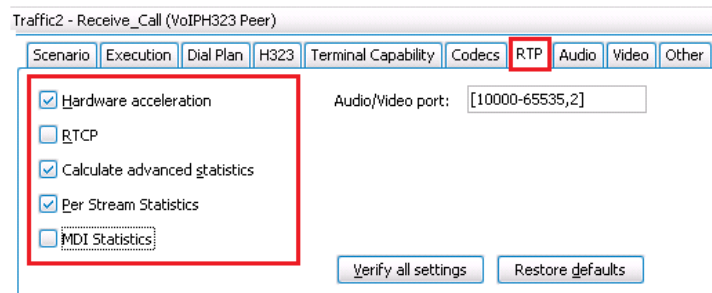


Figure 59. Configuration example for the RTP Settings page

Configuring the Audio Settings for the Receive_Call Activity

1. Click the **Audio** tab.
2. Verify that the **Enable audio on this activity** check box is selected.
3. Set the duration of the **Play for** to *5 minutes*
4. Select the **Perform MOS** check box to also automatically select the **Calculate One Way Delay** check box.

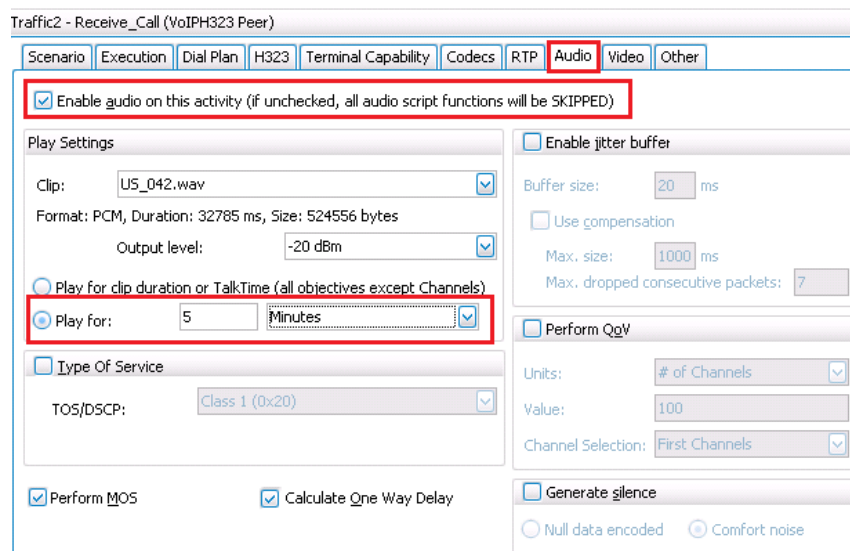


Figure 60. H323 Receive_Call - Configuration example for the Audio Settings page

Test Case: Determining the Maximum Number of Concurrent Calls

Configuring the Timeline & Objective

1. Click **Timeline & Objective** from the **Test Configuration** panel.
2. Set the test **Objective Type** to **Channels**.
3. Set the test **Objective Value** to **8,000**.
4. Under **Timeline**, set the **Ramp Up Value** to **100**.
5. Set the **Ramp Up Interval** to **1 second**.
6. Set the **Sustain Time** to **6 minutes**.

Note: The **Sustain Time** must be higher than the **Call Hold Time**. It is also good practice to add some extra time (for example, 30 seconds) as a buffer for any delays that may occur during the *call setup* and *end call* phases. Hence, ramp-up should have a value higher than $300 \text{ sec} + 30 \text{ sec} = 330 \text{ sec} = 5.30 \text{ min}$.

7. Set the **Ramp Down Time** to **30 seconds**.

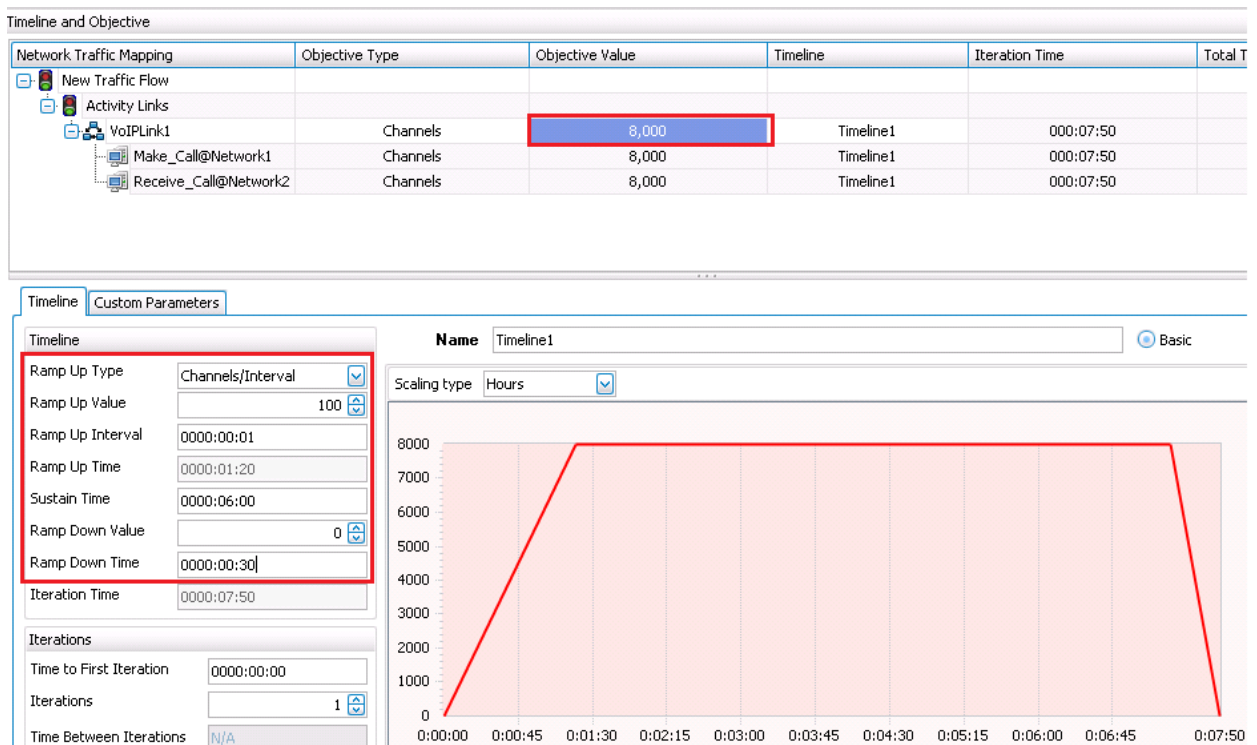


Figure 61. Configuration example for the H323 concurrent calls test objective

Test Case: Determining the Maximum Number of Concurrent Calls

Configuring Test Options

1. From the Test Configuration panel, click **Test Options**.
2. Select the **Forcefully Take Ownership** check box.
3. Select the **Reboot Ports Before Configuring** check box.
4. Set **CSV Polling Interval** to 1 second.

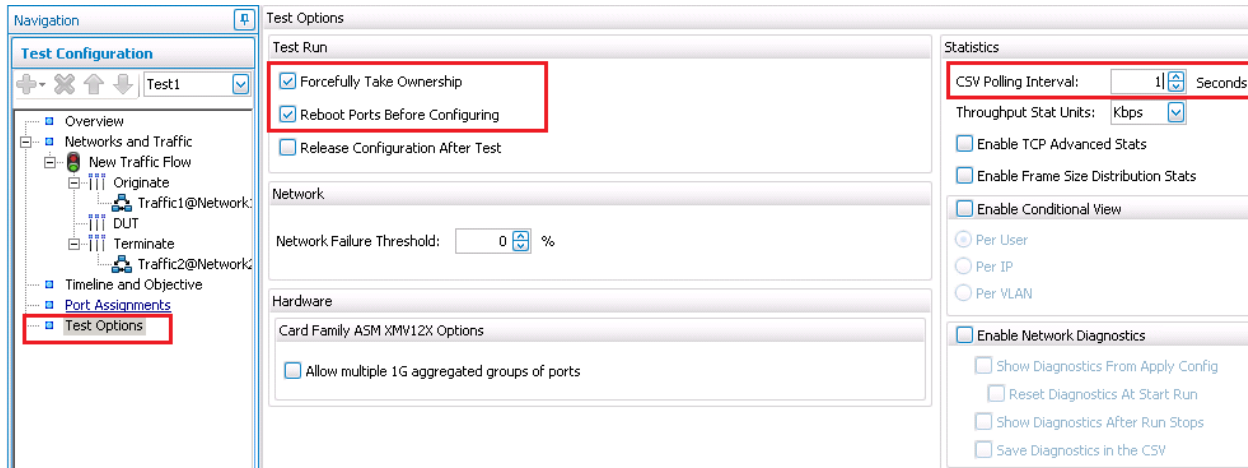


Figure 62. Configuring Test Options

Configuring Port Assignments

1. From the Test Configuration panel, click **Port Assignments**. Add your assigned chassis: 10.200.128.34.
2. Identify your assigned cards and ports using the **Annex** provided.
3. Add your first assigned port to the **VoIP H.323Peer1** activity.

Test Case: Determining the Maximum Number of Concurrent Calls

4. Add your second assigned port to the **VoIP H.323Peer2** activity.

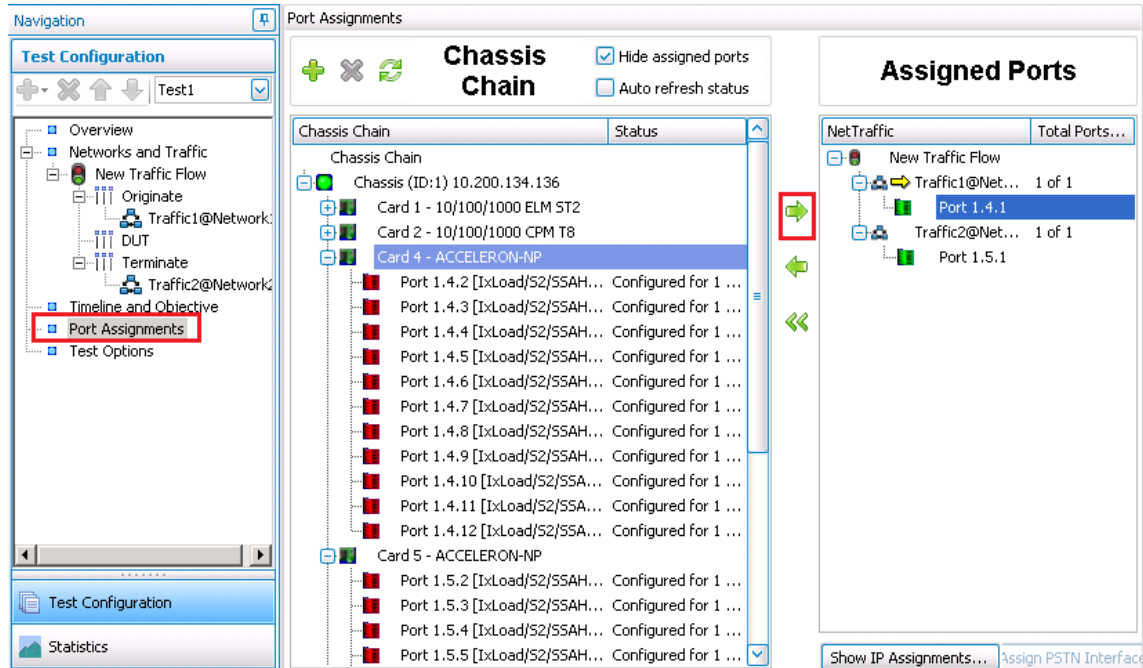


Figure 63. - Assigning test ports

Running the Test

1. Click **Run** to start test execution.
2. IxLoad will automatically display **Statistic Views** after the execution starts.

Configuration Highlights for H.323 Fast Start Mode

H.323 version 2 introduces a new method of call setup called fast start or fast connect. In this mode, an H.323 endpoint will bypass some steps to make call setup faster. In addition to the speed improvement, fast start allows the media channels to be operational before the CONNECT message is sent, which is a requirement for certain billing procedures.

To change the configuration to use H.323 fast start mode, repeat the following steps for both H.323 activities:

1. Click the **H.323** configuration page of **Make_Call**.
2. Verify that the **Enable signaling on this activity** check box is selected.

Test Case: Determining the Maximum Number of Concurrent Calls

3. Select the **Enable FastStart** check box.

Traffic1 - Make_Call (VoIPH323 Peer)

Scenario Execution Dial Plan **H323** Terminal Capability Codecs RTP Audio Video Other

Enable signalling on this activity (if unchecked, all H323 script functions will be SKIPPED)

H323 Specific Settings

Advanced Signalling Options

Enable FastStart

Enable Tunneling

Enable Parallel H245

Send Call Alerting

Send Call Proceeding

Bandwidth And Terminal Type

Bandwidth (in Kbps) 64

Terminal Type Terminal Entity without MC (50)

Versions

H.225 Version 0.0.8.2250.0.5

H.245 Version 0.0.8.245.0.9

Figure 64. H.323 Configuration example for FAST START

Configuration Highlights for H.245 in Parallel with Fast Start

Another method used to speed up the negotiation of audio/video parameters is defined in H.323 version 4. The method uses binary embedding, in which the calling endpoint creates its H.245 request **terminalCapabilitySet** at the very beginning of the call and embeds this message in the H.225.0/Q.931 message setup. In this way, the called party knows the caller's entire capability list right from the beginning.

H.323v4 allows H.245 to start in parallel with fast connect by including H.245 messages in the Setup message. By starting H.245 early, two endpoints can establish an H.245 session faster in the event that the called endpoint rejects fast connect.

To configure the H.323 activities to use H.245 in parallel with fast start repeat the following steps for both activities:

1. Click the **H.323** configuration page of **Make_Call**.
2. Verify that the **Enable signaling on this activity** check box is selected.
3. Select the **Enable FastStart** check box.

Test Case: Determining the Maximum Number of Concurrent Calls

4. Select the **Enable Parallel H.245** check box.

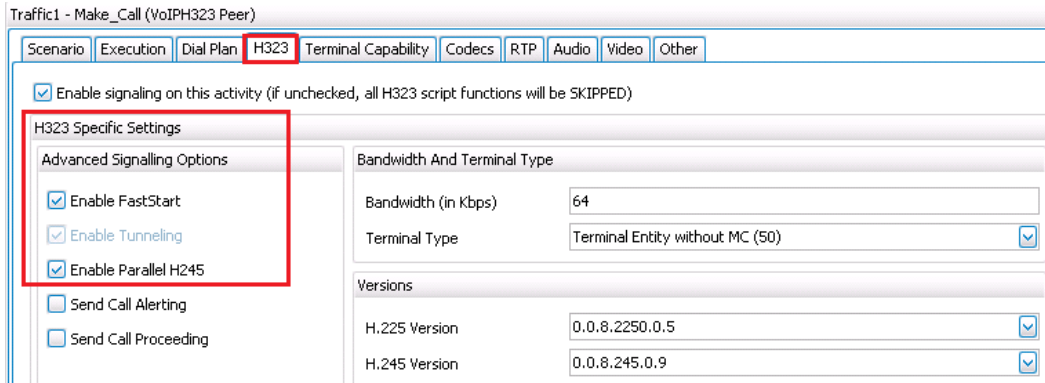


Figure 65. Configuration example for H.245 in parallel with FAST START

Results Analysis

The following questions are provided to help discover performance issues more readily:

1. Has the test objective been achieved? Use the **Throughput Outbound/Inbound** statistics and **Concurrent RTP Streams** statistics available in the **RTP Streams** view.

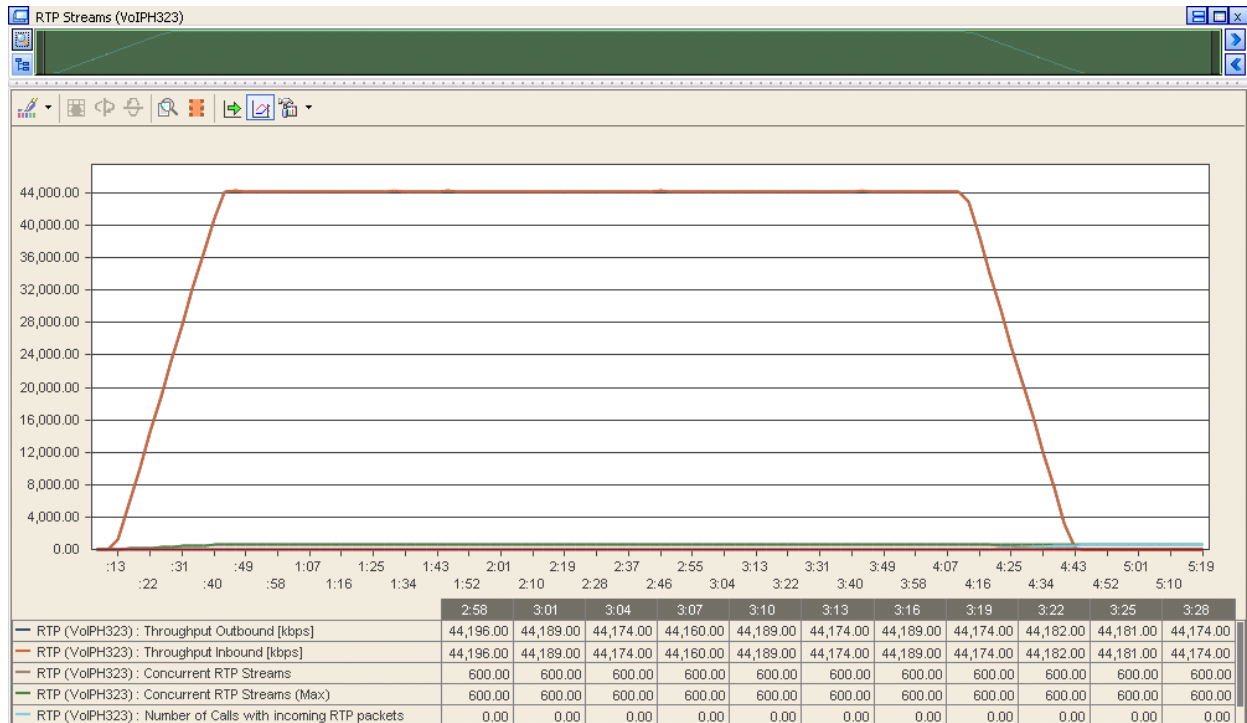


Figure 66. – The expected characteristic of throughput

Test Case: Determining the Maximum Number of Concurrent Calls

2. What is the call rate?

Table 16. Call rate statistics

Statistic Name	Value	Questions
Calls Attempted per Second		1. Have the calls been attempted continuously at a constant call rate during the Sustain Time? 2. How do the <i>calls attempted rate</i> and the <i>calls connected rate</i> compare with each other?
Calls Connected per Second		

3. Have any call failures been reported? Check the **Calls** view.

Table 17. Call Statistics

Statistic Name	Value	Questions
Calls Attempted		1. Has any call attempt failed? Compare: c. Calls Attempted and Calls Received. d. Calls Attempted and Calls Connected.
Calls Connected		
Calls Received		
Calls Answered		
End Calls Initiated		1. Has any attempt to hang up the call failed? Compare: c. <i>End Calls Initiated</i> and <i>End Calls Received</i> . d. <i>Calls Attempted</i> and <i>End Calls Received</i> .
End Calls Received		
End Calls Completed		2. Have all the calls attempted ended successfully? Compare: [2 * <i>Calls Connected</i>] and [<i>End Calls Completed</i>]. 3. Why do we need to compare <i>End Calls Completed</i> with twice the number of <i>calls connected</i> ?

4. Have any scenario loop failures been reported? Check the **Loops** statistics view.

Table 18. Statistics highlighting the pass/fail result based on the call flow execution

Statistic Name	Value	Questions
Total Loops		1. Are the <i>Successful Loops</i> and <i>Total Loops</i> equal? 2. Have any <i>failed loops</i> , <i>aborted loops</i> or <i>warning loops</i> been reported? Note: failed/aborted and warning loops highlight failures at the scenario level.
Successful Loops		
Failed Loops		
Aborted Loops		
Warning Loops		

Test Case: Determining the Maximum Number of Concurrent Calls

5. Has the QoS for signaling met the expected quality? Check the **Call Times** and **Delays** statistic views. Use the maximum value reported.

Table 19. Statistics used to determine the QoS for the SIP signaling

Statistic Name	Value (max/avg/min)	Questions
Call Setup Time		1. Is the maximum Call Setup Time less than 4 seconds?
End Call Time		2. Is the maximum End Call Time less than 2 seconds?
Talk Time		
Media Delay TX/RX		3. Is the maximum Media Delay (Tx or Rx) less than 4 seconds?
Post Dial Delay		4. Is the maximum Post Dial Delay less than 2 seconds?
Post Pickup Delay		5. Is the maximum Post Pickup Delay less than 2 seconds? 6. For every stat listed in this table, compare its value distribution in time.

6. Has the QoS for media met the expected quality? Check the **RTP MOS RTP QoS, RTP Advanced QoS, RTP Jitter Distribution, RTP Consecutive Lost Datagram Distribution, and RTP Streams** statistic views.

Table 20. MOS Statistics

Statistic Name	Value(s)	Questions
RTP MOS Best RTP MOS Worst	Max = _____ Min = _____	1. How do the last values reported by the <i>RTP MOS Best</i> and <i>RTP MOS Worst</i> compare with each other? 2. How does the <i>RTP MOS Worst</i> score compare with the max theoretical score for the CODEC used?
RTP MOS Instant (Best/Avg/Worst)	Max = _____ Min = _____ Avg = _____	1. Are there any times without an instantaneous MOS value? 2. How frequent are the changes in the instantaneous MOS values?
RTP MOS Per Call (Best/Avg/Worst)	Max = _____ Min = _____ Avg = _____	1. How are the MOS per Call statistics compared with the RTP MOS Best and RTP MOS Worst statistics?

Test Case: Determining the Maximum Number of Concurrent Calls

Table 21. Basic RTP QoS Statistics (see RTP QoS and RTP Advanced QoS statistics views)

Statistic Name	Value(s)	Questions
RTP Packets Sent RTP Packets Received RTP Packets Lost		<ol style="list-style-type: none"> Are there any differences between RTP Packets Sent and RTP Packets Received? Do the differences match the value of RTP Lost Packets?
RTP One Way Delay [us]		<ol style="list-style-type: none"> Is the One Way Delay higher than 100 ms?
RTP Delay Variation Jitter [us] RTP Interarrival Jitter [us]		<ol style="list-style-type: none"> What is the max Delay Variation Jitter? What is the max Interarrival Jitter?

Table 22. RTP Jitter Distribution statistics

Statistic Name	Value(s)	Questions
Packets with Delay Variation Jitter up to 1 ms		<ol style="list-style-type: none"> Assuming Jitter was reported, what is the distribution of the Delay Variation Jitter values?
Packets with Delay Variation Jitter up to 3 ms		
Packets with Delay Variation Jitter up to 5 ms		
Packets with Delay Variation Jitter up to 10 ms		
Packets with Delay Variation Jitter up to 20 ms		
Packets with Delay Variation Jitter up to 40 ms		
Packets with Delay Variation Jitter over 40 ms		

Test Case: Determining the Maximum Number of Concurrent Calls

Table 23. : Distribution of RTP Consecutive Lost Packets

Statistic Name	Value(s)	Questions
Consecutive Loss of One Packet Sequence		1. Assuming that packet loss was reported, what is the distribution of the lost RTP packets?
Consecutive Loss of Two or Three Packet Sequences		
Consecutive Loss of Four or Five Packet Sequences		
Consecutive Loss of Six to Ten Packet Sequences		
Consecutive Loss of Eleven or More Packet Sequence		

Table 24. RTP Streams

Statistic Name	Value(s)	Questions
Concurrent RTP Streams		1. What is the max number of concurrent RTP Streams reported by IxLoad? 2. For what duration does the Concurrent RTP Streams indicate that all the channels were active?
Concurrent RTP Streams (max)		
Number of calls with incoming RTP packets		3. Are any calls without RTP? 4. Are any calls with RTP? Is this number equal to 2 x Calls Connected?
Number of calls without incoming RTP packets		

Troubleshooting and Diagnostics

The following table summarizes some of the common issues that may be encountered running a capacity test

Table 25.

Issue	Troubleshooting Solution
The <i>Incoming RTP throughput</i> is not constant during the Sustain Time or has lower values than the <i>Outgoing RTP Throughput</i>	Check the reported <i>RTP Packets Lost</i> , <i>RTP Consecutive Packets Lost</i> , and compare the RTP Packets Sent with RTP Packets Received. This issue may be caused by a large number of packets being dropped or by calls without RTP packets. Endpoints connected in calls without receiving any media are counted using the <i>Calls without RTP packets</i> statistic.
All calls were connected but the expected maximum throughput is not reached	Verify that requests to disconnect the calls are not received from the DUT during the <i>Talk Time</i> .
The <i>incoming throughput</i> and <i>outgoing throughput</i> values are lower or higher than expected	Verify the <i>Codec Distribution</i> and <i>RTP Packet Distribution</i> statistics – these may be used to confirm that the calls negotiated the desired audio/video CODEC. Problems may be fixed by correcting the DUT configuration or by setting the same CODEC on both calling and receiving activities.

Test Variables

Test Tool Variables

Table 26.

Parameter Name	Current Value	Additional Options
<i>IP version</i>	IPv4	IPv6
<i>Number of channels</i>	8,000	Up to 96,000 per card (assumes ASM1000XMV12-RTP cards)
<i>Call Duration</i>	300 sec	User defined
<i>Network Configuration</i>	Static IPs	IP/DHCP, IP/IPsec, IP/PPPoE
Codec parameters for the negotiated audio CODEC (type, packet size & frequency)	G.711u, 160 bpf every 20ms	G.711, G.729, G.723, G.726, iLBC, AMR
<i>RTCP</i>	On	Off
H.323 Call Setup Mode	Slow Start	Fast Start, Tunneling
IP Mapping rules for signaling and media	N to 1	1 to 1, 1 to N, N to 1 and N to N
Mix with data protocols (for example, FTP, HTTP, Telnet)	Not included	Any combination of data protocols supported by IxLoad
Secure RTP	Off	On
TOS/DSCP marking for H.323 & RTP	Off	On (individual settings for H.323 & RTP)

DUT Test Variables

Table 27.

Parameter Name	Current Value	Additional Options
IP version	IPv4	IPv6
Transport Protocol	SIP/UDP	SIP/TCP, SIP/TLS
Dial Plan size	-	-

Conclusions

This example demonstrated how to configure IxLoad to maintain 8,000 calls active for 220 seconds. This configuration allows you to measure the maximum number of simultaneously active calls. Guidance on how to assess the QoS for media and signaling is also included.

Test Case: VoIP Quality of Service in Converged Networks

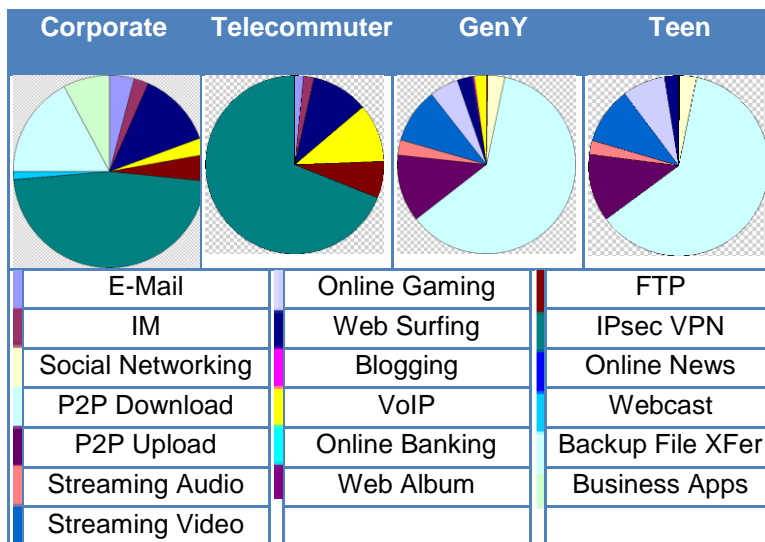
Overview

The migration to packetized voice (VoIP) is driven by the desire to use existing data networks for application traffic, video, and voice. This means that effective VoIP performance testing cannot be done in isolation, because few networks run VoIP exclusively. Services of all types that use a range of protocols are seen in modern multiplay networks, including:

- **Data** – HTTP, HTTPS, FTP, E-mail
- **Voice over IP** – SIP, MGCP, RTP
- **IPTV** – RTSP, IGMP
- **Peer-to-peer** – BitTorrent, eDonkey, Gnutella
- **Infrastructure** – DHCP, DNS, RADIUS
- **Security** – SSL, TLS, IPsec

The table below illustrates typical network usage for different user profiles.

Table 28. Traffic distribution profiles



VoIP is present in all profiles, constituting from 0.5 percent to 15 percent of network utilization. Even though only a small part of the overall throughput is used for VoIP, the quality of experience (QoE) for voice calls is critical. Users expect the same quality for VoIP calls as for land-line service.

Transporting real-time voice over the same network used for all other data traffic presents challenges for service quality. Verifying VoIP performance under conditions of high data stress is important to ensure expected results.

Objective

These tests will determine if the DUT supports the desired level of load without degradation of the QoS for VoIP when subject to background traffic. The QoS is measured by the call completion rate and quality of voice (that is, MOS). The distribution of traffic follows a subscriber model with a traffic throughput distribution of 85 percent video, 22 percent data transfer, and 3 percent VoIP.

Setup

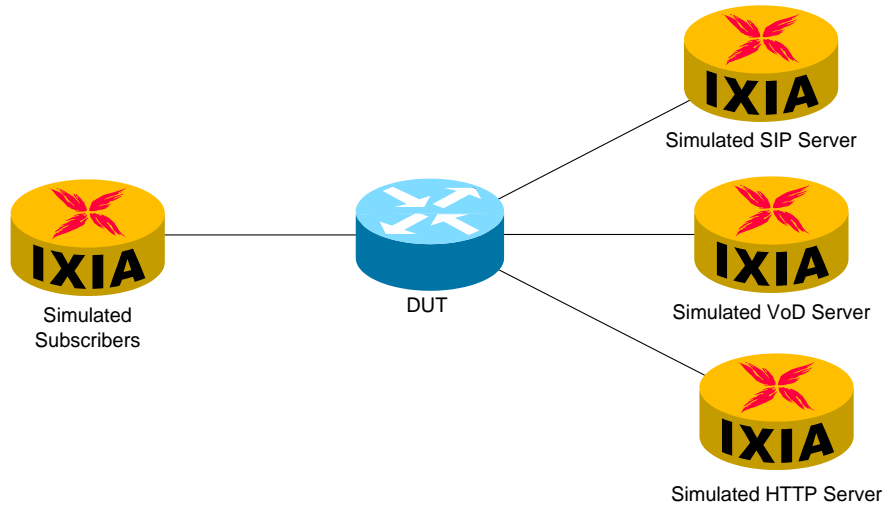


Figure 67. - Typical topology for VoIP QoS measurement in the presence of data and video traffic

For this example, the DUT is a router used as an edge device between home subscribers and the core network. The subscribers access the services provided by the session initiation protocol (SIP), video, and HTTP servers.

Step-by-Step Instructions

The final IxLoad configuration as a result of these steps is provided on blackbook.ixiacom.com Web site - see *IxLoad 5.10 Voice - QoV in Converged Networks.crf*. To import a Compressed Repository File (crf) in IxLoad use the command **Import** under the **File** menu.

The step-by-step instructions highlight how to set the essential parameters of this configuration and explain additional options, which may be used to change the behavior of the test.

1. Configure a test with two traffic flows, one for subscriber simulation and one for servers simulation. Use the network parameters defined in the following tables:

Table 29. Network parameters for the Subscriber simulation

Network Name	HomeSubscribers
Type of "Traffic Flow Element"	Subscriber
IP Type	IPv4
Count	100
Address	201.0.0.11
Mask	24
Increment	0.0.0.1
Gateway	0.0.0.0
Gateway Increment	0.0.0.0
MSS (RX)	1460

Table 30. Network parameters for the Servers simulation

Network Name	Servers
Type of "Traffic Flow Element"	Net Traffic
IP Type	IPv4
Count	1
Address	201.0.0.1
Mask	24
Increment	0.0.0.1
Gateway	0.0.0.0
Gateway Increment	0.0.0.0
MSS (RX)	1460

Test Case: VoIP Quality of Service in Converged Networks

This example assigns a single IP address to all servers (HTTP, IPTV, and SIP). For high-scale tests, it is possible to use multiple application load module ports to simulate servers; in this case, the number of IP addresses should be equal with the number of ports used for servers simulation.

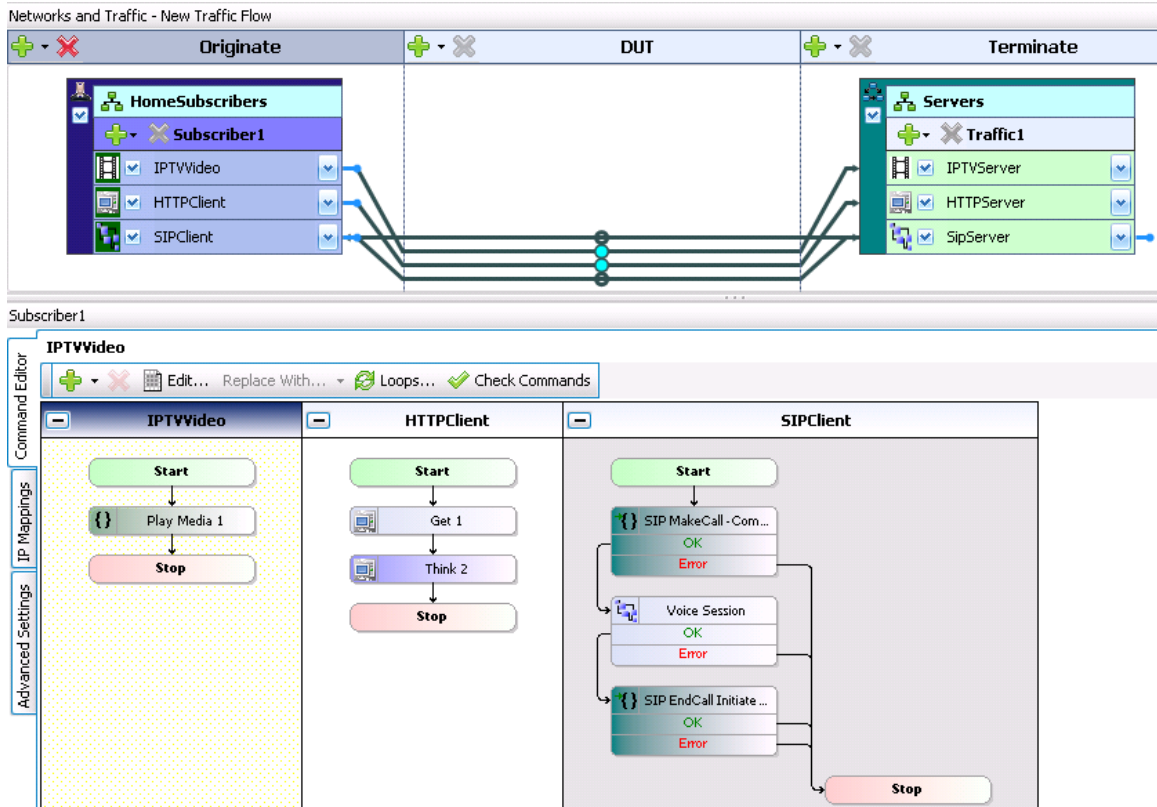


Figure 68. - IxLoad configuration for multiplay testing

2. In the **Servers** network, add an **IPTV/Video Server** activity. In this example, the activity has been renamed to *IPTVServer*. In the **Video Config** page set the **Stream Count** parameter to *40* and **Duration** to *300* seconds. In the **Advanced Options** change the **UDP Port Range** to *[10000-20000]*; the same IP address is used for Video and VoIP traffic, so the UDP port domain is shared between the IPTV and VoIP activities.

Keep the rest of the parameters at their default values.

The test is now configured for unicast video streaming; the voice QoS should be tested also with multicast video traffic.

3. In the **Server** network, add an **HTTP Server** activity. In this example, the activity has been renamed to *HTTPServer*. Leave the default parameters unchanged.

Test Case: VoIP Quality of Service in Converged Networks

- In the **HomeSubscribers** network add an **IPTV/Video Client** activity. In this example, the activity has been renamed to *IPTVClient*. Drag and drop the lollipop of the *IPTVClient* activity to *IPTVServer* activity. Automatically the command **Play** is added in the **Command List** with the parameters shown in the table below.

Table 31. IPTV Client, Play command parameters

Parameter	Value
Server IP Address	Traffic1_IPTVServer:554
Media	Stream0

Leave the rest of the parameters at their default values.

- In the **HomeSubscribers** network, add an **HTTP Client** activity. In this example, the activity is renamed to *HTTPClient*. Drag and drop the lollipop of the *HTTPClient* activity to *HTTPServer* activity. Automatically the command **Get** is added in the **Command List**. Set the Page/Object to */128k.html*. The parameters of the **GET** functions are shown in the table below:

Table 32. HTTP Client, Get Parameters

Parameter	Value
Destination (IP or IP:Port)	Traffic1_HTTPServer:80
Page/Object	/128k.html

Add a **Think** command and set the **Random Duration Between** 1000 and 3000 ms. This will reduce the throughput to around 22Mbps for 40 users (the test objective)

Leave the remainder of the parameters at their default values.

- In the **HomeSubscribers** network, add a **VoIPSIP Peer** activity. In this example, the activity has been renamed to *SIPClient*. Edit the **Scenario** by adding the **SIP Make Call - Complete** procedure, the **Voice Session** function, and the **End Call Initiate** procedure. Save the scenario; in this example, the scenario is save as *SIP_MakeCall – with RTP.tst*

Test Case: VoIP Quality of Service in Converged Networks

TIP: an existing test scenario may be loaded instead of creating it from scratch. For example, you may load *VS_022_DUT_SIPv4 MakeCall - EndCall with RTP - 33s.tst*, provided with the product.

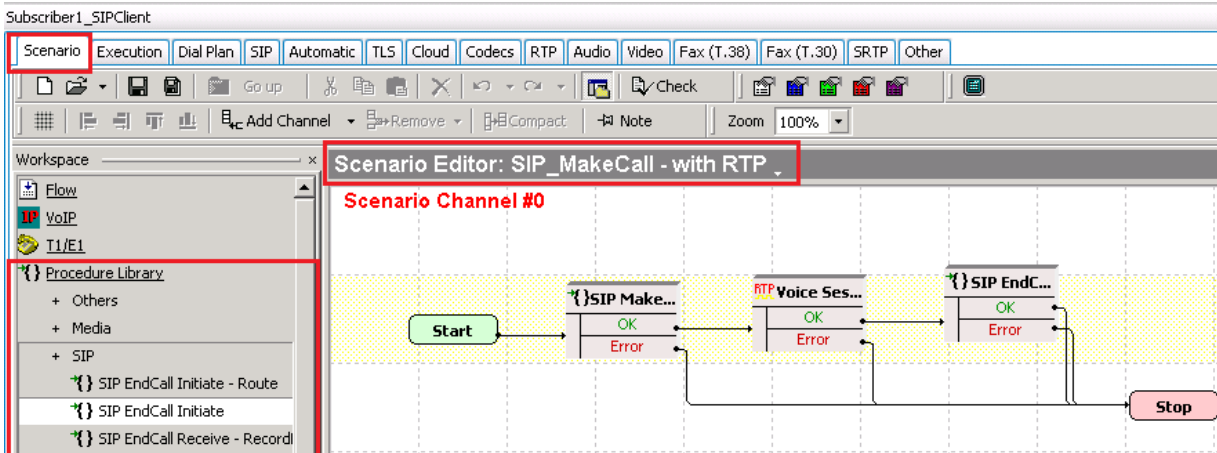


Figure 69. SIP Make Call flow

7. In the **Servers** network, add a **VoIPSIP Peer** activity. In this example, the activity has been renamed to *SIPServer*. Edit the **Scenario** by adding the **SIP Receive Call** procedure, the **Voice Session** function, and the **End Call Receive** procedure. Save the scenario; in this example, the scenario is save as *SIP_ReceiveCall - with RTP.tst*

TIP: an existing test scenario may be loaded instead of creating it from scratch. For example, you may load *VS_027_DUT_SIPv4 ReceiveCall - EndCall with RTP - 33s.tst*, provided with the product.

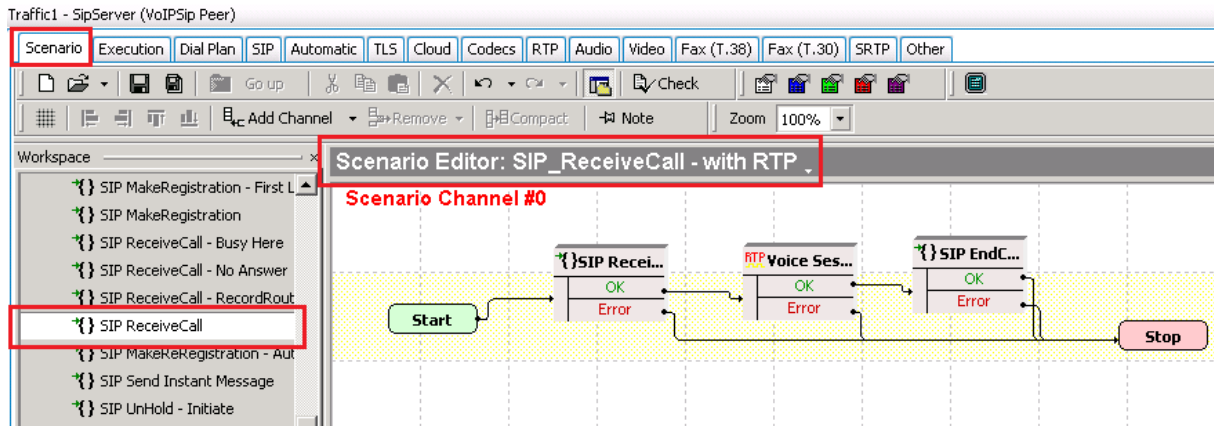


Figure 70. SIP Receive Call test scenario

Test Case: VoIP Quality of Service in Converged Networks

8. Set the following parameters for the **SIPClient** activity (part of the **HomeSubscribers** network):

Table 2 – Configuration parameters of the SIP simulated UAs (VoIPSIP activity)

Configuration Page	Parameter	Value
Dial Plan	Destination IP	<i>Subscriber1_SIPClient:[5060-]</i> - by clicking from the list
Automatic	Enable retransmissions	Enable
Audio	Enable audio	Enable
	Enable Jitter buffer	Enable
	Perform MOS	Enable

Leave the remainder of the parameters at their default values.

9. Set the following parameters for the **SIPServer** activity (part of the **Servers** network):

Table 33. Configuration parameters of the SIP simulated trunk (“VoIPSIPServer” activity)

Configuration Page	Parameter	Value
Execution	Channel mapping rules for SIP UA / IP Address	Use same value (per port)
	Channel mapping rules for SIP UA / UDP/TCP/TLS port	Use same value
	Channel mapping rules for SIP UA / Phone no	Use consecutive values (per activity)
	Channel mapping rules for RTP / IP Address	Use same value (per port)
	Channel mapping rules for RTP/ UDP port	Use consecutive values (per port)
RTP	Audio/Video port	<i>[30000-65535,4]</i> The same IP address is used for Video and VoIP traffic, so the UDP port range is shared between the IPTV and VoIP activities.
Audio	Enable audio	Enable
	Enable Jitter buffer	Enable
	Perform MOS	Enable

Test Case: VoIP Quality of Service in Converged Networks

10. Set the test objective and timeline for each activity.

Table 34. Test objective

Activity	Objective Type	Objective Value
Subscriber1@HomeSubscribers	Subscribers	40
SIPServer	Channels	40

For all activities, use the same **Timeline (Timeline 1)** with **Ramp Up** value *10* and **Sustain Time** *10:00* minutes.

The objective values for each activity are set to meet the proposed throughput distribution between the different protocols. With these values, the total inbound throughput is 175 Mbps.

11. Assign the ports to generate traffic – at least four ports are required. The test may be scaled up by increasing the number of ports and increasing the test objectives proportionally.
12. Start the test and activate the following views:
- **Video Client – Data Rates**
 - **HTTP Client – Throughput Objective**
 - **Calls (VoIPSIP)**
 - **RTP MOS (VoIPSIP)**

Test Variables

Depending on the characteristics of the DUT the test should be repeated with following variations.

Table 35. - Test tool variables

Parameter Name	Current Value	Additional Options
IP Type	Static IPv4	Static IPv6, IPv4 or IPv6 /DHCP
Data Traffic	http	In addition to HTTP traffic, you may add other types of activity (ftp, smtp, pop3) to better simulate the type of traffic generated/consumed by subscribers and passing the DUT.
Video Traffic	VoD	Video traffic in this example is unicast video on demand. Multicast IPTV can be added or replace the VoD traffic.
Audio Codec	G.711 uLaw	Lower bit-rate codecs (G.729, G.723) request lower bandwidth; DUT may scale better for lower bit-rate codecs.
Audio Codec Packet Time	20ms	5 ms, 10 ms, 30 ms –RTP overhead is significant for small packets and the number of RTP packets increases with a reduction in packet time. At 20 ms, if there are 10,000 packets/sec for 100 full duplex calls, the number of packets becomes 40,000/sec at 5 ms packet time. IP Ethernet throughput is doubled by changing the packet time from 20 ms to 5 ms.
TOS/DSCP for Signaling or RTP traffic	Best Effort	Class 1, 2, 3, 4 Express Forwarding, Control, or Custom according to the settings on the DUT.
Total Inbound Throughput	300 Mbps	Increase the volume of traffic to the maximum level supported by the DUT.

Results Analysis

Analysis is focused on VoIP measurements. Beside the quality of voice metrics, the most important statistics are video and data traffic throughput. Check the **Video Client – Data Rates** view and compare the obtained values with the expected ones. Check the **HTTP Client – Throughput Objective** values for this traffic.

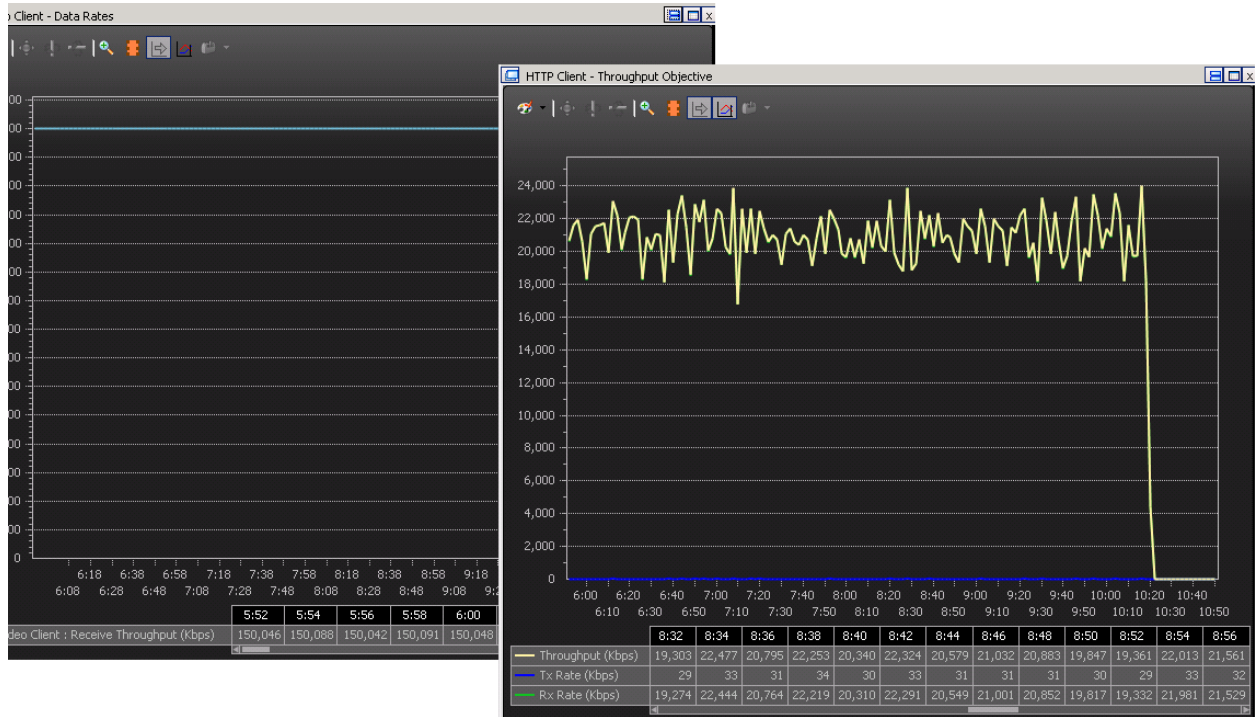


Figure 71. Video Client – Data Rates and HTTP Client – Throughput Objective

Test Case: VoIP Quality of Service in Converged Networks

To verify the QoS for VoIP, the call completion statistics for signaling and MOS values for media are the most important.



Figure 73. Calls (VoIPSIP) and RTP MOS (VoIPSIP)

In this example, there are 40 simulated user agents that originate SIP calls. There are 760 **Attempted Calls** – each simulated user agent attempted 19 calls during the test duration. The number of attempts depends on the call duration (the **Talk Time** in the **Voice Session** function). The number of **Connected Calls** is equal to the number of **Attempted Calls** indicating that there are no problems with the call setup and the DUT can handle this volume of calls in the presence of video and data traffic. The number of **End Calls Completed** is 1,520 (double the number of attempted / connected calls because both call legs are successfully ended).

Other statistics that should be checked in addition to the number of calls completed and the RTP MOS:

- Number of retransmissions (open the **SIP Messages** view to see the **Retransmitted Msg** statistics) – no retransmission should occur.
- **VoIP SIP Errors** – no errors should occur.
- **RTP Streams** – the number of calls with incoming RTP should equal the number of completed calls. The number of calls without incoming RTP should be zero.
- **Lost Packets** – should be close to zero (less than 0.1 percent of the total RTP packets).
- **One Way Delay (Max)** – should be less than 100 usec. This should be checked against DUT specifications.

Troubleshooting and Diagnostics

Issue	Diagnosis, Suggestions
The target throughput is not reached for HTTP and IPTV traffic. Throughput goes up and down.	If the device is not reaching steady state, check the TCP failures. High TCP timeout and RST packets can indicate that the device is unable to handle the load.
Not all SIP attempted calls are connected	If the device incurs UDP packet loss, or the routing delay of packets is high, some SIP calls are not completed. Change the transport protocol used for SIP from UDP to TCP, or keep the transport set to UDP and enable retransmissions for the SIP activities.
MOS score is less than 4.41	The expected MOS score when using the G.711 uLaw codec is 4.41. If the score is lower, be sure this is not the effect of the codec; if the test is configured to use a different codec, the score will be lower. Verify the One Way Delay statistics; a value bigger than 150 ms affects the MOS score. Verify packet loss; a packet loss of 3 percent reduces the MOS score by more than 0.5.
SIP calls are not established at all	Verify that the DUT is routing SIP messages properly. If the DUT has SIP intelligence, it may include a proxy server that requires registration of the user agents before executing the tests. If this is the case, modify the test scenario to include a register procedure before originating calls.

Test Case: Subjective Quality of Voice

Overview

Speech quality in a telephony system is a subjective judgment that depends on technical attributes of the system and the participants' speaking and listening preferences. The ITU-T P.800 specification defines methods for subjective determination of transmission quality. These methods utilize a large number of human subjects who listen to sentences read aloud by professional male and female speakers and transmitted over the telephony system. The listeners rate the quality of the audio signal. Individual results are averaged into a MOS that provides a numerical indication of the quality of transmission, in the range of 1 to 5:

Table 36. Mean Opinion Score

MOS	Quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

PESQ (perceptual evaluation of speech quality) is a mechanism that measures the quality of speech in an automated way defined by ITU-T standard P.862. PESQ is an objective measurement method that predicts the results of subjective listening tests. The PESQ algorithm produces results analogous with the subjective MOS standard. A mapping between PESQ results and MOS was defined after the release of the P.862 recommendation. PESQ-LQ (PESQ- Listening Quality) is defined in ITU-T Rec. P.862.1, and improves the correlation with subjective test results at the high and low ends of the scale. IxLoad measures both values: PESQ-LQ and PESQ-LE (Listening Effort).

PESQ is a full-reference method designed for end-to-end quality of voice assessment using a psycho-acoustic and cognitive model. PESQ analyses the degraded audio signal (the signal after passing through the communication system) versus the reference (the signal injected in the system). The method requires access to actual audio information in both reference and degraded signals, performs level and time alignment to accommodate attenuations and delays, process the disturbances, and finally applies the cognitive model. This is done using signal processing algorithms requiring considerable processing power.

In packet networks quality of voice measurements can be performed by assessing the packets transmission using the E-Model and then generating the metric R-Factor. As defined by ITU-T G.107, R-Factor combines a number of values measuring the effect of various impairments; some of these are:

- The effect of coding/decoding – defined as constants for every codec

Test Case: Subjective Quality of Voice

- Jitter, packet loss, and delay
- The effect of audio signal capture (mouth to microphone) and reproduction (speaker to ear), defined as a constant.

The E-Model doesn't require reference signal information as it does not look at the actual audio content of the degraded signal. This method requires far less processing power than PESQ.

The R-Factor method predicts a user satisfaction on a scale from 0 to 100, where 100 is the highest satisfaction. A formula is defined for conversions between R-Factor and MOS. For example, a perfect transmission with the codec G.711 has an R-Factor of 94 and a MOS of 4.4.

Table 37. R-Factor

R-Factor	User Satisfied
91-100	Very satisfied
81-90	Satisfied
71-80	Some users dissatisfied
61-70	Many users dissatisfied
51-60	Nearly all users dissatisfied
0-50	Not recommended

R-Factor and PESQ both characterize a voice transmission system, the former considers the packet networks impairments, while the latter compares the received audio signal with the expected signal. Beside the effect of the impairments of the transmission network PESQ also captures the effects of trans-coding, voice activity detector, echo cancelation, and any other type of audio signal alteration.

Because the PESQ algorithm is computationally intensive, it is not practical to use it for testing the speech quality on high scale devices or systems. However, if E-Model is used exclusively some issues may remain hidden if the system performs audio signal processing. The best practice is to combine the two methods and perform E-Model measurement on all calls and PESQ on a smaller percentage of them.

Objective

The test that is described here will determine if the DUT provides the desired level of load without quality of voice degradation. If degradation is observed, the cause will be determined.

Setup

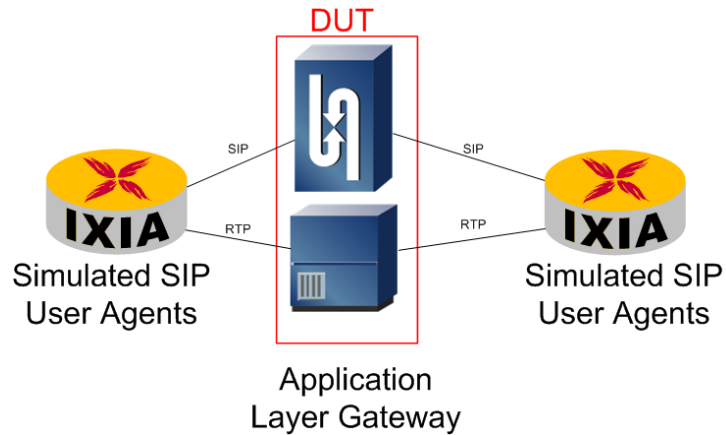


Figure 74. Typical topology for QoV measurement in an end-to-end voice communication system

The DUT used in this example is a distributed application layer gateway used as a media border element performing trans coding. G.711 codec is used in the private network and G.729 in the public network.

Step-by-Step Instructions

The final IxLoad configuration as a result of these steps is provided on blackbook.ixiacom.com Web site – see *IxLoad 5.10 Voice - PESQ.crf*. To import a Compressed Repository File (crf) in IxLoad, use the command **Import** under the **File** menu.

The step-by-step instructions highlight how to set the essential parameters of this configuration and explain additional options, which may be used to change the behavior of the test.

1. Configure a test that uses IxLoad Voice endpoint pairs to simulate SIP calls with media.

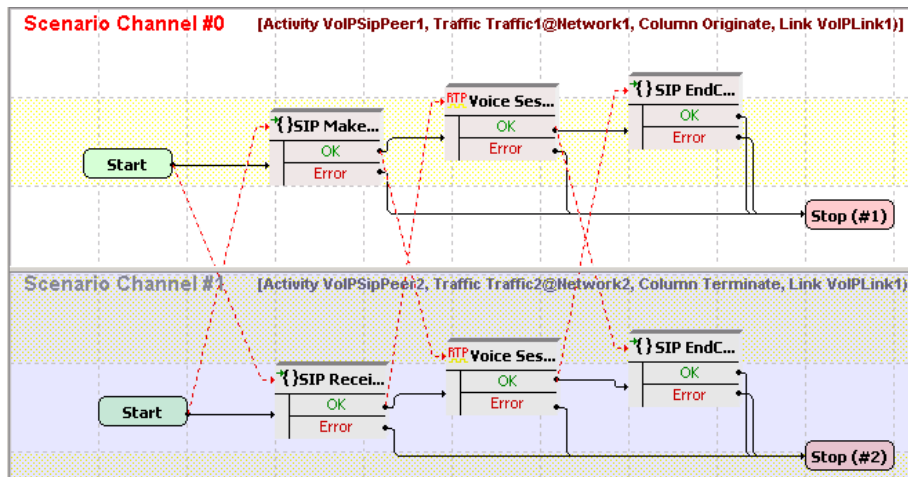


Figure 75. Test scenario to establish SIP calls with media

Use the instructions for [Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems](#) listed in this booklet or refer to the IxLoad user manual for network configuration, test scenario editing, timeline establishment, objective setting, and port assignment. The test is configured with two networks, each with 1,500 distinct static IP addresses. The SIP (VoIPPeer) protocol is used on both networks and the test objective is 1,500 channels (number of simultaneously active calls).

2. Set the parameters of **Voice Session Properties** for quality of voice PESQ analysis. PESQ computes the quality of voice scores by comparing the degraded clip (the audio received) with the reference. The clip used as the reference on the receiving side should match the clip played on the other side.
 - a. In the **Talk Parameters** tab, select the **Overwrite playback activity settings** check box.
 - b. In the **Talk Parameters** tab, set the clip in **Playback Settings** to a voice clip from the **Speech Clips** pool.

Test Case: Subjective Quality of Voice

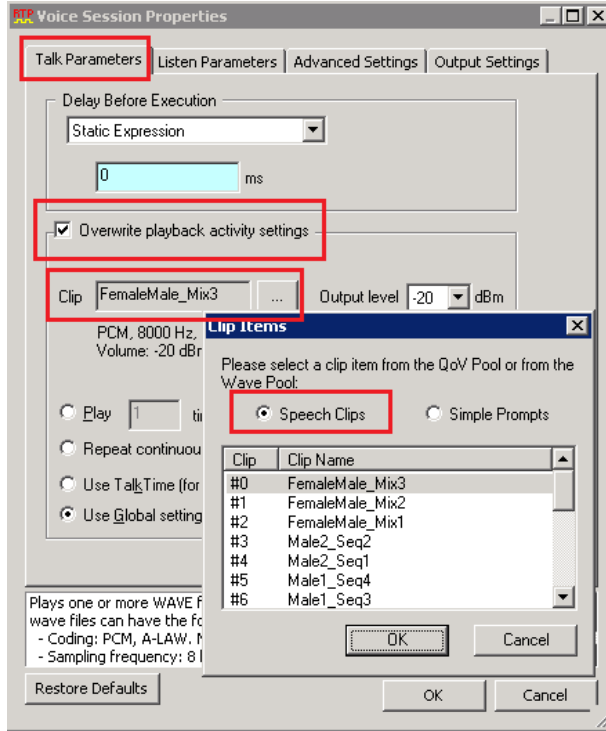


Figure 76. Setting the voice clip to be played

- c. In the **Talk Parameters** tab, select the **Overwrite playback activity settings** check box.

Test Case: Subjective Quality of Voice

- d. In the **Listen Parameter** tab, select the **Perform QoV measurements** check box and select the same clip as in the previous step. Leave the **Listen** duration at its default value.

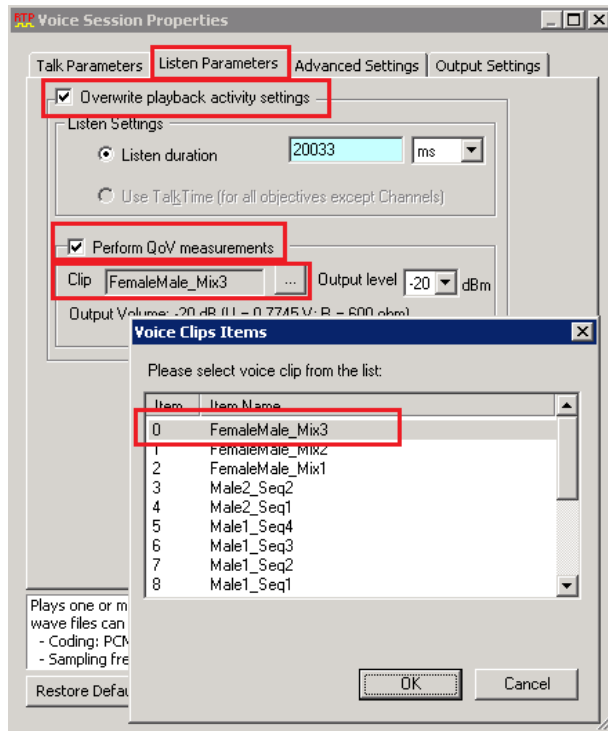


Figure 77. Set the reference for PESQ analysis on the listener side

Test Case: Subjective Quality of Voice

- At the activity level, on the **Audio** settings tab, select the **Perform QoV** check box. Set the parameters in accordance with the volume of traffic desired and available resources. One Ixia VQM load module can perform real-time PESQ analysis on 300 audio streams. The test described here uses a single VQM load module, and because RTP traffic is full duplex and the quality of voice is evaluated at both ends the number of channels with PES.

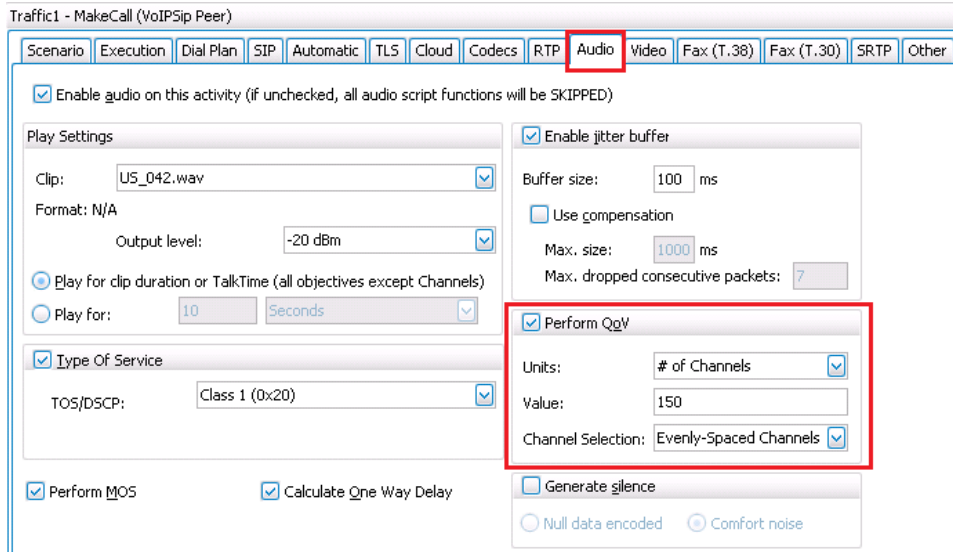


Figure 78. Audio settings

- Analysis is set to 150. In this example, the test objective is set to 1500 channels with the **Evenly Spaced Channels** option selected, which will result in analysis of every tenth channel (1500 divided by 150). Similar settings should be applied to both activities (call origination and termination). The number of channels that can be analyzed depends on the resources available.

Test Case: Subjective Quality of Voice

If insufficient resources are available to analyze all the calls, the subset should be selected based on the expected behavior of the DUT:

- Evenly or random distributed, if no errors are expected on the DUT or if the errors occur randomly.
- In blocks, if errors are expected on a specific subset of calls (for example, the calls on a trunk, or on a specific VLAN).

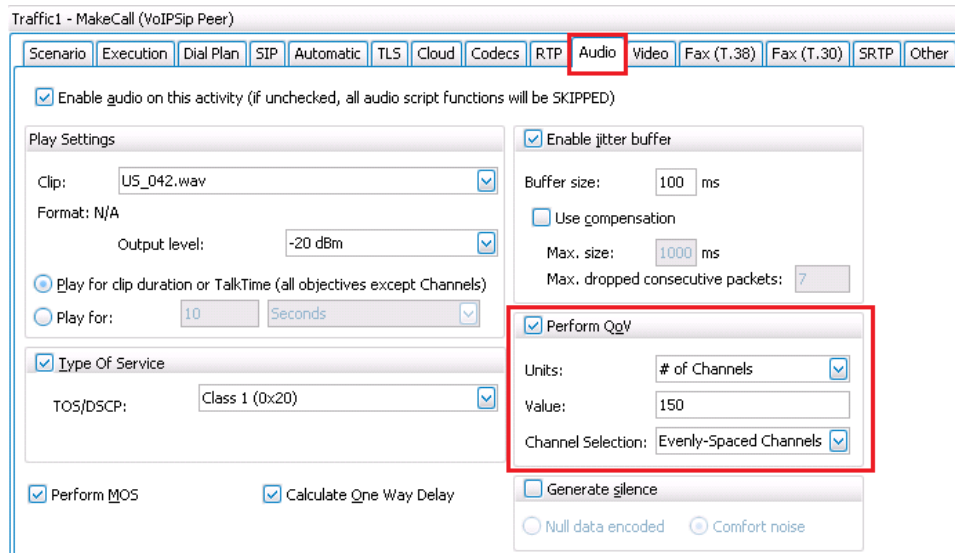


Figure 79. RTP parameters at activity level for PESQ test

5. Select the **Perform MOS** check box. This setting specifies that QoV MOS will be computed for all calls. The MOS scores, in conjunction with PESQ scores, will allow identification of the cause of any quality of voice degradation.
6. Select the **Enable jitter buffer** check box. Keep the default value for Jitter Buffer Size if the expected jitter in the network under test is known to be less than 20 ms or set it to the appropriate value.
7. Set the test objective and timeline in accordance with the DUT's capacity. The number of concurrent calls and the CPS rate may affect the QoV through the DUT.

Test Case: Subjective Quality of Voice

- Assign the ports that will generate traffic, and assign the **Quality of voice Resource Module**. If PESQ is enabled for at least one activity, at least one **Quality of voice Resource Module** must be assigned. The resources of this module are shared by all activities that have PESQ enabled.

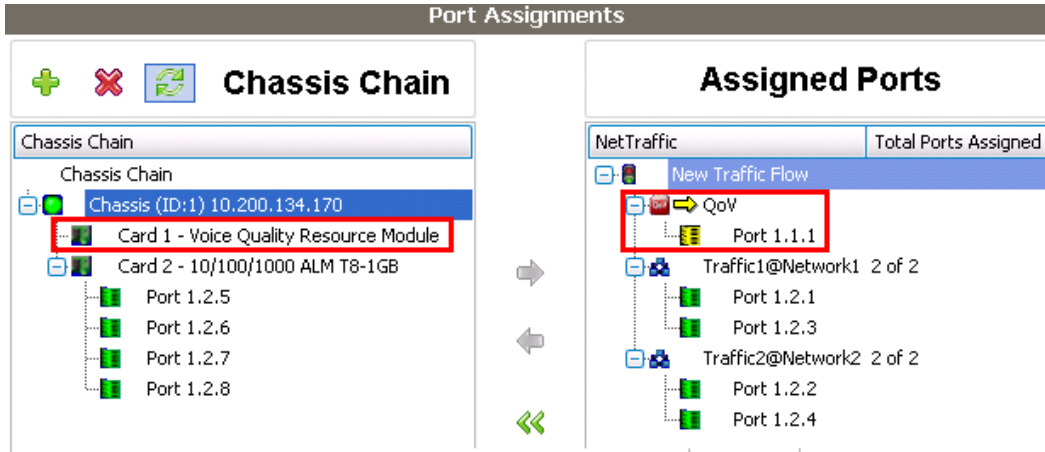


Figure 80. QoV resource assignment for a PESQ test

- Start the test and activate the following statistic views: **RTP MOS**, **QoV PESQ**, **RTP QoS**, and **RTP Per Channel**. Each of these views is provided on a per-protocol basis. For example, if two VoIP signaling protocols are used in same test (for example, SIP and H.323) two **RTP MOS** views are available, one for simulated SIP User Agents and one for H.323 simulated endpoints. In this test, only one signaling protocol is used (SIP); so only one **RTP MOS** view will be shown.

Test Variables

Depending on the characteristics of the DUT, the test should be repeated with the following variations:

Test Case: Subjective Quality of Voice

Test Tool Variables

Table 38. - Test tool variables – PESQ test

Parameter Name	Current Value	Additional Options
Codec Ptime	20ms	10, 30 ms
Call duration	20 sec	3 min, 60 min To extend the call duration, multiple Voice Session functions should be placed in the test scenario. TIP: a loop controlled by the Variable Test and Variable Set functions can be used.
Speech material – vary the language and speaker's gender	Mix Female / Male – English	English Female, English Male, Chinese Female, Chinese Male.
Active Level	-20dBm	-25dBm, -30dBm, -35dBm Variation of the Active level parameter may influence the Quality of Voice if the DUT implements AGC (automatic gain control) or VAD (voice activity detection).
RTP traffic direction	Full duplex	Use simplex RTP traffic by replacing the Voice Session function on one channel with the Talk function and the Voice Session function on the other channel with Listen function. The parameters of these functions should be changed to play a speech clip, and to perform QoV measurement, respectively. Use half-duplex RTP traffic by replacing the Voice Session functions on both channels with a sequence of Talk/Listen and Listen/Talk functions, respectively. The parameters of these functions should be changed to play a speech clip (for the Talk functions), and to perform QoV measurement (for the Listen functions), respectively.

DUT Test Variables

Table 39. DUT test variables – PESQ test

Parameter Name	Current Value	Additional Options
Codec on public network	G.729A	G.729B, G.723, G.726

Results Analysis

The results analysis is done by comparing PESQ scores with expected values. If they are different and if the measured PESQ score is 5 percent or more less than the expected score, then the source of degradation will be determined by analyzing other statistics provided by IxLoad. These include MOS, packet loss, jitter, and delay. Shown below are the results of three test runs: the first is without any degradation, the second with packet loss, and the third with transcoding errors.

IxLoad provides a set of voice clips that have been selected to cover all the characteristics of the human speech. PESQ scores vary with the content and level of the voice clip, and more importantly with the codec used. A table of scores for various clips obtained by connecting load module ports to each other in a back-to-back fashion, with volume levels and codec is provided in the IxLoad documentation. If the DUT performs ideally, then the scores for PESQ should match the values listed in this table.

In the current example, the test is configured to use the voice clip **FemaleMale_Mix3** with an **Active** level of -20dBm. The DUT transcodes the clip from G.711uLaw to G.729A. The expected score is **3.512** as marked in the excerpt from the PESQ LE scores table shown in Table 4, below:

Table 40. Baseline of PESQ LE scores

Active Level	Codec					iLBC 13.33 kbps
	G711 Mulaw	G723 5.3kbps	G729A	G729B		
Female1_Seq1	-20dBm	4.423	3.476	3.751	3.65	3.453
	-25dBm	4.441	3.493	3.739	3.578	3.289
	-30dBm	4.424	3.427	3.716	3.740	3.490
	-35dBm	4.399	3.437	3.683	3.611	3.601
FemaleMale_Mix3	-20dBm	4.457	3.383	3.512	3.512	3.663
	-25dBm	4.436	3.391	3.585	3.635	3.668
	-30dBm	4.440	3.398	3.611	3.623	3.682
	-35dBm	4.397	3.375	3.622	3.600	3.685
Male1_Seq1	-20dBm	4.448	3.410	3.711	3.682	3.766
	-25dBm	4.477	3.39	3.723	3.746	3.701
	-30dBm	4.475	3.367	3.538	3.591	3.488
	-35dBm	4.449	3.431	3.701	3.599	3.628

Test Case: Subjective Quality of Voice

No degradations

If the DUT performed ideally and did not induce any degradation, the results will show a PESQ LE of 3.512 in the **QoV PESQ** view and a MOS of 4.12 in the **RTP MOS** view, as shown in the following figure.

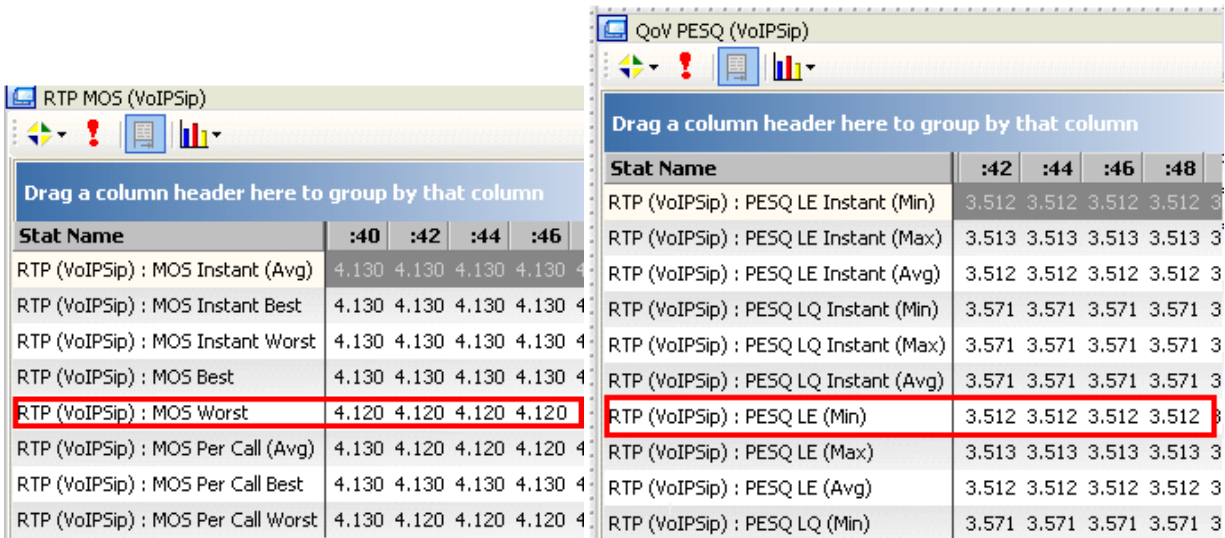


Figure 81. MOS and PESQ values for the ideal conditions

The values for **Lost Packets** are zero, as shown in the following **RTP QoS** view.

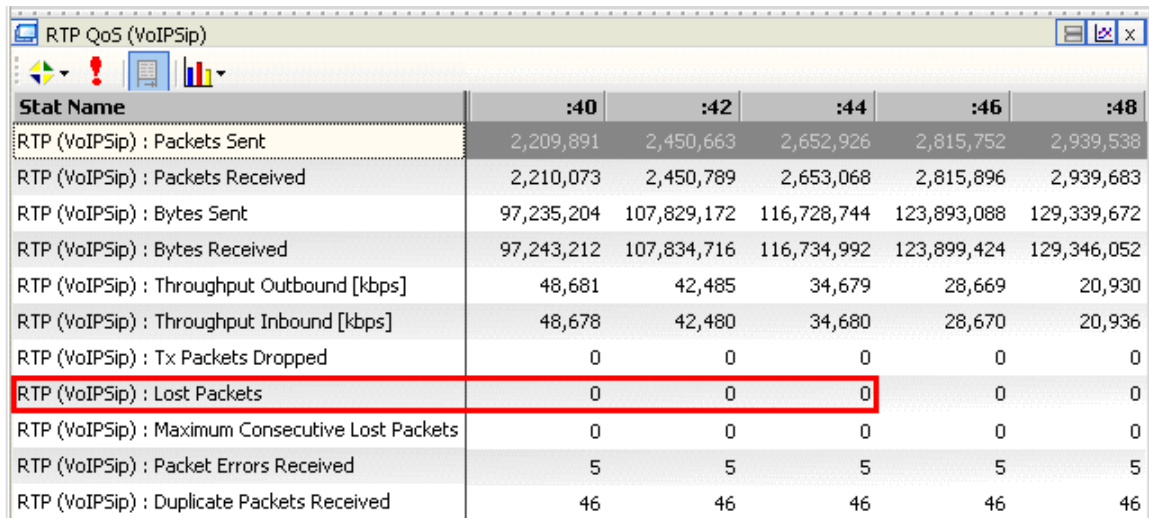


Figure 82. RTP QoS in the case of No degradation

Test Case: Subjective Quality of Voice

Packet loss

If there are impairments at the network level, such as packet loss, delay, and jitter, the quality of voice reported by both metrics (MOS and PESQ) will be affected. The results in this case show a minimum MOS score of 2.95 and a minimum PESQ LE score of 2.858. Under ideal conditions, these values should be 4.12 and 3.512, respectively.

Stat Name	:40	:42	:44
RTP (VoIPsip) : MOS Instant (Avg)	4.100	4.100	4.090
RTP (VoIPsip) : MOS Instant Best	4.130	4.130	4.130
RTP (VoIPsip) : MOS Instant Worst	2.990	2.990	2.990
RTP (VoIPsip) : MOS Best	4.130	4.130	4.130
RTP (VoIPsip) : MOS Worst	2.950	2.950	2.950
RTP (VoIPsip) : MOS Per Call (Avg)	3.960	3.960	3.960
RTP (VoIPsip) : MOS Per Call Best	4.130	4.130	4.130
RTP (VoIPsip) : MOS Per Call Worst	3.480	3.480	3.480

Stat Name	:40	:42	:44
RTP (VoIPsip) : PESQ LE Instant (Min)	2.858	2.858	2.858
RTP (VoIPsip) : PESQ LE Instant (Max)	3.513	3.513	3.513
RTP (VoIPsip) : PESQ LE Instant (Avg)	3.444	3.437	3.447
RTP (VoIPsip) : PESQ LQ Instant (Min)	2.614	2.614	2.614
RTP (VoIPsip) : PESQ LQ Instant (Max)	3.571	3.571	3.571
RTP (VoIPsip) : PESQ LQ Instant (Avg)	3.472	3.461	3.476
RTP (VoIPsip) : PESQ LE (Min)	2.858	2.858	2.858
RTP (VoIPsip) : PESQ LE (Max)	3.513	3.513	3.513
RTP (VoIPsip) : PESQ LE (Avg)	3.455	3.470	3.478
RTP (VoIPsip) : PESQ LQ (Min)	2.614	2.614	2.614

Figure 83. MOS and PESQ scores in case of "Packet loss"

Poor scores for both metrics is an indication that the DUT has induced degradation in the packet flow. To identify the type of degradation, **RTP QoS** values can be used. In the case discussed here, there are lost packet: 4,209 out of 3,079,504. This is 0.1 percent of the total number of packets, a small amount that cannot cause a significant degradation of quality of voice. Packet loss may not affect all the calls – some calls may be unaffected while others can have high level of packet loss.

Test Case: Subjective Quality of Voice

In this case, the Maximum MOS and Maximum PESQ values are equal to the ideal values and the Average MOS and Average PESQ values are close to the maximum values. This is an indication that only some calls are affected by packet loss. This is confirmed by RTP statistics per channel.

Stat Name	:40	:42	:44	:46	:48	:50
RTP (VoIPSip) : Packets Sent	2,766,612	2,885,426	2,974,697	3,034,819	3,073,629	3,079,504
RTP (VoIPSip) : Packets Rec...	2,762,291	2,881,077	2,970,315	3,030,364	3,069,027	3,074,946
RTP (VoIPSip) : Bytes Sent	121,730,928	126,958,744	130,886,668	133,532,036	135,239,676	135,498,176
RTP (VoIPSip) : Bytes Recei...	121,540,804	126,767,388	130,693,860	133,336,016	135,037,188	135,297,624
RTP (VoIPSip) : Throughput ...	35,305	28,969	24,880	18,670	7,211	0
RTP (VoIPSip) : Throughput ...	35,251	28,948	24,825	18,667	7,183	42
RTP (VoIPSip) : Tx Packets ...	0	0	0	0	0	0
RTP (VoIPSip) : Lost Packets	4,209	4,209	4,209	4,209	4,410	4,548
RTP (VoIPSip) : Maximum Co...	4	4	4	4	4	4
RTP (VoIPSip) : Packet Error...	0	0	0	0	0	0

Figure 84. RTP QoS in case of packet loss

The information concerning the calls that are affected is provided in the **RTP Per Channel** view:

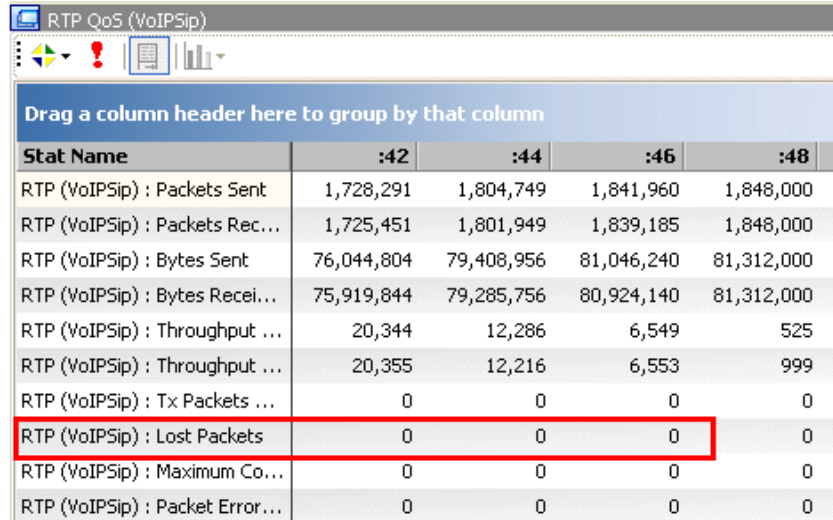
Stat Name	s Se	Bytes Recei ved	Packets Rec eived	Lost Packet
/10.200.134.170/Card02/Port02/VoIPSipPeer2/Channel00024	0	44,616	1,014	0
/10.200.134.170/Card02/Port02/VoIPSipPeer2/Channel00025	0	44,616	1,014	0
/10.200.134.170/Card02/Port02/VoIPSipPeer2/Channel00026	0	44,616	1,014	0
/10.200.134.170/Card02/Port02/VoIPSipPeer2/Channel00027	0	44,616	1,014	0
/10.200.134.170/Card02/Port02/VoIPSipPeer2/Channel00028	0	44,616	1,014	0
/10.200.134.170/Card02/Port02/VoIPSipPeer2/Channel00029	0	44,616	1,014	0
/10.200.134.170/Card02/Port02/VoIPSipPeer2/Channel00030	0	42,900	975	39
/10.200.134.170/Card02/Port02/VoIPSipPeer2/Channel00031	0	43,824	996	18
/10.200.134.170/Card02/Port02/VoIPSipPeer2/Channel00032	0	43,384	986	28
/10.200.134.170/Card02/Port02/VoIPSipPeer2/Channel00033	0	43,516	989	25
/10.200.134.170/Card02/Port02/VoIPSipPeer2/Channel00034	0	43,296	984	30
/10.200.134.170/Card02/Port02/VoIPSipPeer2/Channel00035	0	43,516	989	25
/10.200.134.170/Card02/Port02/VoIPSipPeer2/Channel00036	0	43,516	989	25

Figure 85. RTP per channel statistics in case of "Packet loss"

Test Case: Subjective Quality of Voice

Transcoding errors

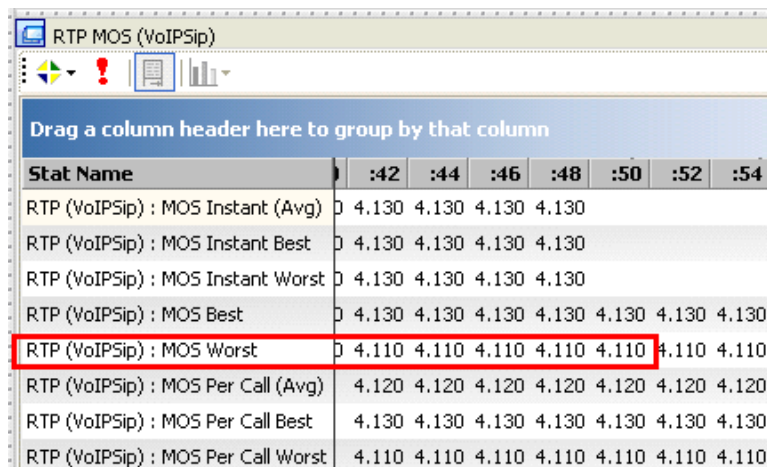
If there are no degradations in the packet transmission, then the MOS score will be perfect, but the PESQ score can still be lower than expected. This is an indication of errors in the speech processing component of the DUT. The **RTP QoS** statistics in the following table shows no loss, errors or delay of the packets.



Stat Name	:42	:44	:46	:48
RTP (VoIPsip) : Packets Sent	1,728,291	1,804,749	1,841,960	1,848,000
RTP (VoIPsip) : Packets Rec...	1,725,451	1,801,949	1,839,185	1,848,000
RTP (VoIPsip) : Bytes Sent	76,044,804	79,408,956	81,046,240	81,312,000
RTP (VoIPsip) : Bytes Recei...	75,919,844	79,285,756	80,924,140	81,312,000
RTP (VoIPsip) : Throughput ...	20,344	12,286	6,549	525
RTP (VoIPsip) : Throughput ...	20,355	12,216	6,553	999
RTP (VoIPsip) : Tx Packets ...	0	0	0	0
RTP (VoIPsip) : Lost Packets	0	0	0	0
RTP (VoIPsip) : Maximum Co...	0	0	0	0
RTP (VoIPsip) : Packet Error...	0	0	0	0

Figure 86. RTP QoS in case of transcoding errors

Consequently, the MOS score is perfect:

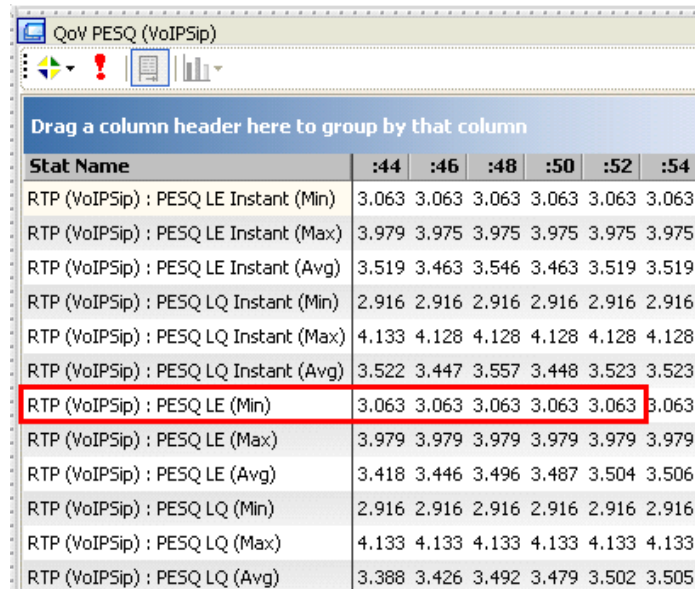


Stat Name	:42	:44	:46	:48	:50	:52	:54
RTP (VoIPsip) : MOS Instant (Avg)	4.130	4.130	4.130	4.130			
RTP (VoIPsip) : MOS Instant Best	4.130	4.130	4.130	4.130			
RTP (VoIPsip) : MOS Instant Worst	4.130	4.130	4.130	4.130			
RTP (VoIPsip) : MOS Best	4.130	4.130	4.130	4.130	4.130	4.130	4.130
RTP (VoIPsip) : MOS Worst	4.110	4.110	4.110	4.110	4.110	4.110	4.110
RTP (VoIPsip) : MOS Per Call (Avg)	4.120	4.120	4.120	4.120	4.120	4.120	4.120
RTP (VoIPsip) : MOS Per Call Best	4.130	4.130	4.130	4.130	4.130	4.130	4.130
RTP (VoIPsip) : MOS Per Call Worst	4.110	4.110	4.110	4.110	4.110	4.110	4.110

Figure 87. MOS values in case of transcoding errors

Test Case: Subjective Quality of Voice

Although the MOS score is perfect, the PESQ LE score is 3.063, which is significantly lower than the expected 3.512.



Stat Name	:44	:46	:48	:50	:52	:54
RTP (VoIP5ip) : PESQ LE Instant (Min)	3.063	3.063	3.063	3.063	3.063	3.063
RTP (VoIP5ip) : PESQ LE Instant (Max)	3.979	3.975	3.975	3.975	3.975	3.975
RTP (VoIP5ip) : PESQ LE Instant (Avg)	3.519	3.463	3.546	3.463	3.519	3.519
RTP (VoIP5ip) : PESQ LQ Instant (Min)	2.916	2.916	2.916	2.916	2.916	2.916
RTP (VoIP5ip) : PESQ LQ Instant (Max)	4.133	4.128	4.128	4.128	4.128	4.128
RTP (VoIP5ip) : PESQ LQ Instant (Avg)	3.522	3.447	3.557	3.448	3.523	3.523
RTP (VoIP5ip) : PESQ LE (Min)	3.063	3.063	3.063	3.063	3.063	3.063
RTP (VoIP5ip) : PESQ LE (Max)	3.979	3.979	3.979	3.979	3.979	3.979
RTP (VoIP5ip) : PESQ LE (Avg)	3.418	3.446	3.496	3.487	3.504	3.506
RTP (VoIP5ip) : PESQ LQ (Min)	2.916	2.916	2.916	2.916	2.916	2.916
RTP (VoIP5ip) : PESQ LQ (Max)	4.133	4.133	4.133	4.133	4.133	4.133
RTP (VoIP5ip) : PESQ LQ (Avg)	3.388	3.426	3.492	3.479	3.502	3.505

Figure 88. PESQ values in case of "Transcoding errors"

The source of this degradation is a malfunction of the DUT's transcoding module under stress conditions.

Extension to Other Protocols

Quality of voice analysis using PESQ is performed on RTP traffic and is independent of the call establishment protocol. The same test methodology can be used for different signaling protocols. The same methodology applies when different signaling protocols are used by each party, for example, H.323 on one side and SIP on the other.

Conclusions

Quality of voice is the most important measurement of a voice communication system's QoS, as experienced by users. Two methods and metrics are commonly used: MOS based on E-Model and MOS based on PESQ. E-Model is computed using transmission metrics and associated constants to approximate the degradation induced by sampling and coding of the speech signal. PESQ compares the audio signal received with the expected one. PESQ requires much higher processing power than E-Model, but has the capability to detecting degradations that are not visible with E-Model.

If a DUT only operates at the packet level and does not affect the audio streams, MOS scores based on the E-Model will provide a good and complete characterization of the expected Quality of Voice. If the DUT performs media operations, such as AGC, VAD, decoding/encoding, an in-depth analyze of QoV should be performed by the perceptual, full reference PESQ method.

Test Case: Determining the Maximum Transaction Rate for VoIP Protocols

Overview

With respect to VoIP bandwidth consumption, signaling protocols only account for a small fraction of the total traffic, with the media segment consuming most of the bandwidth. This often results in network design that maximizes subscribers' QoE based solely on media requirements. Signaling requirements, however, play a crucial role in the call setup process, impacting subscribers' QoE as much as media.

Signaling protocols are used mainly to establish calls between two or more endpoints. The associated performance metric is the CSR and is measure in calls per second (CPS). This is covered in Test Case: Determining the Maximum Call Setup Rate (CPS) in this booklet. Signaling protocols are used for other purposes as well; they need to be designed into networks to ensure positive QoE and avoid service interruption.

With respect to the messages exchanged during a SIP call, six signaling messages are used with a total payload of approximately 40kb. For a call duration of 3 minutes using a G.711 CODEC (64Kbps), the signaling represents less than 1 percent of the VoIP traffic.

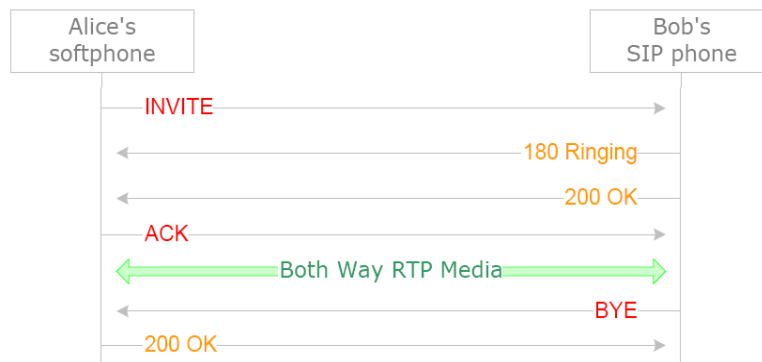


Figure 89. Basic SIP call flow

Most VoIP networks require that devices register before being able to originate or terminate a call. The registration traffic is insignificant, but may be an issue if large numbers of devices register at the same time. The primary causes for registration flooding are:

- **Power outages** – when power returns to a region all at once, all VoIP devices will initiate registration transactions at the same time.
- **Defective infrastructure device** – if an edge device goes down, serviced endpoints will attempt to re-register, causing signaling messages to flood the backup device.

Test Case: Determining the Maximum Transaction Rate for VoIP Protocols

These situations are relatively rare and are addressed in the protocol specifications (for example, Retransmissions and Configurable timeouts). The system will converge and all devices will eventually become active.

The most significant source of high volume signaling traffic is because of the periodic transmission of messages related to the following:

- Keep alive mechanisms
- Content embedded with signaling messages, as in Instant Messaging
- Misconfigured or defective endpoints
- Network attacks

SIP itself does not define a keep-alive mechanism. As a result, there are various implementations that re-use other SIP messages. Common mechanisms include use of the Register message, the Invite message during a call or the Option message – although some devices do not support this message.

The remainder of this booklet discusses the methodology and configuration steps needed to determine the transaction rate for Registration and Instant Messaging.

Registration Rate

In a traditional wired network, each device has a fixed location that the switch determines from its associated circuit. In VoIP networks, a phone's location is not pre-determined; a subscriber can move his soft IP phone and plug it in anywhere in an IP network. It is necessary for each VoIP device to identify itself to the network and the network must grant access to the device.

SIP registration is performed in one or two transactions depending on whether the registration server is configured to require authentication or not.



Figure 90. SIP register without authentication

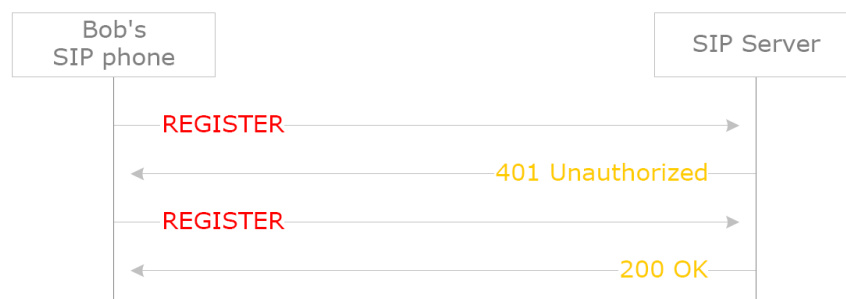


Figure 91. SIP register with authentication

Test Case: Determining the Maximum Transaction Rate for VoIP Protocols

After an endpoint has registered, the network has the IP address location of the endpoint. In the case of SIP, the Registrar server holds that information. Using the known location of an endpoint, SIP proxies will know how to route a connection. Even after a successful registration, an endpoint may become unavailable without the Registrar server or SIP proxies becoming aware of the event. To protect against this, the Registrar server may ask the endpoint to periodically retransmit its registration method.

If the endpoint and Registrar/SIP server are separated by a NAT device, the register message also has the role of opening a firewall pinhole from the server to the endpoint.

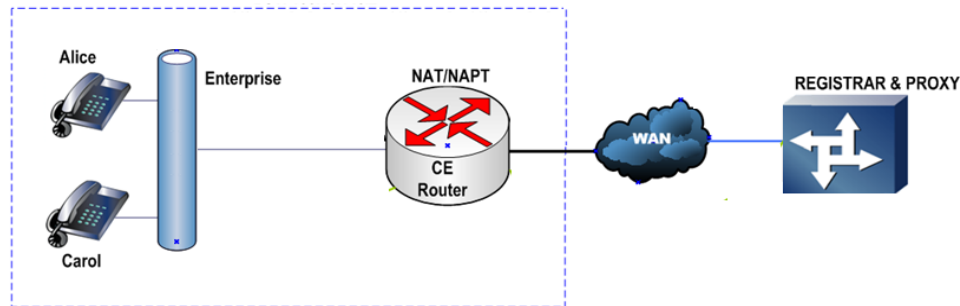


Figure 92. SIP endpoints behind a firewall

NAT devices will discard any messages originating from a SIP proxy server located in the public network unless an initial message from an endpoint has been sent to the Registrar and Proxy. Even after the pinhole has been opened, it only stays open for a limited time. To ensure that the endpoint is reachable at all times, it must retransmit messages to the SIP servers through the NAT device.

The interval between retransmits must be small enough to keep the pinhole open – typically 30 seconds. If there are a large number of endpoints, this can add up to a great deal of traffic – more than that associated with call setup. For example, if there are 1 million users behind a NAT device who register every 30 seconds without authentication (only 2 messages are required), there will be 33,333 registration requests per second. For the same number of users with a nominal call rate of 2 calls per hour, there will be 556 calls per second. This equates to 133 Mbps of bandwidth for registration versus 11 Mbps for call control.

With a transaction rate 30 times larger than call control and with a bandwidth request 12 times higher, the registration rate requires high performance from the network and VoIP devices, especially SBCs and registrar servers.

Message Rate

In addition to call control, SIP defines a mechanism to support instant messaging. It consists of a two SIP message exchange; the originator sends a MESSAGE method followed by a response from the receiver.

Test Case: Determining the Maximum Transaction Rate for VoIP Protocols

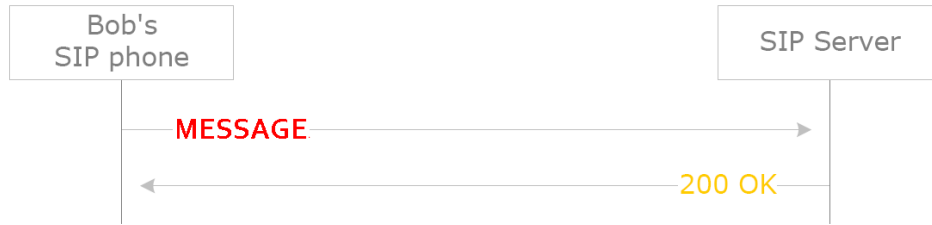


Figure 93. Instant messaging transaction

In spite of its simplicity, the proliferation of instant messaging has placed a high load on networks and VoIP devices, especially SBCs and SIP proxies.

Objective

This test attempts to establish a steady 500 registrations per second. The test demonstrates how to use the transaction rate test objective to measure the registration or instant messaging rate.

The following instructions explain how to configuring IxLoad to emulate SIP endpoints and SIP Registrar server.

Setup

A pair of Acceleron ports is used to simulate 8000 user endpoints in a medium-sized Enterprise deployment that uses a Registrar server in the public network. The test runs for 5 minutes.

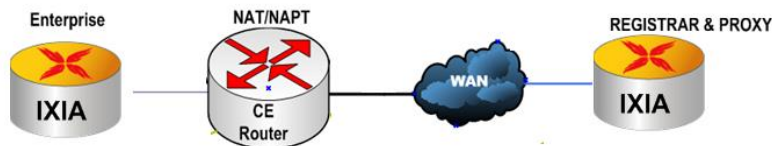


Figure 94. Test Topology – IxLoad generates and receives the traffic passing the NAT

Step-by-Step Instructions

The final IxLoad configuration as a result of these steps is provided on blackbook.ixiacom.com Web site - see *IxLoad 5.10 Voice - Registration per Second.crf*. To import a Compressed Repository File (crf) in IxLoad, use the command **Import** under the **File** menu.

The step-by-step instructions highlight how to set the essential parameters of this configuration and explain additional options, which may be used to change the behavior of the test.

Open the Configuration Template

1. Start the IxLoad GUI.
2. Create and configure two **NetTraffics**. In this example, the networks are named **Public** and **Private**. Set **8000** IP addresses that will be used for emulated endpoints on the **Originate** side and **8001** IP addresses on the **Terminate** side.

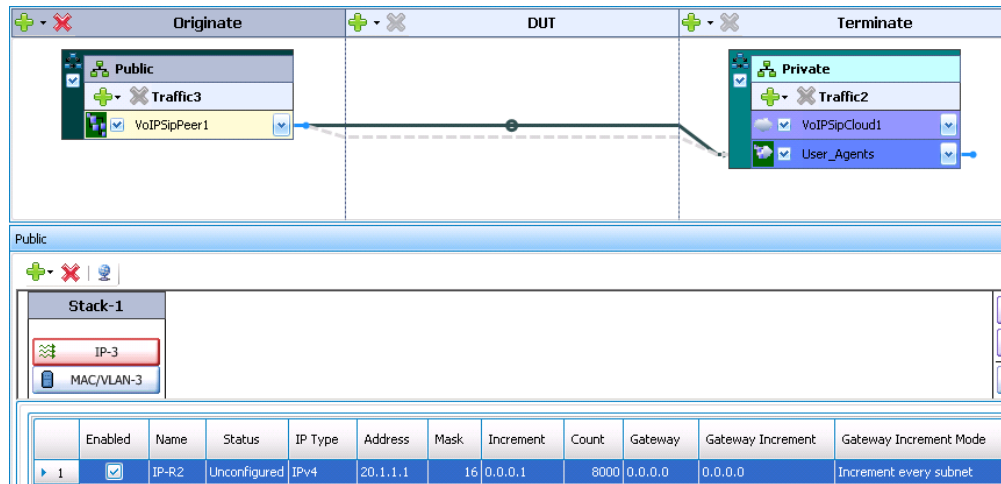


Figure 95. Originating network IP range

	Enabled	Name	Status	IP Type	Address	Mask	Increment	Count	Gateway	Gateway Increment	Gateway Increment Mode
1	<input checked="" type="checkbox"/>	IP-R1	Unconfigured	IPv4	20.1.100.1	16	0.0.0.1	1	0.0.0.0	0.0.0.0	Increment every subnet
▶ 2	<input checked="" type="checkbox"/>	IP-R3	Unconfigured	IPv4	20.1.101.1	16	0.0.0.1	8000	0.0.0.0	0.0.0.0	Increment every subnet

Figure 96. Terminating network IP ranges

On the **Terminate** network, the first range, with 1 IP address, defines the IP address of the emulated SIP Registrar sever. The second range, with 8000 IP addresses, is used to map the test scenarios that execute the call flow for the registration requests. Should a single IP address has to be used in the second range, **Channel mapping rules for SIP UA** will be set to **Use same value (per port) for IP address** and **UDP/TCP/TLS port** will be set to **Use consecutive values (per port)**; when a SIP Peer activity is used under a SIP cloud, the emulated endpoints have to use distinct IP address / UDP Port tuples.

3. On the **Terminate** network, associate each IP ranges with one of the activities. To access the **IP mapping** configuration window click on **Traffic**, and then on the tab **IP Mappings**.

Test Case: Determining the Maximum Transaction Rate for VoIP Protocols

4. Add a SIP peer activity on the **Originate** network, and **SIP Cloud** and **SIP Peer** activity on the **Terminate** network.

Traffic2		Activities & Endpoints		
Network Ranges By Port Distribution Group	VoIP5ipCloud1	User_Agents		
	VoIP5ipCloud	SIP	RTP	
<input checked="" type="checkbox"/> Group1: Consecutive IPs				
Network Range IP-R3 in Private (20.1.101.1+8000)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/> rr1: IP Round Robin				
Network Range IP-R1 in Private (20.1.100.1+1)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figure 97. IP Mappings on Terminate network

The IP ranges groups must be set to **Round Robin** for the SIP server (**VoIP5ipCloud**) and **Consecutive IPs** for the User Agents.

5. Create the test scenario, editing the following call flow on two channels:

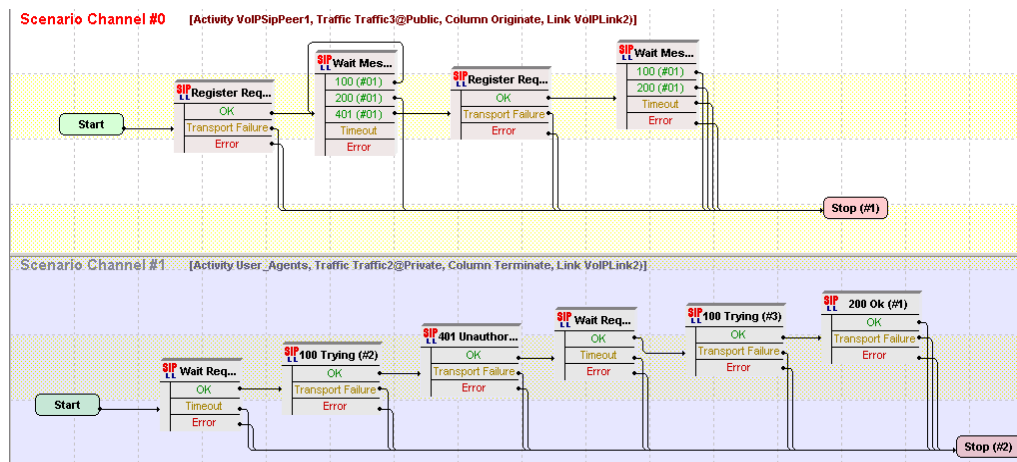


Figure 98. Register test scenario

Channel #0 sends the registration request, and then waits for the response. If the response is *401 Unauthorized*, then a new registration request is sent, this time with following authorization header:

```
REGISTER |AUTO-REQUEST-URI| SIP/2.0
Via: |AUTO-VIA|
From: |AUTO-FROM|
To: |AUTO-TO|
Call-ID: |AUTO-CALL-ID|
CSeq: |AUTO-CSEQ|
Contact: |AUTO-CONTACT|
Max-Forwards: 70
Content-Length: |AUTO-CONTENT-LENGTH|
Authorization: |AUTO-AUTHORIZATION|
```

Test Case: Determining the Maximum Transaction Rate for VoIP Protocols

The call flow on the second channel emulates the SIP registrar with Authorization enabled. It waits for a Registration request and responds with *100 Trying* and *401 Unauthorized*. It then waits for a new Registration request with an Authorization header and responds with *100 Trying* and finally *200 OK*. Using this call flow, every emulated endpoint will exhibit the following traffic:

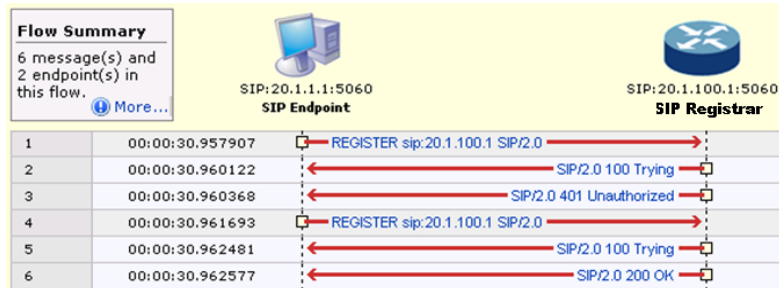


Figure 99. SIP Registration traffic captured on an emulated SIP Endpoint

If the use case requires emulation of a SIP registrar server without Authorization, then the test scenario should be changed. On Channel #1, which is the call flow of the SIP Registrar server, connect the **Ok** output of the first **Send 100 Trying** function to the **Send 200 Ok** function. In this way the **Send 401 Response** and **Wait for a second Registration request** functions are skipped.

- Define the **Dial Plan** and destination IP for the emulated endpoints (the **Originate** network traffic). The source dial plan should be configured according to the test procedure specifications. The destination IP address is not used in this type of test.

The configuration window is divided into two main sections: Source and Destination.

Source Section:

- IPs: The source IP addresses are taken from the associated Network (see Traffic - Network mapping tables in the test)
- Phone numbers:
 - Phone book entry: <None>
 - User defined: 160[00000000-]
 - Use Tel URI parameters: phone-context=example.com

Destination Section:

- IPs: None
- Override phone numbers from destination activity:
- Phone book entry: <None>
- User defined: 170[00000000-]
- Use Tel URI parameters: phone-context=example.com

Buttons at the bottom: Phone book: Edit ... | Verify all settings | Restore defaults

Figure 100. Emulated endpoints dial plan

- Set the authorization credentials of the emulated SIP endpoints. This step is needed only for authorization enabled use cases.

In the SIP property page of the emulated endpoints (the **Originate** network traffic), define the username and password sequences under **Authentication UAC**. For back-to-back setups in which IxLoad simulates both the endpoints and the registrar server, the actual values of these fields are not important because the emulated Registrar server

Test Case: Determining the Maximum Transaction Rate for VoIP Protocols

does not validate them. When the DUT is a real Registrar server, however, the credentials must match the settings on the server or the registration will fail.

The screenshot shows the SIP settings configuration page for emulated endpoints. The page is divided into several sections: 'Transport settings', 'Authentication UAC', 'Type Of Service', 'Use external server', and 'Construction of SIP messages'. The 'Use external server' section is highlighted in yellow. The 'Server address' field is set to '20.1.100.1', 'Server port' is '5060', and 'Domain name or local IP' is '20.1.100.1'. The 'Registrar server' checkbox is checked, and 'Auto register simulated user agents' is unchecked. The 'Override registrar' field is set to 'IP:PORT'. The 'Construction of SIP messages' section has several unchecked options, including 'Override default contact settings', 'Override default destination domain name or host:port', 'Use Tel URI scheme for Source', and 'Use Tel URI scheme for Destination'. The 'SIP Port' field at the top is set to '[5060-]'. The 'Maximum message size on UDP' is set to '1024'. The 'User name' is 'User160[00000000-]' and the 'Password' is 'Pass160[00000000-]'. The 'TOS/DSCP' is set to 'Best Effort (0x00)'. At the bottom, there are buttons for 'Verify all settings' and 'Restore defaults'.

Figure 101. SIP settings for emulated endpoints

8. Define the SIP registrar IP address.

To register a SIP device the IP address of the registrar server must be known. This can be specified in the SIP settings property page for the **VoIPPeerSIP** activity. Refer to above figure.

On the emulated endpoints activity (the **Originate** network traffic), select the following check boxes:

- **Use external server.** Place the IP address of the Registrar server in the **Server address**. In more complex networks this field will contain the IP address of the proxy or session border controller serving the emulated endpoint. The **Outbound Proxy** check box should be selected if you want all the SIP messages to be sent through this server—this is a typical case for NAT topologies.
- **Registrar Server.** By default, the IP address specified in the **Server address** field is used as the IP address of the Registrar server. When the Proxy and Registrar servers have different IP address, the IP address of the Registrar server may be entered in the **Overwrite registrar** field. IxLoad has the ability to automatically send registration requests at the beginning of the test; this simplifies the call flow, allowing the user to concentrate on the call feature under test. For a registration test case, the automated behavior should be disabled; **Auto register simulated endpoints** must remain cleared.

Test Case: Determining the Maximum Transaction Rate for VoIP Protocols

- Define the Dial Plan and destination IP on the emulated SIP Registrar (**Terminate** network traffic).

The **SIP Cloud** activity uses the information in incoming SIP Register Requests to route the message to the proper resource (channel) of the **SIPVoIPPeer** activity. By default, all incoming requests are dispatched according to one of the default rules, applied in the following order:

- Phone number in the *To* header must match the phone number in the Dial Plan Source of this activity.
- Phone number in the *From* header must match the phone number in the Dial Plan Destination of this activity.

The **To** header of SIP Register request messages contain the phone number of the endpoint initiating the registration. This is different than in other SIP request messages, where the *To* header contains the information of the destination endpoint.

The **Source** dial plan on the emulated SIP Registrar activity should be configured identically to the dial plan on the **Originate** network traffic to satisfy the dispatching rules.

The screenshot shows the configuration interface for the Emulated SIP Registrar dial plan. The 'Dial Plan' tab is selected. The 'Source' section is configured with 'User defined' phone numbers set to '160[00000000-]' and 'Use Tel URI parameters' set to 'phone-context=example.com'. The 'Destination' section is configured with 'User defined' phone numbers set to '170[00000000-]' and 'Use Tel URI parameters' set to 'phone-context=example.com'. The 'Override phone numbers from destination activity' checkbox is unchecked. The 'Phone book' is set to 'Edit ...'. The 'Verify all settings' and 'Restore defaults' buttons are visible at the bottom.

Figure 102. Emulated SIP Registrar dial plan

- Define the test objective.

IxLoad uses a generic approach to determine the transaction rate with the **Loops per second** test objective. When this test objective is used, IxLoad instantiates all the channels, or emulated endpoints, and starts the call flow for the value specified in the test objective. The test scenario is built such that one registration is performed per loop. In this way, the registration rate is directly controlled by the loop rate. If the test scenario were used for a different call flow, instant messages for example, then the loop rate would control the instant messages rate.

Test Case: Determining the Maximum Transaction Rate for VoIP Protocols

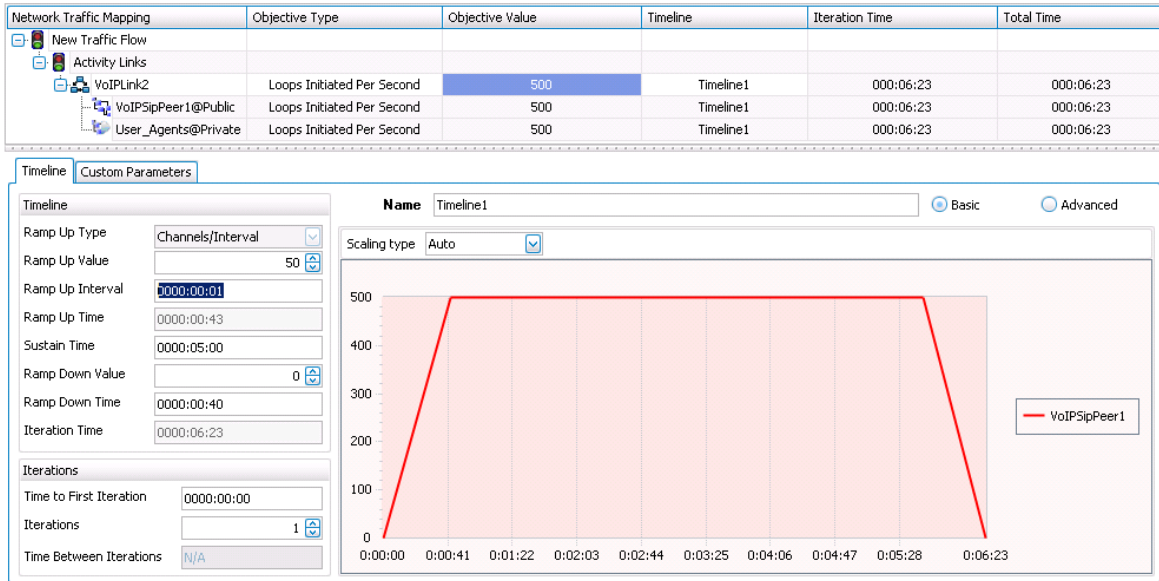


Figure 103. Test objective and Timeline

For this test, the test objective is 500 loops per second, validating whether a rate of 500 registrations per second can be maintained.

11. Map the ports; a pair of Acceleron ports is enough for this test.

Running the Test

1. Click **Run** to start the test execution.
2. IxLoad will automatically display **Statistic Views** after execution starts.

Results Analysis

The following questions provide guidelines on how to recognize problems during the test or in post analysis:

1. To determine if the test objective been achieved, check the **Loops Rate** view.

The value of the **Loops per Second** statistic should follow the time line specified in the test objective. In this example, the expectation was for a steady value of 500 loops per second. If the graph shows variations in the **Loops per Second** value that were not specified in the advanced time line, then the test has been misconfigured or the DUT cannot maintain the desired rate.

The IxLoad configuration should use enough emulated endpoints to maintain a specific rate. If the registration of one endpoint takes 2 seconds, then 1000 endpoints are required to maintain a registration rate of 500 per second. In this example, 8,000 endpoints are defined. To maintain a rate of 500 loops per second, each loop may take up to 16 seconds (8,000 / 500). If one loop requires more than 16 seconds, then there

Test Case: Determining the Maximum Transaction Rate for VoIP Protocols

won't be sufficient endpoints available to start a new loop after 16 seconds, causing the loop rate to drop.

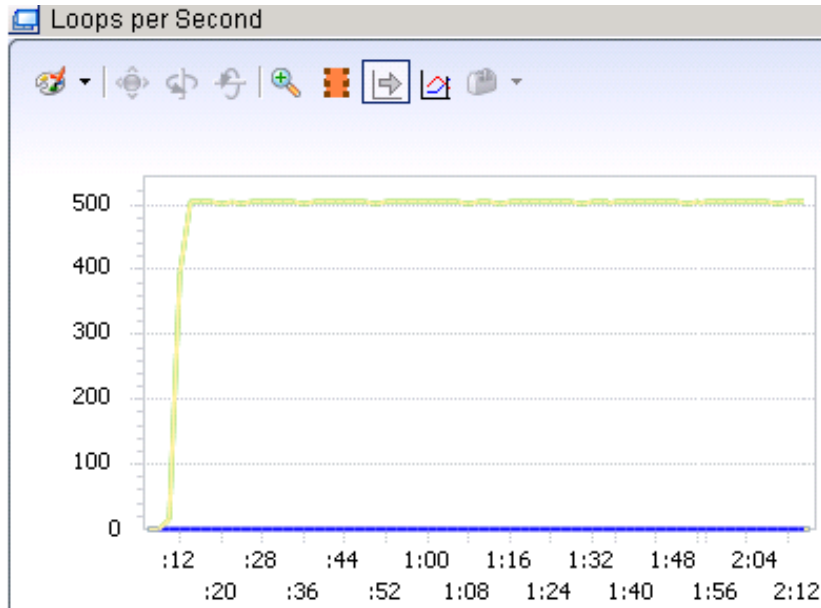


Figure 104. Loop rate view

- To determine if the registration rate has been achieved, check the **Registration Rate** view.

Statistic Name	Value	Questions
Registration initiated per second		5. Has the Registration Initiated rate attained a constant rate during the Sustain Time ?
Registration completed per second		6. How do the Registration Initiated and Registration Completed rates compare?

A constant loop rate does not necessarily mean that the registration rate was maintained; some of the loop can have failed. For example, the server can answer with a *500 Internal Error* or *401 Unauthorized*. The *500* response is an indication in the Registrar server's database or a misconfiguration of one of the SIP Register message parameters. The *401* response is an indication of a credentials mismatch.

Test Case: Determining the Maximum Transaction Rate for VoIP Protocols

- To determine if the SUT handled all SIP message properly, check the **SIP Messages** view.

Stat Name	2:20	2:22	2:24	2:26	2:28	2:30	2:32
SIP (VoIPSip) : Requests Sent	97,612	99,104	100,584	102,073	103,594	105,128	106,599
SIP (VoIPSip) : REGISTER Requests Matched	97,611	99,103	100,582	102,071	103,593	105,126	106,597
SIP (VoIPSip) : REGISTER Requests Parsed	97,611	99,103	100,582	102,071	103,593	105,126	106,597
SIP (VoIPSip) : 1xx Responses Sent	97,611	99,103	100,582	102,071	103,593	105,126	106,597
SIP (VoIPSip) : Requests Matched	97,611	99,103	100,582	102,071	103,593	105,126	106,597
SIP (VoIPSip) : Requests Parsed	97,611	99,103	100,582	102,071	103,593	105,126	106,597
SIP (VoIPSip) : 1xx Responses Matched	97,611	99,103	100,582	102,071	103,593	105,126	106,597
SIP (VoIPSip) : 1xx Responses Parsed	97,611	99,103	100,582	102,071	103,593	105,126	106,597
SIP (VoIPSip) : 4xx Responses Sent	49,363	50,104	50,902	51,607	52,308	53,043	53,769
SIP (VoIPSip) : 2xx Responses Sent	48,247	49,000	49,680	50,464	51,285	52,083	52,828
SIP (VoIPSip) : Responses Orphans	0	0	0	0	0	0	0
SIP (VoIPSip) : Requests Orphans	0	0	0	0	0	0	0
SIP (VoIPSip) : Ignored Retransmissions	0	0	0	0	0	0	0
SIP (VoIPSip) : Retransmitted Msgs	0	0	0	0	0	0	0

Figure 105. SIP messages view

The **SIP Messages** view provides information about the number of SIP messages sent and received, including retransmissions. The values shown in this view should be consistent with the call flow. In this example, two **Registration Messages** were sent, two *100* responses, one *200* response, and one *401*. That means that the number of *Register* requests and *100* responses should be double the number of *200* and *401* responses at a given time.

Due to the statistics sampling interval (every 2 seconds in this case), it is possible that small differences will be seen. For example, the number of *200* and *401* responses may not be equal, but they should balance by the end of the test.

Other statistics that should be checked include the following:

- Un-dispatched SIP Messages** under **SIP Cloud** stats: if the number of un-dispatched messages is greater than zero, then the test has been misconfigured, possibly because the dial plan of the emulated endpoints and the emulated SIP Registrar are not similar.
- VoIP SIP Errors**: no errors should occur.

Test Variables

Test Tool Variables

Table 41.

Parameter Name	Current Value	Additional Options
IP Version	IPv4	IPv6

DUT Test Variables

Table 42.

Parameter Name	Current Value	Additional Options
IP version	IPv4	IPv6
Transport Protocol	SIP/UDP	SIP/TCP, SIP/TLS
Authorization Enabled	On	Off

Extension to Other Transaction Types

Using the **Loops per Second** test objective it is possible to measure the DUT's capacity and rate for any type of SIP transaction. For example, to test the instant message rate, only a few changes are required:

- The call flow should implement that shown in Figure 93 - Instant messaging transaction.
- The dial plan must be changed on the emulated endpoints and emulated server. In the emulated endpoints dial plan (**Originate** network traffic) the destination IP addresses can use the symbolic link to the **VoIPPeerSIP** on the **Terminate** network traffic. This dial plan can be different from the dial plan of the emulated endpoints, while the test simulates sending messages from one endpoint over a proxy to other endpoint.

All other settings may remain the same. In this case one execution loop is one transaction for instant messaging, so the test objective "Loops per Second" is equivalent with "Instant Message Transactions per Second".

Conclusions

To assure a high QoE in a VoIP network, it is important to assure that the network not only possesses the capacity to sustain the desired volume of VoIP traffic, but also a transaction rate for a number of endpoint operations.

This example demonstrated in detail how to configure IxLoad to maintain a rate of 500 registrations per second for 5 minutes. With small changes the same configuration can be used to measure the rate of other type of transactions such as the instant messaging rate.

Test Case: Using VoIP to Measure NAT/PAT Performance

RFCs

RFC 1631 (May 94) - obsolete by RFC3022

The IP Network Address Translator (NAT)

RFC 2663 (Aug 99)

IP Network Address Translator (NAT) Terminology and Considerations

RFC 3022 (Jan 01)

Traditional IP Network Address Translator (Traditional NAT)

RFC 4787 (Jan 07)

Network Address Translation (NAT) Behavioral Requirements for Unicast UDP

Overview

Network address translation methods are defined by RFC 3022, which obsoleted RFC 1631, has gained a lot of popularity in today's networks. It is a common function found in routers, firewalls, operating systems, and applications.

Why Use NAT?

The primary role of NAT is to address the shortage of public IPv4Addresses. The 32 bit addressing scheme used by IPv4 allows, in theory, up to 2^{32} (4,294,967,296) IP addresses. In practice, however, the number of real IP public addresses that can be used is actually around 3.2 billion because of a special set of reusable IP addresses, which was defined by Internet Assigned Numbers Authority (IANA):

10.0.0.0	-	10.255.255.255
172.16.0.0	-	172.31.255.255
192.168.0.0	-	192.168.255.255

The three ranges of IP addresses define the private IP addresses that are used within private LANs and not routed to the Internet.

While the IPv6 addressing scheme will resolve this problem by using a larger address space consisting in 2^{128} addresses, its implementation requires the modification of the entire Internet infrastructure. For a time, IPv4 and IPv6 networks will continue to co-exist, because IPv4 provides a better solution for small and medium networks.

IP address translation is useful when a network's internal IP addresses cannot be used outside the network either for privacy reasons or because they are invalid for use outside the network.

In addition, a public network topology can change in time, for example by the service provider. Whenever external topology changes, such changes can be hidden from local domain users by centralizing those changes to a single device NAT enabled.

In this document we will discuss the traditional NAT, which consists in a combination of basic NAT and NAPT (network address port translation).

Basic NAT and NAPT are two variations of traditional NAT. Basic NAT translation is limited to IP addresses alone, whereas NAPT translation is extended to include IP address and transport identifier (such as TCP/UDP port or ICMP query ID).

In a traditional NAT, sessions are unidirectional, being originated only from the private network on outbound sessions. Sessions in the opposite direction may be allowed on an exceptional basis using static address maps for pre-selected hosts.

Network Address Translation (NAT)

NAT is a common method that allows IP addresses to be mapped from one group to another in a transparent way to end users.

NAT was designed to allow hosts from a private network to use a single device – NAT enabled – as a gateway to the public network by translating or substituting the private IP addresses of the hosts to the public IP address of the gateway.

While NAT allows a private network to connect to the public network, that is, the Internet, it also allows a private network to connect to another private network. Regardless of how the networks connect, the concept is the same and quite simple to understand. Further we will refer to the internal network as **private network** and to the external network as **public network** regardless if the external network is the Internet (WAN) or another LAN.

We will also refer to the IP addresses from the private network as **private IPs** and respectively as **public IPs** for the ones in the public network

One of NAT's requirements is to remain transparent to the network – that is, all devices from the private network are not required to be reconfigured to access the public network. This is possible by configuring the NAT device as a gateway to the public network. However, the NAT function cannot by itself support all applications transparently and often must co-exist with application level gateways (ALGs) for this reason. Except for ALGs, NAT devices do not examine or modify the payload of the packet. For this reason, NAT devices can often cause difficulties. Some situations where traditional NAT will not work are when an application payload itself includes an IP address or when end-to-end security is needed.

IPsec techniques, which are intended to preserve the endpoint addresses of an IP packet will not work with traditional NAT because protocols such as AH and ESP protect the contents of the IP headers (including the source and destination addresses) from modification. Yet, NAT's fundamental role is to alter the addresses in the IP header of a packet.

Test Case: Using VoIP to Measure NAT/PAT Performance

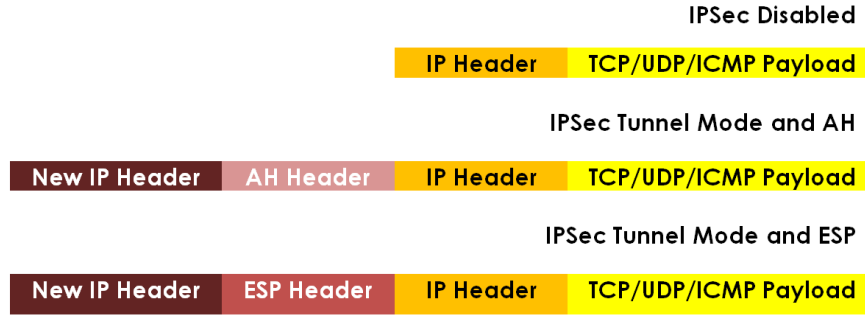


Figure 106. IPSec AH and ESP headers in Tunnel Mode

NAT Concept

The example, from the figure shown below, explains the NAT concept using a simple private network consisting in two hosts connected to the public network using the customer edge (CE) router, which is NAT enabled. The hosts use private IP addresses from class C – **192.168.1.100/24** and **192.168.1.200/24**. They are configured to use as a gateway the IP address of the CE router on the private interface: **192.168.1.1**.

In this example, the CE router uses the public IP address **75.83.202.16/22** to connect to the public network.

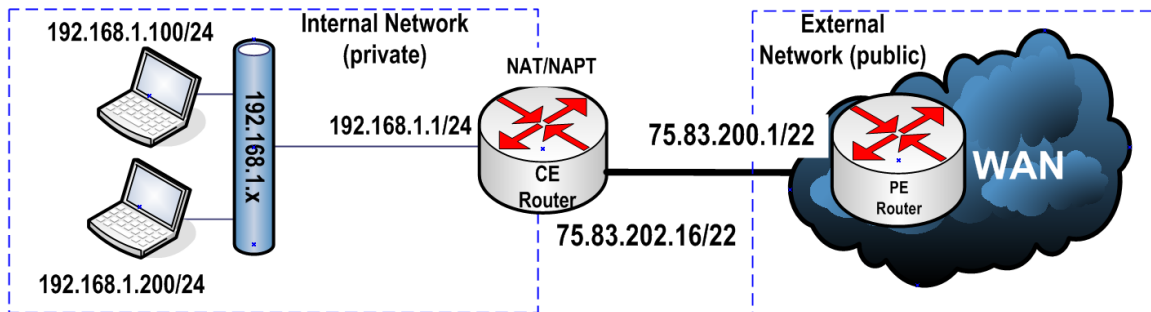


Figure 107. NAT concept explained

As you can see, all traffic from the hosts is sent to the public network through the CE router, which performs NAT on these packets. The packets are then forwarded to their destination via the provider edge (PE) router, which is configured as a gateway for the public interface of the CE router.

Now let us assume that host **192.168.1.100/24** uses an HTTP client to retrieve an HTML page from an external server with the address **209.132.176.30** (www.redhat.com). The client initiates a **GET** request using the source IP **192.168.1.100** and source port number **32,000** with the destination **209.132.176.30** and destination port **80**.

Test Case: Using VoIP to Measure NAT/PAT Performance

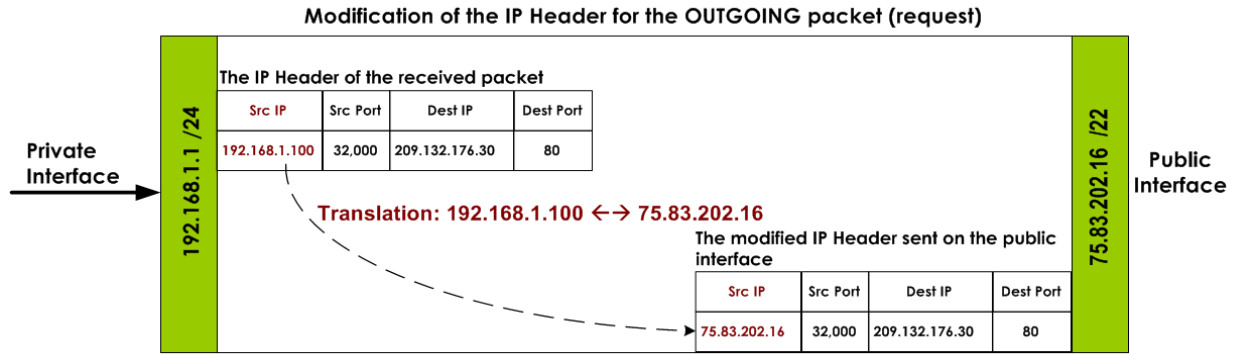


Figure 108. Processing of the first outgoing packet (the HTTP GET)

As the CE router receives each packet on its private interface, it removes the source IP address **192.168.1.100** from the IP header and replaces it with its own public IP address **75.83.202.16** while maintaining the source port number, and then calculates and modifies the IP and TCP checksum. No changes are made to the payload.

After those operations complete, the packet is sent to the PE router (**75.83.200.1**), which further routes the packet towards the initial destination (**209.132.176.30/22: 80**). When the HTTP server located in the external domain (redhat.com) receives the GET request it replies with a **200 OK** response, which is sent back using the source **209.132.176.30/22: 80** and with the IP destination.

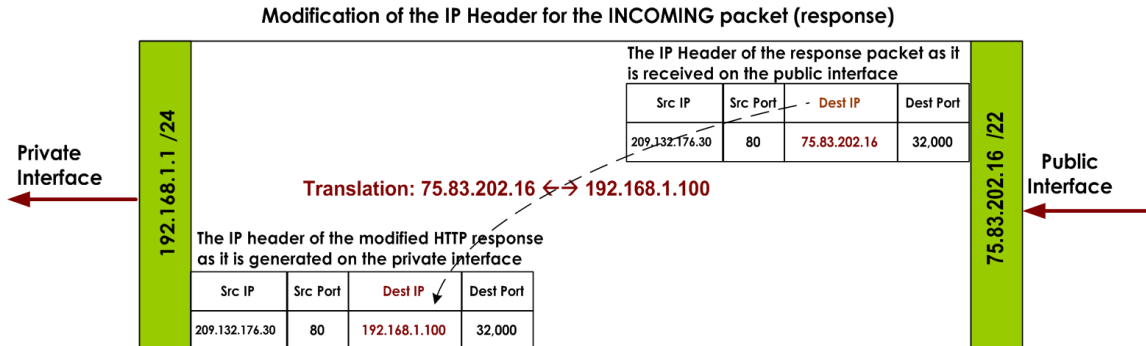


Figure 109. Modification of the INCOMING packet (HTTP Response)

In this example, we see how NAT works by substituting the source IP address from the private network one private address (in our example, **192.168.1.100**) with a public IP address (in our example, **75.83.200.1**).

If the second host **192.168.1.200** from the private network initiates a similar request while the transaction from first host is still alive, a secondary public IP address will be required. Because with basic NAT only the IP addresses are substituted, a public IP address must be available for each concurrent session.

Test Case: Using VoIP to Measure NAT/PAT Performance

Because all the packets initiated by the CE router will replace the original source address with its own public address, the external network will see the entire private network as a single device – the CE router – with the IP address **75.83.202.16/22**, as shown in the following figure.

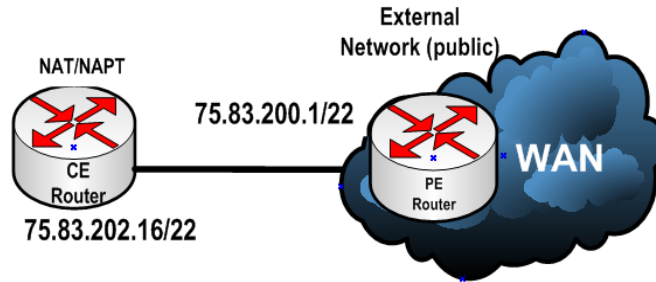


Figure 110. How the public network sees the NATed network

This example also demonstrates how NAT provides a privacy mechanism by hiding the private network from everyone in the public network. Routers and firewalls use NAT as a layer of security because a communication pinhole between the private IP and the public IP of the router/firewall is created only when a solicited request is received, that is a request or connection initiated from the private network. An unsolicited request, that is a request or a connection that is initiated from the public network is by default dropped, because no communication path exists. This behavior has the disadvantage of taking away the end-to-end significance of an IP address, which can disable some applications; VoIP is one of those applications. Various workarounds exist, but those add an extra layer of complexity and processing, which can result in a performance decrease.

The central part of all the NAT operations is the NAT table, which typically resides in the memory of the device implementing NAT. The NAT table has a dynamic behavior – it is populated as new connections are created and after the connections are closed, the associated bindings are removed. Larger NAT table allow more bindings can be tracked.

The basic NAT can be further classified in two types:

- Static NAT
- Dynamic NAT

Static NAT

As described earlier, traditional NAT sessions are unidirectional, originating from the private network while sessions in the opposite direction may be allowed on an exceptional basis using static address maps for pre-selected hosts. Most NAT devices today allow the network administrator to configure permanent entries into the NAT table, with the goal of allowing inbound sessions to reach designated devices within the private network (for example, a Web server). This type of configuration is referred to as static NAT or port forwarding.

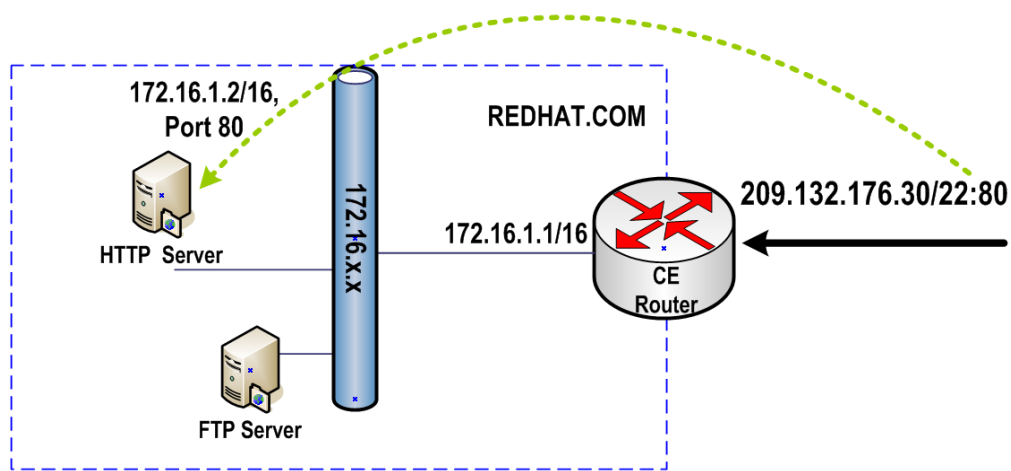


Figure 111. Static NAT for INCOMING connection

Test Case: Using VoIP to Measure NAT/PAT Performance

Static NAT can also be used for outbound sessions initiated from the private network. While this is not a common use, it can still be useful in certain situations. As described in the NAT Concept section, basic NAT requires a public IP address for each concurrent connection that may pass through the NAT. The bindings between specific private IP addresses and specific public addresses can be statically defined.

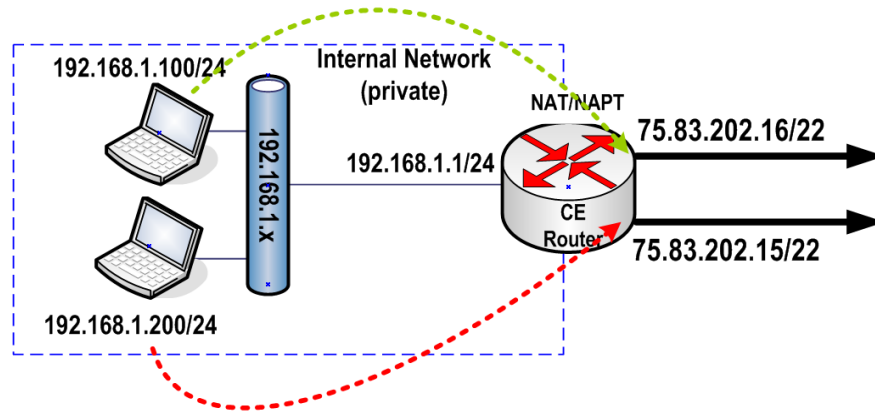


Figure 112. Two concurrent NAT translations

Dynamic NAT

Static NAT provides a one-to-one private to public static IP mapping, whereas dynamic NAT provides the same functionality by using a dynamic mapping to the public IP addresses, based on a group of publicly available IPs.

Let's assume a simple configuration consisting in three hosts located in the private network, and a NAT router that is configured with a dynamic pool of three public IP addresses. When a new outgoing connection is received from the private network, the NAT router substitutes the private IP address with the first available address in the pool. When the second host initiates an outgoing connection (assuming that the first host has its connection still active), the NAT router will use in the secondary public IP address for the translation. Similarly, the 3rd host will receive the 3rd address from the pool.

Test Case: Using VoIP to Measure NAT/PAT Performance

This example is illustrated below:

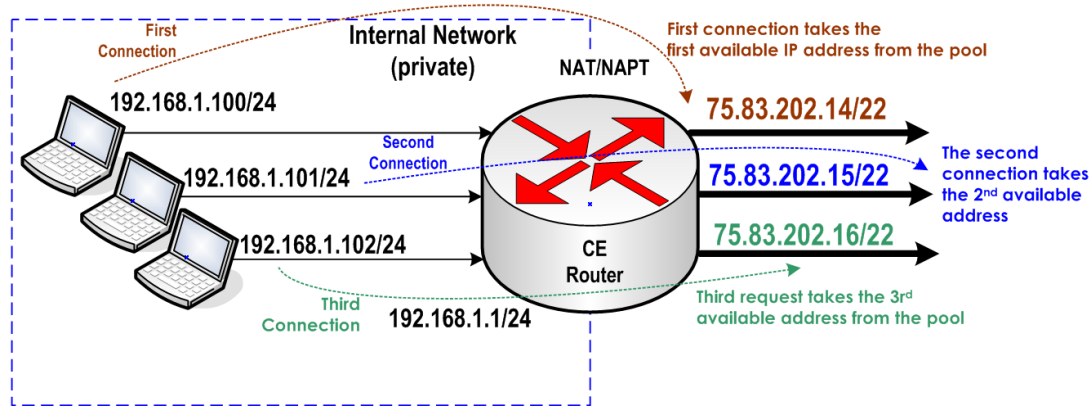


Figure 113. Dynamic NAT example

Dynamic NAT is more common in implementation that combines NAT and NAPT, rather than using it as a standalone mode.

In many cases, basic NAT allows hosts in a private network to transparently access the external network and enables access to selective local hosts from the outside. Organizations with a network setup predominantly for internal use, with a need for occasional external access are good candidates for this scheme.

This does not resolve the main problem, however, which NAT needs to resolve—IPv4 address depletion. Many organizations have multiple network nodes running TCP/UDP applications, which require Internet access, but only one public IP address assigned to their remote router. The NAPT mode resolves this issue by permitting multiple nodes in a local network to simultaneously access remote networks using the single IP address assigned to their router.

Network Address Port Translation (NAPT)

The NAPT mode is also known as port address translation (PAT), IP masquerading or as NAT overload. The different names come from the way NAPT. NAPT is the most common method used.

The difference between NAT and NAPT is quite easy to remember – NAT translates IP addresses (the source IP for outgoing connections and destination IP for their responses) while NAPT translates the ports.

NAPT includes a mix of static NAT and dynamic NAT but with a significant enhancement –port address translation support. NAPT takes a static or dynamic IP address that is bound to the public interface of the NAT enabled router and allows all hosts within the private network to access the public network.

NAPT Explained

Assume a host on a private network has an IP address of **192.168.1.100**, which sends an HTTP GET packet to a public HTTP server with the IP Address **209.132.176.30**. As the original packet passes through the NAPT enabled router, the source IP address field is changed by the router from **192.168.1.100** to **209.132.176.30**. However, because this was the first and the only connection active, the source port number field is not changed. This is better illustrated in the following figure.

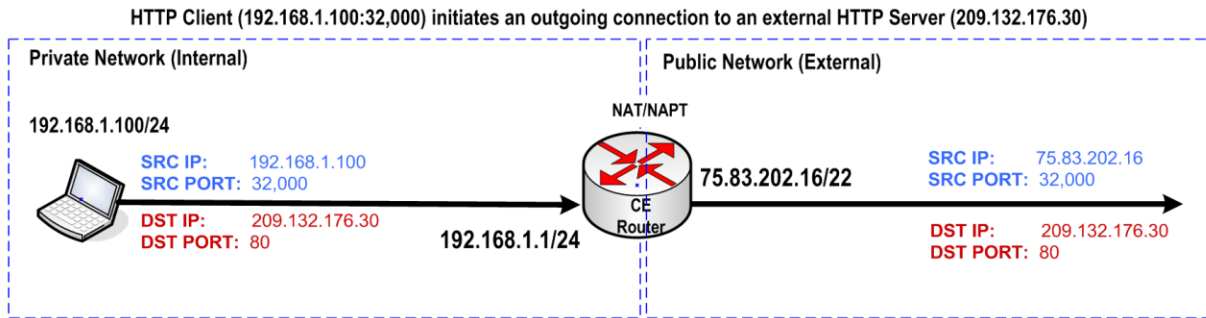


Figure 114. Outgoing HTTP request

The incoming response from the HTTP server uses an IP address of 209.132.176.30 and port 80. Because the HTTP server received the packet from the public address of the NAPT enabled router, **75.83.202.16**, using port **32,000** it will reply back to this address.

When the router receives the response packet, it will strip off the destination IP address and replace it with the host address **192.168.1.100**, after performing a lookup operation in the existing entries of the NAT table. The same lookup operation retrieves the destination port as well. In this case, it remains unchanged.

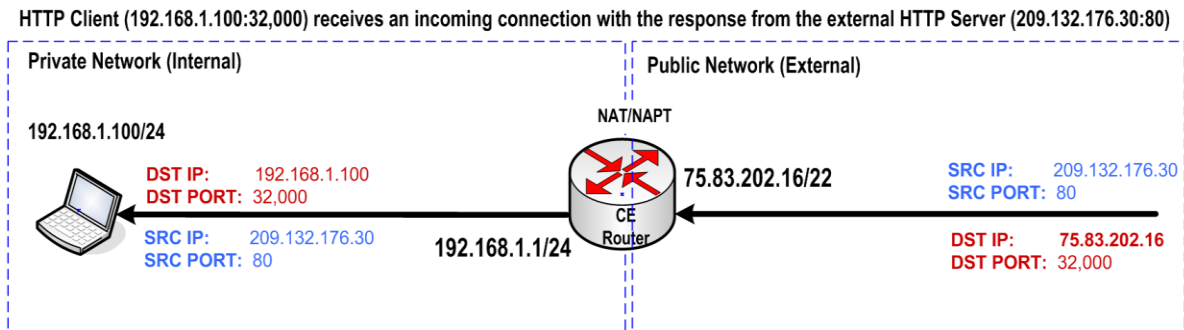


Figure 115. Incoming connection (HTTP response)

To better explain the difference between NAPT mode and basic NAT mode, we need to add a secondary host in the private network, let's say **192.168.1.101**, and see what happens when a single public IP address **75.83.202.16** is configured on the public interface of the NAPT enabled router.

Let us also assume the second host initiates a telnet connection from port 2500 to an external telnet server **207.46.197.32**, which listens on port **23** for telnet connections. If the first

Test Case: Using VoIP to Measure NAT/PAT Performance

connection is still active, basic NAT will simply not work, because a secondary public IP address must be. With NAT, the router can reuse the same public IP address **75.83.202.16** but a different port other than 32,000, which was used in the first connection.

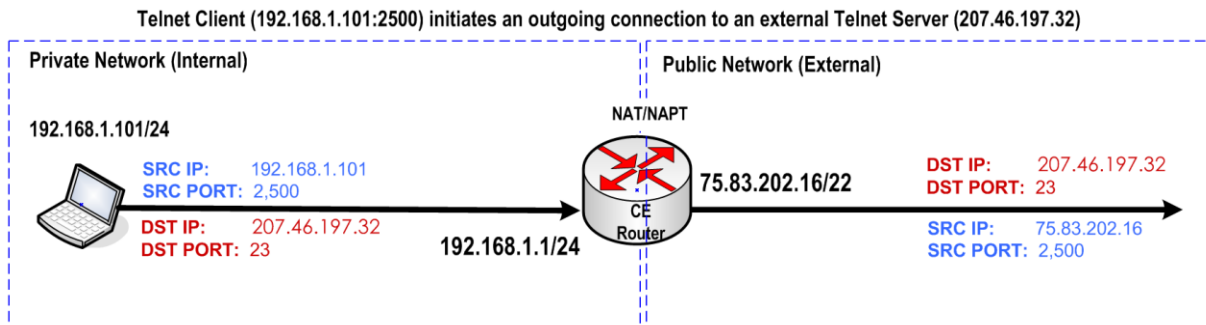


Figure 116. Outgoing Telnet client connection

After the router receives the response packet, it will perform a lookup in the NAT table and strip off the destination IP address and replace it with the host address **192.168.1.101**. The same lookup operation retrieves the destination port as well, which remains unchanged.

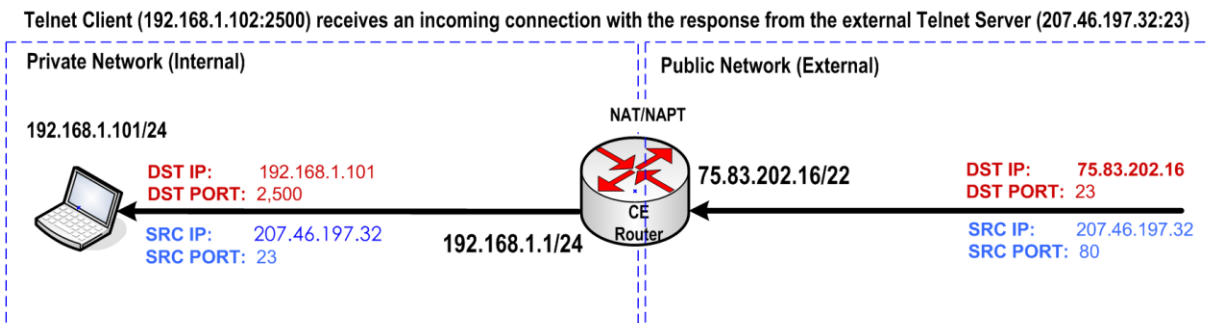


Figure 117. Incoming connection (Telnet response)

This example demonstrated how two hosts can share a public IP address in the same time.

Now let's see how NAT works when two hosts initiates the same type of connection using same source port and also uses the same destination IP address and port. In fact, the outbound destination and port will not change the behavior of the NAT enabled router, but two connections sharing the same source port number affects the behavior of the router. As we described in the previous examples, the port numbers are preserved.

The next figure illustrates how the outgoing sessions are handled when they use the same port. For both connections, NAT changes the source address to its public address **75.85.202.16**. For the first connection, the port number is not changed because no other connection exists with port **32,000**. However, because the second connection is initiated also from the same port **32,000**, NAT changes the port number to the first application port available, which is different than the active port numbers on the public interface. In our example, the source port number is changed to 1025.

Test Case: Using VoIP to Measure NAT/PAT Performance

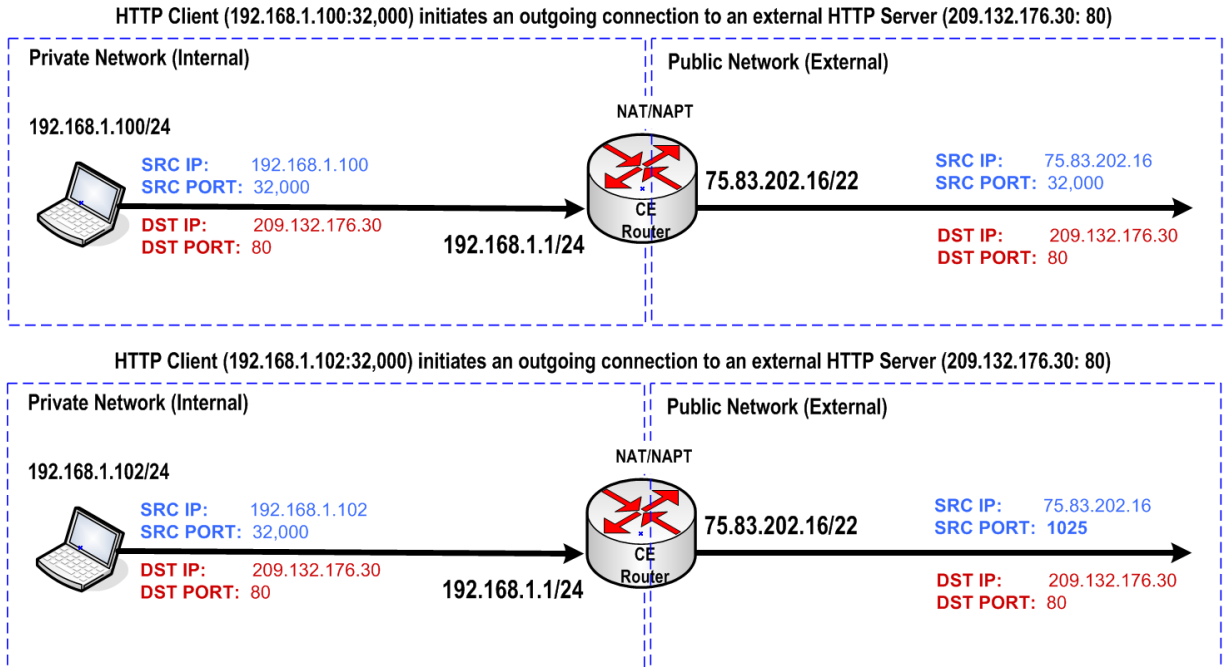


Figure 118. NAT/PAT example when two hosts initiate an external connection using the same port

Port numbers between 1 and 1023 are reserved for well-known service port numbers. Applications port numbers start with 1024 and end with port number 65535.

Full Cone NAT, Restricted Cone NAT, Port Restricted Cone NAT, and Symmetric NAT

Based on how NAT devices handle the UDP traffic we can differentiate the following NAT types:

- **Full Cone NAT**

All requests from the same internal IP address and port are mapped to the same external IP address and port. By sending a packet to the mapped external address, any external host can send a packet to the internal host.

- **Restricted Cone**

Uses the same IP and port mapping as a full cone NAT, but unlike it, an external host, with IP address **IP1**, can send a packet to the internal host only if the internal host had previously sent a packet to IP address **IP1**.

- **Port Restricted Cone**

Port restricted cone NAT is similar with a restricted cone NAT, but the restriction includes port numbers rather than an IP address. For example, an external host can send a packet, with source IP address **IP1** and source port **P1**, to the internal host only if the internal host had previously sent a packet to IP address **IP1** and port **P1**.

- **Symmetric NAT**

All requests from the same internal IP address and port, to a specific destination IP address and port, are mapped to the same external IP address and port. If the same host sends a packet with the same source address and port, but to a different destination, a different mapping is used. Furthermore, only the external host that receives a packet can send a UDP packet back to the internal host.

NAT Capacities and Performance

The capacity of the NAT table depends on the amount of memory that it has available. However, the required memory per translation is relatively small – a few hundred bytes – and today's devices are typically equipped with enough memory to support a large number of NAT translations.

NAT performance typically depends on several factors, including:

- The NAT mode (NAT versus NAPT)
- The type of application traffic
- The type of applications running concurrently
- The number of messages passing the NAT - more messages results in more translations

Translation Phases

The translation phases of traditional NAT are as follows:

- **Address binding**

When an outgoing session originates from the private network, the private address that initiated the session is bound to an external address – the public address that is used to route the message further.

In the case of NAPT, where many private addresses are mapped to a single public address, the binding would be from the tuple of [private address, private TCP/UDP port] to the tuple of [assigned address, assigned TCP/UDP port]. As with basic NAT, this binding is determined when the first outgoing session is initiated by the [private address, private TU port]tuple of on the private host.

- **Address lookup and translation**

After an address binding or [address, TCP/UDP port] tuple NAPT binding is established, a soft state is maintained for each of the connections using the binding. Packets belonging to the same session will be subject to session lookup for translation purposes.

- **Address unbinding**

When the last session based on an address or [address, TCP/UDP port] tuple binding is terminated, the binding itself may be terminated.

NAT Traversal for VoIP

The number of broadband users has increased significantly in the last years, leading to a faster adoption of the VoIP services by both home users and enterprises. Regardless of the endpoint type, that is, (soft phone or hard phone) or service type (residential or business), the majority of the VoIP endpoints use private IP addresses that are mapped to public IP addresses using NAT/NAPT, a common function of the broadband access routers. Additionally, VoIP endpoints may be situated behind one or more firewalls.

The following characteristics of the VoIP protocols impose challenges while traveling through NAT:

1. A VoIP call can be seen as two separate sessions, which needs to be correlated:
 - a. A signaling session that uses protocols such as SIP, H323, MGCP, and H248 that:
 - Establishes and tears down a media connection
 - Negotiate the common set of capabilities
 - Agree on the source and destination IP address and port(s)
 - b. A media session, which represents the actual conversation that happens between the IP addresses and media ports dynamically negotiated by the signaling session.
2. A VoIP session uses the source and destination IP address and port numbers inside the IP payload.
3. A VoIP user may be either the call originator or call terminator, regardless of his location on a private network or public network.
4. A VoIP call uses even port numbers for RTP traffic and next available odd number for RTCP; deviation from this rule will break the protocol conformance with RFC 3550.

We will use a SIP example to better understand why those characteristics represent a challenge to NAT'ed networks. The same concepts can be extended to all VoIP protocols. In this example, we will use a simplified service provider network diagram.

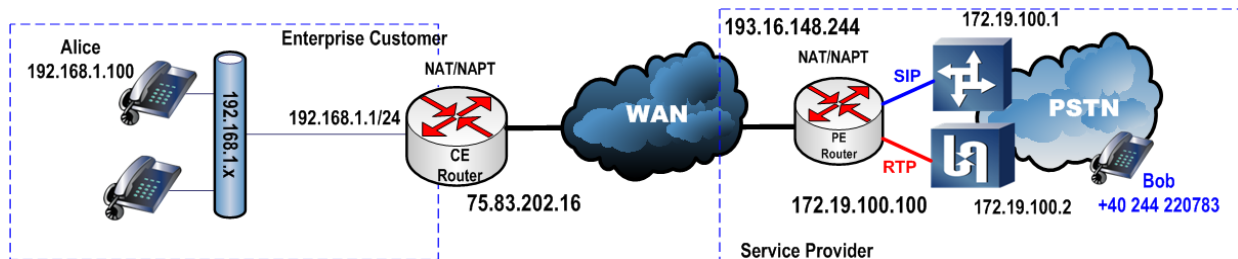


Figure 119. VoIP NAT Traversal

Test Case: Using VoIP to Measure NAT/PAT Performance

The IP phones in the enterprise network use the **192.168.1.x** private address range and are placed behind a NAT/NAPT device that translates the private addresses to a single external address **75.83.202.16**.

The softswitch and media gateway located in service provider's network are also placed behind a NAT/NAPT device that translates the **172.19.100.x** private address range to an external address **193.16.148.244**.

The IP phones from the enterprise network are configured to use **193.16.148.244:5060** as their outbound proxy, as instructed by the service provider.

Let's assume Alice wants to call her friend Bob. Before placing the call, her phone must register by sending a REGISTER request to REGISTRAR's public IP: **193.16.148.244:5060**.

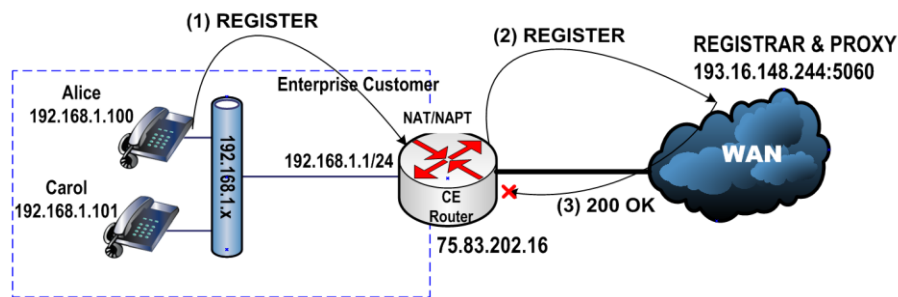


Figure 120. SIP Registration fails when attempted behind a NAT/NAPT device

The **(1) REGISTER** request initiated by Alice's phone has the following content:

SRC IP	SRC PORT	DEST IP	DEST PORT
192.168.1.100	5060	193.16.148.244	5060
<pre>REGISTER sip:myserviceprovider.com SIP/2.0 Via: SIP/2.0/UDP 192.168.1.100:5060 From: "Alice" <sip:8184443118@myserviceprovider.com>;tag=3261c4561 To: "Alice" <sip:8184443118@myserviceprovider.com>; Contact: "Alice" <sip:8184443118@192.168.1.100:5060> Content-Length: 0</pre>			

Figure 121. IP header and IP payload in "(1) REGISTER"

On receipt of the REGISTER request, the NAT device located at the edge of the enterprise network changes the source IP address and port number from the IP header of the REGISTER packet, leaving the destination IP address and port unchanged. The payload is not modified. The first available port number is selected for the port mapping; let us assume 1025. The REGISTER request has the following structure:

Test Case: Using VoIP to Measure NAT/PAT Performance

SRC IP	SRC PORT	DEST IP	DEST PORT
75.83.202.16	1025	193.16.148.244	5060
REGISTER sip:myserviceprovider.com SIP/2.0 Via: SIP/2.0/UDP 192.168.1.100:5060 From: "Alice" <sip:8184443118@myserviceprovider.com>;tag=3261c4561 To: "Alice" <sip:8184443118@myserviceprovider.com>; Contact: "Alice" <sip:8184443118@192.168.1.100:5060> Content-Length: 0			

Figure 122. IP Header and IP payload data for (2) REGISTER message after NAT traversal

After this operation, the NAT table includes the following information:

NAT TABLE after the REGISTER message passed the NAT				
ID	SRC IP	SRC PORT	DEST IP	DEST PORT
1	192.168.1.100	5060	75.83.202.16	1025

Figure 123. NAT table after REGISTER sent through NAT

We will assume that the REGISTRAR server is not behind a NAT'ing device and responds with a 200 OK. Upon receipt of REGISTER request, the REGISTRAR server extracts the IP address and port number from the top **Via** header, which includes the address of the last hop that sent the message, and use it as a destination for the 200 OK response. Because the Via header was not updated by the NAT, it includes the private IP address **192.168.1.100:5060**, which is not routable.

The structure of the response message is shown below:

SRC IP	SRC PORT	DEST IP	DEST PORT
193.16.148.244	5060	192.168.1.100	5060
200 OK SIP/2.0 Via: SIP/2.0/UDP 193.16.148.244:5060 From: <sip:8184443118@myserviceprovider.com>;tag=3261c4561 To: "Alice" <sip:8184443118@myserviceprovider.com>; Contact: "Alice" <sip:8184443118@192.168.1.100:5060> Content-Length: 0 Expires: 180			

Figure 124. Packet structure for the 200 OK response

The SIP message includes several header fields that include IP and port information related to the delivery of the message:

- The top **Via** header indicates the IP address and port number of the last hop of the SIP message. The responses must follow the path indicated by Via header.
- The **Contact** header indicates the current location of the endpoint. This location can be further used by the remote party to send subsequent messages, such as ACK and BYE, unless a Record Route mechanism is forcing the messages to always pass the PROXY.

Test Case: Using VoIP to Measure NAT/PAT Performance

- The session description protocol (SDP) is used to negotiate how the media session must be handled. The following two fields carry information related to the media IP address and port number that can cause NAT problems:
 - The connection (**c**) field includes IP addresses used to send and received media.
 - The media (**m**) field includes the port number used to send and receive media.

Even assuming a successful establishment of the SIP session, the media path will fail, because the **c** and **m** fields included in the SDP body of the INVITE request sent by Alice's phone will use the private IP address **192.168.1.100** and a port number dynamically selected by the phone for media transmission and reception.

Hence, after the call is established, the incoming media from the public network will be:

SRC IP	SRC PORT	DEST IP	DEST PORT
192.168.1.100	5060	193.16.148.244	5060

```
INVITE sip:remoteserviceprovider.com SIP/2.0
Via: SIP/2.0/UDP 192.168.1.100:5060
From: "Alice" <sip:8184443118@myserviceprovider.com>;tag=4000a5000
To: "Bob" <sip:9195642244@remoteserviceprovider.com>;
Contact: "Alice" <sip:8184443118@192.168.1.100:5060>
Content-Length: 0

v=0
o=Alice 2890844526 2890844526 IN IP4 192.168.1.100
s=
c=IN IP4 192.168.1.100
t=0 0
m=audio 10000 RTP/AVP 0 8 101
a=rtpmap:0 PCMU/8000
```

Figure 125. Sample INVITE message with SDP offer

VoIP/NAT Traversal Issue Solutions

The solutions can be classified as:

- Client-based solutions
- Server-based solutions

Client-based solutions add intelligence to the endpoint that detects the IP address and port number used by the NAT translation, applying translation within the application specific content. Examples of such solutions include:

- Simple Traversal of UDP through NAT (STUN)
- Traversal using Relay NAT (TURN)
- Interactive Connectivity Establishment (ICE)

Server-based solutions assume a public server will resolve the translations specific to the application layer. Solution examples include:

- Application layer gateway
- Session border controllers

STUN – Simple Traversal of UDP Through NAT

Defined by RFC 5389, STUN is a lightweight protocol that allows a wide variety of applications to work through existing NAT infrastructures. Using STUN, applications can discover the presence and the types of NATs and firewalls between them and the public Internet. It can also be used to determine the public IP addresses and ports allocated by NAT. STUN works with many existing NATs, and does not require any special behavior from them. To determine the IP/port mapping information, STUN uses an external STUN server.

STUN is not a NAT traversal solution by itself, but it can be used as a tool in the NAT context. For example, a VoIP client can use STUN to discover whether it is behind a NAT, determine the NAT type, discover the public IP address and port number on the outermost NAT, and then utilize that IP address and port within its protocols.

A system using STUN will include one or more STUN clients located in the private network and one STUN server located in the public network. The following example assumes that a STUN client is located on a SIP endpoint in the private network and uses an external STUN server to discover the IP and port mapping information as well as opening the SIP signaling port: 5060.

The figure below illustrates the first step in the communication between a STUN client and server:

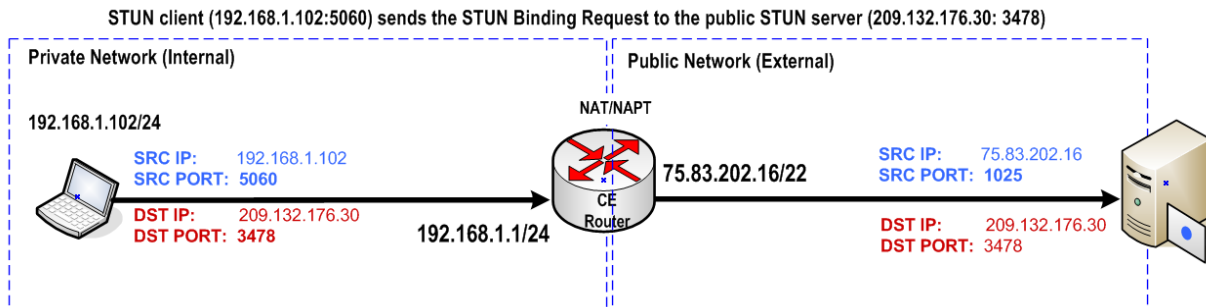


Figure 126. STUN client initiating a Binding Request

- The STUN client initiates a STUN binding request using the source address **192.168.1.102** and port number **5060**; the packet is sent to the external STUN server at **209.132.176.30**, which listens on the standard STUN port **3478**.
- The NAT gateway receives the STUN binding request, translates the **192.168.1.102** address to the public IP address **75.83.202.16** and the private port **5060** to the public port **1025**.
- On receipt of the request, the STUN server issues a STUN binding response from **209.132.176.30:3478** with the destination set to the public IP address/port used by the NAT gateway to initiate the STUN binding request: **75.83.202.16:1025**. The payload of the STUN binding response message includes the MAPPED-ADDRESS field set to the public IP/port address of the NAT gateway: **75.83.202.16**.

Test Case: Using VoIP to Measure NAT/PAT Performance

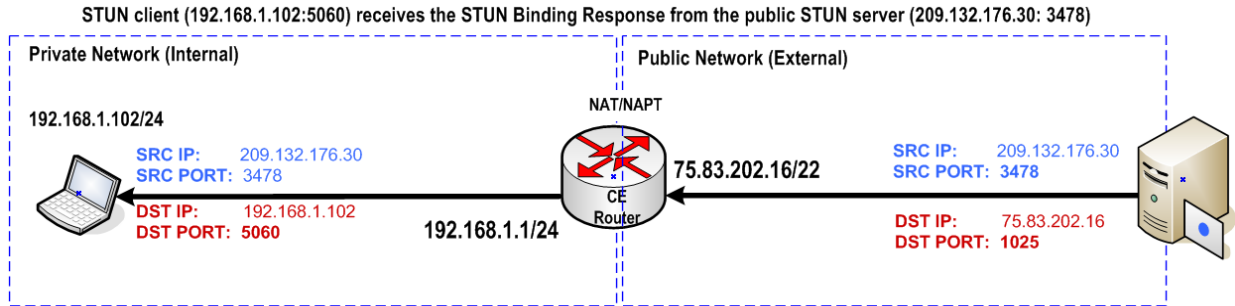


Figure 127. STUN Binding Response traveling through NAT

- The NAT gateway receives the STUN binding response, translating the destination IP address and port number from the IP header back to the address and port used by the STUN client when it initiated the STUN binding Request. Further, the application – in our example the SIP client with STUN support, uses the public IP address and port from the MAPPED-ADDRESS field of the STUN binding response to construct the payload of the SIP requests initiated by the SIP client.

The mapping between the private IP/port address and the public IP/port address is typically deleted after a timeout, when no traffic passes through the NAT. To avoid removal of the NAT entries, the STUN client periodically sends the STUN binding request to refresh the mapping on the NAT gateway.

Symmetric NAT prevents STUN from working, due to the rules it uses to create the pinhole. For non-symmetric NAT, the same pinhole will be used whenever the same endpoint from the private network sends a packet to a different destination. With symmetric NAT, a new pinhole will always be created each time the destination of the packet is changed, regardless if the source remains the same.

With STUN, the client sends the binding request packet, and the STUN server replies with the public location of the pinhole (MAPPED-ADDRESS) that was used to send the binding request, so an application can use this information in its payload.

The SIP client will further use the extracted MAPPED-ADDRESS to build the payload of the SIP requests due to the different destination of the SIP packets, as the SIP server has a different address than the STUN server, a new pinhole will be created by a symmetric NAT, leading to the use of an invalid MAPPED-ADDRESS information into the payload of the SIP request.

TURN – Traversal Using Relay NAT

TURN is a simple protocol that tries to resolve STUN's issue with symmetric NATs by allocating a public IP/port on a public server that is used to relay media between communicating parties.

While TURN can almost always provide connectivity to a client, it has the disadvantage of introducing a high cost to the service provider – hosting the TURN server.

ICE – Interactive Connectivity Establishment

ICE describes a methodology for network address translator traversal for the session initiation protocol (SIP). ICE is a framework that combines existing protocols such as STUN, TURN, and real specific IP (RSIP) by choosing the best interconnection method. ICE provides a NAT traversal solution that is independent from the types or number of NATs in use.

The advantages provided using ICE are as follows:

- It provides the most economical solution for a service provider
- It always results in the minimum voice latency
- It can be done without any increase in the call setup delay
- It facilitates the transition of the Internet from IPv4 to IPv6, supporting calls between dual IPv4/IPv6 stack clients and IPv6 clients behind an IPv4 to IPv6 NAT device.
- It uses SIP preconditions to guarantee that the phone does not ring unless the users will both hear and see each other when they pick up.

The disadvantages are:

- ICE is not yet standardized
- ICE is designed only to work with SIP

Application Layer Gateway (ALG)

The ALG is a common algorithm implemented in firewalls and NATs that helps devices be aware of information available at the application-layer, that is, within the IP payload data. Being aware of the application-layer, the ALG inspects packets for embedded IP address and port information and can perform NAT without breaking applications such as FTP, RTSP, ICMP, SIP, H323, SKINNY, and MGCP.

Compared with regular NAT, ALG allows a pinhole to be dynamically-created to permit the data exchange for the session, and correlates control and data sessions. ALG can use the same common timeout value or they can keep them independent.

Firewalls policies can be configured to trigger the ALG module based on either application or by service type.

ALGs solution for NAT traversal is similar with the solution based on SBCs, explained in the next section.

Session Border Controllers

From the NAT point of view, the SBC performs the role of an ALG, translating the addresses and ports in the application data between private and public addressing schemes.

An SBC can be divided in two logical components:

- A Session Border Element that handles the signaling part of a VoIP call and

Test Case: Using VoIP to Measure NAT/PAT Performance

- A Data Border Element that handles the media traffic associated with a call

Those components can be collocated or it can be distributed.

From the deployment point of view, the SBCs can be seen in five common scenarios:

- At the border between two service providers
- At the border between a service provider and their customers
- As a central media transcoder
- Between different VPNs provided by a service provider
- As a network resource controller

By rewriting the IP addresses and ports included in the signaling headers, the SBC allows VoIP traffic to be transmitted and received from a device behind a firewall/NAT without requiring the customer's firewall/NAT to perform that function.

Registration

To place and receive calls, Alice's phone must register with the registrar server located in the service provider's network. The service provider placed a session border controller at the edge of his network to address the NAT traversal issue. His enterprise customer is not required to upgrade his NAT/Firewall device at the edge of the enterprise network.

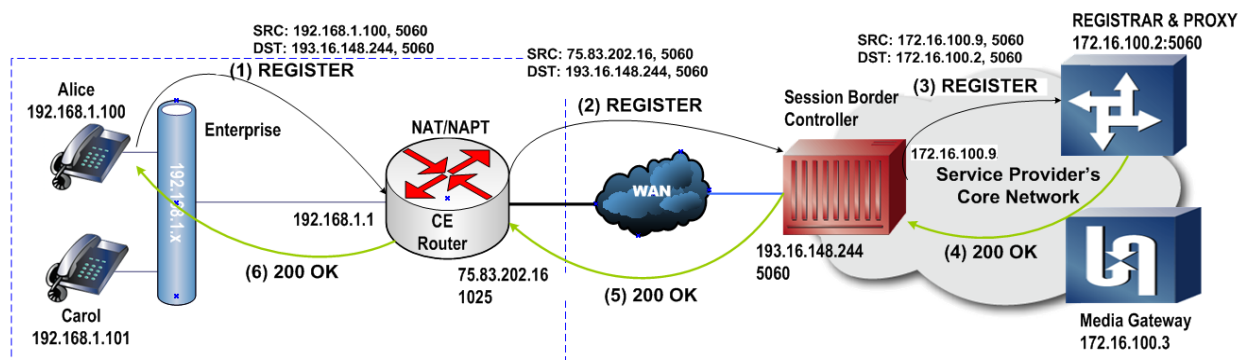


Figure 128. How a Session Border Controller resolves the NAT traversal issue for SIP messages

When Alice's phone sends the REGISTER request, it sends it to the public IP address of the SBC, **193.16.148.244** port **5060**. This address is provided by the service provider to configure the registrar server and outbound proxy server address on the IP Phone. The REGISTER request is transmitted using the private address **192.168.1.100** from port **5060**.

The message (1) REGISTER has the following structure when issued by Alice's phone:

Test Case: Using VoIP to Measure NAT/PAT Performance

SRC IP	SRC PORT	DEST IP	DEST PORT
192.168.1.100	5060	193.16.148.244	5060

```
REGISTER sip:myserviceprovider.com SIP/2.0
Via: SIP/2.0/UDP 192.168.1.100:5060
From: "Alice"
<sip:8184443118@myserviceprovider.com>;tag=3261c4561
To: "Alice" <sip:8184443118@myserviceprovider.com>;
Contact: "Alice" <sip:8184443118@192.168.1.100:5060>
```

Figure 129. Message "(1) REGISTER" as sent by Alice's phone

The REGISTER message will be modified first by the NAT gateway placed at the customer edge. This modification will only rewrite the source IP/port number in the IP header; no modifications are made in the destination IP/port or inside the SIP message itself. The NAT gateway issues the REGISTER message with the following structure:

SRC IP	SRC PORT	DEST IP	DEST PORT
75.83.202.16	1025	193.16.148.244	5060

```
REGISTER sip:myserviceprovider.com SIP/2.0
Via: SIP/2.0/UDP 192.168.1.100:5060
From: "Alice"
<sip:8184443118@myserviceprovider.com>;tag=3261c4561
To: "Alice" <sip:8184443118@myserviceprovider.com>;
Contact: "Alice" <sip:8184443118@192.168.1.100:5060>
Content-Length: 0
```

Figure 130. The structure of message "(2) REGISTER" after passing the NAT gateway

The CE-NAT gateway updates its NAT table with the following entry:

NAT TABLE after the REGISTER message passed the NAT				
ID	SRC IP	SRC PORT	DEST IP	DEST PORT
1	192.168.1.100	5060	75.83.202.16	1025

Figure 131. NAT Table on the CE gateway after REGISTER request is forwarded

The REGISTER request next reaches the public interface of the SBC, which receives the message on **193.16.148.244**, port **5060**. The SBC acts as a registrar server for Alice's phone, answering the REGISTER request with **200 OK**. Before transmitting the **200 OK** response back to Alice's phone, it acts as a SIP user agent, or endpoint, for the registrar server located in the Service provider's network. Hence, the SBC separates the communication in two distinct SIP dialogs – one with the phone, and one with the registrar server. We will refer to this role as a back-2-back user agent (B2BUA).

To address the NAT traversal issue, the SBC detects the presence of a basic NAT by comparing the IP address and port number set in the Via headers with the source IP and port number set in the IP header. Instead of responding with **200 OK** to the address set in the Via header, it sends the response to **75.83.202.16**, port **1025**, which represents the public address of the CE-NAT gateway that issued the REGISTER request.

The SBC also adds the Expire header in the response, which controls how often the endpoint will re-register. The original IP and port will be re-used inside the payload.

Test Case: Using VoIP to Measure NAT/PAT Performance

Because the SBC needs to register as an endpoint with the registrar server using the private interface that connects the SBC to the registrar server, it also uses a NAT translation that maps the public address (**192.16.148.244, 5060**) to its private address (**172.16.100.9, 5060**). The NAT translation also includes the Expire value, which will be updated upon receipt of 200 OK response from the server.

The following figure illustrates the path of the **200 OK** response from registrar server to an IP phone, via an SBC and CE-NAT gateway.

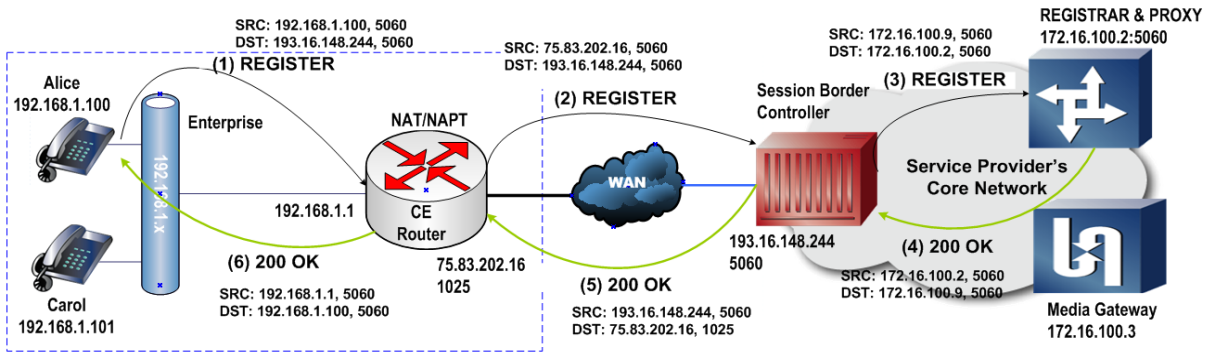


Figure 132. Path of the 200 OK response initiated by SBC

When the **(3) REGISTER** request is sent by the SBC to the registrar server, it will have different Call-IDs and transaction information compared to the initial request.

SRC IP	SRC PORT	DEST IP	DEST PORT
172.16.100.9	5060	172.16.100.2	5060
<pre>REGISTER sip:myserviceprovider.com SIP/2.0 Via: SIP/2.0/UDP 172.16.100.9:5060 From: "Alice" <sip:8184443118@myserviceprovider.com>;tag=5000c5000 To: "Alice" <sip:8184443118@myserviceprovider.com>; Contact: "Alice" <sip:8184443118@172.16.100.9:5060> Content-Length: 0</pre>			

Figure 133. Structure of message "(3) REGISTER"

Assuming no NAT device is between SBC and registrar server, the response will look as shown below:

SRC IP	SRC PORT	DEST IP	DEST PORT
172.16.100.2	5060	172.16.100.9	5060
<pre>200 OK SIP/2.0 Via: SIP/2.0/UDP 172.16.100.9:5060 From: <sip:8184443118@myserviceprovider.com>;tag=5000c5000 To: "Alice" <sip:8184443118@myserviceprovider.com>; tag=6000c6000 Contact: "Alice" <sip:8184443118@172.16.100.9:5060> Content-Length: 0 Expires: 1800</pre>			

Figure 134. Structure of the message "(4) 200 OK" received by SBC

As previously mentioned, when the SBC forwards the **200 OK** response back to the IP phone it sends the response to the public address of the CE-NAT gateway (**75.83.202.16, 1025**). It uses

Test Case: Using VoIP to Measure NAT/PAT Performance

the Call-ID and transaction information received in the initial REGISTER request. The response will use the private IP address of the IP phone.

The structure of the **200 OK** response is shown below:

SRC IP	SRC PORT	DEST IP	DEST PORT
193.16.148.244	5060	75.83.202.16	1025

```
200 OK SIP/2.0
Via: SIP/2.0/UDP 193.16.148.244:5060
From: <sip:8184443118@myserviceprovider.com>;tag=3261c4561
To: "Alice" <sip:8184443118@myserviceprovider.com>;
Contact: "Alice" <sip:8184443118@192.168.1.100:5060>
Content-Length: 0
Expires: 180
```

Figure 135. Structure of message "(5) 200 OK"

On receipt of the response, the CE-NAT gateway looks up the destination address in its translation table and applies the reverse translation to the IP header. In this way, the destination IP address and port are updated with the IP address and port of the IP phone.

SRC IP	SRC PORT	DEST IP	DEST PORT
193.16.148.244	5060	192.168.1.100	5060

```
200 OK SIP/2.0
Via: SIP/2.0/UDP 193.16.148.244:5060
From: <sip:8184443118@myserviceprovider.com>;tag=3261c4561
To: "Alice" <sip:8184443118@myserviceprovider.com>;
Contact: "Alice" <sip:8184443118@192.168.1.100:5060>
Content-Length: 0
Expires: 180
```

Figure 136. Structure of message "(6) 200 OK"

The IP Phone receives the **200 OK** response and the registration is completed.

SBC's solution for permitting external calls

To allow incoming calls, the SBC must maintain an open pinhole through the CE-NAT gateway. This can be achieved using different methods, such as forcing the IP phone to re-register at shorter intervals, for example, 30 seconds, or by sending a periodic OPTIONS requests from the SBC to IP phone. The pinhole will be kept open regardless of the response type from the IP Phone (**200 OK** versus **4xx** response). A better approach would use TCP as the transport protocol.

All external calls are received from the SBC. Hence, even if the edge of the service provider has a firewall installed, the security policy can be updated to allow any SIP message received from the IP address of the firewall, which is well known, because the service provider shares this information with their customers.

SBC's solution for non-routable media flows

To address this issue, the signaling border element (SBE) component of the SBC implements the SIP B2BUA functionality for the signaling protocol, which allows the call between the calling and called parties to be separated in two distinct conversations: one between the caller and the SBC and one between SBC and called party.

After the media is established, the data border element (DBE) component of the SBC acts as an RTP proxy between the caller and called party.

The following figure illustrates the path of the **INVITE/SDP** request traveling from Alice's phone to the proxy server and the path of the **200 OK/SDP** response. The IP addresses and port numbers for the signaling messages are highlighted at the top. The media IP addresses and media port numbers are highlighted at the bottom. The SDP portion used to describe the media IP address and port number used in the **INVITE** and **200 OK** response messages is also highlighted at the bottom.

The SDP parameters used to inform the remote party of the IP address and port number that are used to generate and receive media are specified under the **c** line, which contains the media IP and the **m** line, which contains the media port number.

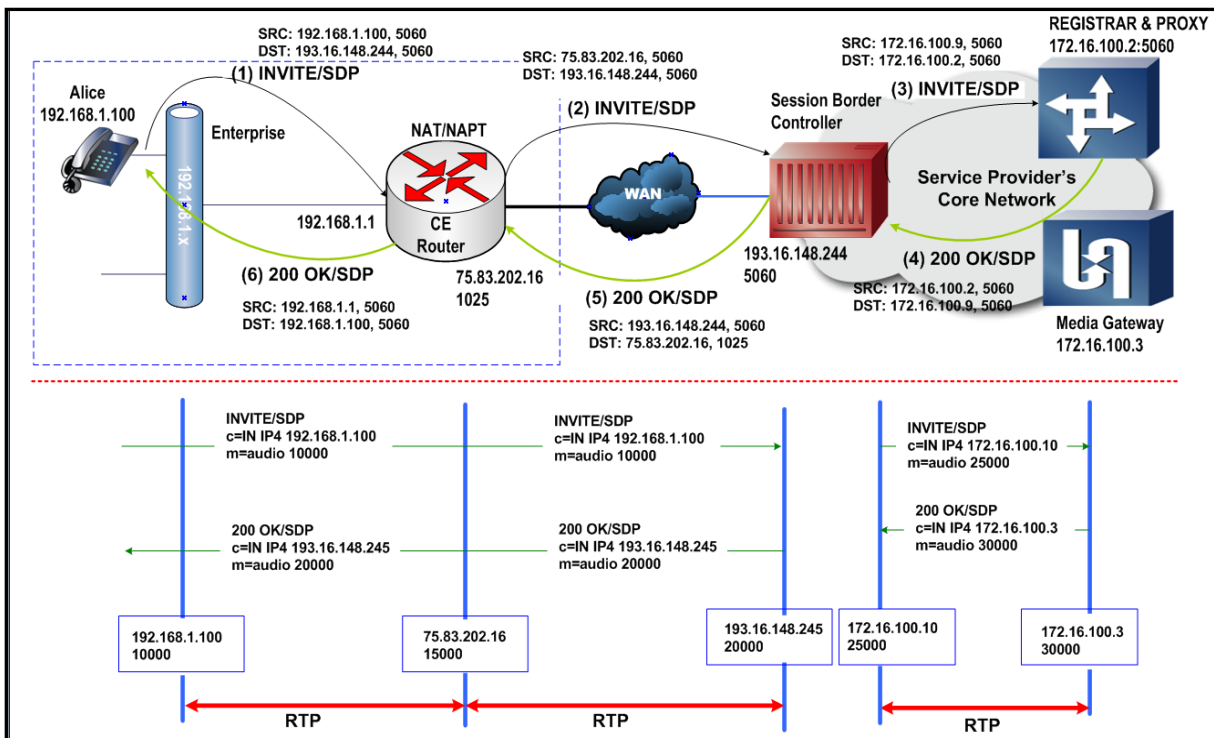


Figure 137. How an SBC resolves the media traversal problem

Test Case: Using VoIP to Measure NAT/PAT Performance

Remember that Alice's phone must be configured to use the public IP address of the SBE-SBC (**193.16.148.244:5060**) as its outbound proxy. In this way, Alice's phone will send all the requests via the public interface of the SBC.

When the **(1) INVITE** request is sent by Alice, the SBC detects the presence of a NAT device between them by comparing the source IP address and port number of the IP header with the IP address and port number included into the **SIP INVITE** message (**192.168.1.100:5060**). Hence, the SBC will return the **200 OK/SDP** back to the public address and port used by the CE-NAT to send **(2) INVITE (75.83.202.16:1025)** rather than using the private IP address and port number (**192.168.1.100:5060**) used in the **Via** header of the **(2) INVITE** message. Upon receipt of the **(2) INVITE**, the SBE-SBC component initiates a new call to the SIP proxy server located in the service provider network, which connects the call to a PSTN network, not displayed in the figure. It initiates a new call by sending **(3) INVITE/SDP**, which establishes a separate call with the proxy server. In this message, the IP header, the SIP headers, and the SDP body will use the signaling IP/port address **172.16.100.9/5060**, while the following SDP lines are used for the media IP address and port:

```
c = IN IP4 172.16.100.10
m = 25000 audio RTP/AVP 0 18 101
```

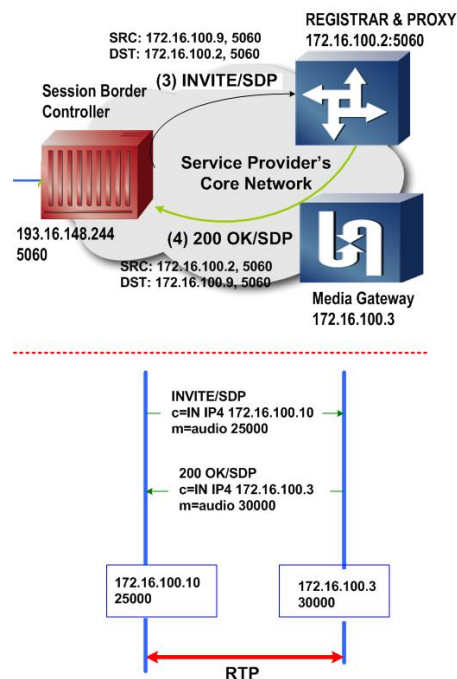


Figure 138. Establishment of media between an SBC and Media Gateway

Test Case: Using VoIP to Measure NAT/PAT Performance

The proxy server replies with **200 OK/SDP** using the address and port (**172.16.100.2/5060**) from inside the IP and SIP headers. For media, the proxy server uses the media IP address and port number exposed by the SBC-DBE media gateway: **172.16.100.3, 30000**. Hence, the SDP body attached to **(4) 200 OK/SDP** response includes:

```
c = IN IP4 172.16.100.3
m = 30,000 audio RTP/AVP 0 101
```

This allows the SBC-DBE component to generate and receive RTP messages between **172.16.100.10: 25000** and **172.16.100.3: 30000**.

On the public side of the call, as seen from SBC's point of view, upon receipt of the **(4) 200 OK/SDP** response, the SBE-SBC component sends the **(5) 200 OK/SDP** response, which includes the public media address of the DBE-SBC component: **193.16.148.245: 20000**. The message is sent to the public address of the CE-NAT: **75.83.202.16: 1025**. The SDP body of the response includes the following media lines:

```
c = IN IP4 193.16.148.245
m = 20,000 audio RTP/AVP 0 18 101
```

Upon receipt of the **(5) 200 OK/SDP** response, the CE-NAT will initiate the reverse translation from its public address **75.83.202.16: 1025** to the private address **192.168.1.100: 5060**, and the message will be sent toward Alice's phone without modifying the payload. The CE-NAT only swaps the destination IP address and port fields of the IP header. This message is highlighted as message **(6) 200 OK/SDP** in the figure below.

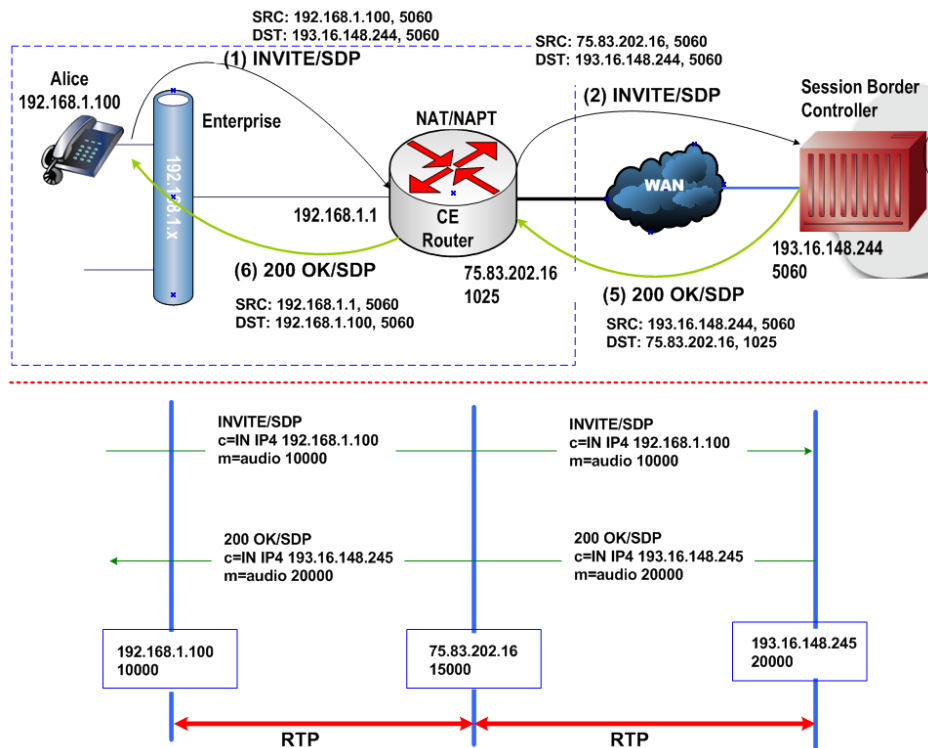


Figure 139. Media establishment between SBC-DBE public interface and Alice's phone

Test Case: Using VoIP to Measure NAT/PAT Performance

Finally, Alice's phone processes the **(6) 200 OK/SDP** message, using the IP address and port numbers received in the **c** and **m** lines of the SDP as a destination for the media packets. After the ACK response is sent back to the SBC, the conversation between Alice's phone and the external PSTN phone will be anchored by the DBE-SBC component. The first RTP packet generated by Alice's phone will open a new pinhole on the CE-NAT gateway (**75.83.202.16:15000**).

Because the DBE-SBC component stays in the middle of the media flow, acting as the actual destination for the RTP packets initiated by Alice's phone, it can provide both source and destination IP: port information. It uses this information to redirect the media packets from SBC to the public address of the CE-NAT gateway **75.83.202.16:15000**, rather than sending the RTP packets directly to the private IP address and port number received in the **(2) INVITE/SDP** message: **192.168.1.100: 10000**. Hence, when the DBE-SBC sends the RTP packets to **75.83.202.16:15000**, the NAT will rewrite the destination IP address and port numbers to **192.168.1.100: 10000** and the RTP packets will be successfully received by Alice's phone.

Objective

This test determines the maximum number of VoIP video calls that can be established through a DUT when NAT is enabled. The test methodology, Determining the Max Call Setup Rate for SIP-Based Devices and Systems, uses voice traffic to determine the maximum number of concurrent calls that can be established through a device, with or without NAT enabled. VoIP video traffic and video conferencing support requires IxLoad 5.00 or later version.

In this test methodology, the configuration is extended to generate voice and video traffic. The video traffic adds more complexity to the NAT traversal because a video connection requires four media ports (RTP audio, RTCP audio, RTP video, and RTCP video) compared with only two for voice traffic (RTP/RTCP audio).

The test methodology, Test Case: Determining the Maximum Call Setup Rate (CPS) uses a SIP configuration example with a CPS objective. In this test we will reuse the same configuration and we will modify the test objective to allow us to sustain a number of concurrent calls where voice and video traffic are both enabled.

The configuration changes required to enable video traffic are common for SIP and H323 traffic.

Setup

In this test topology an Ixia port emulates a private network consisting of VoIP clients that establish calls through a NAT enabled device (e.g.: ALG or SBC) with the signaling server and media gateway located in the public network.

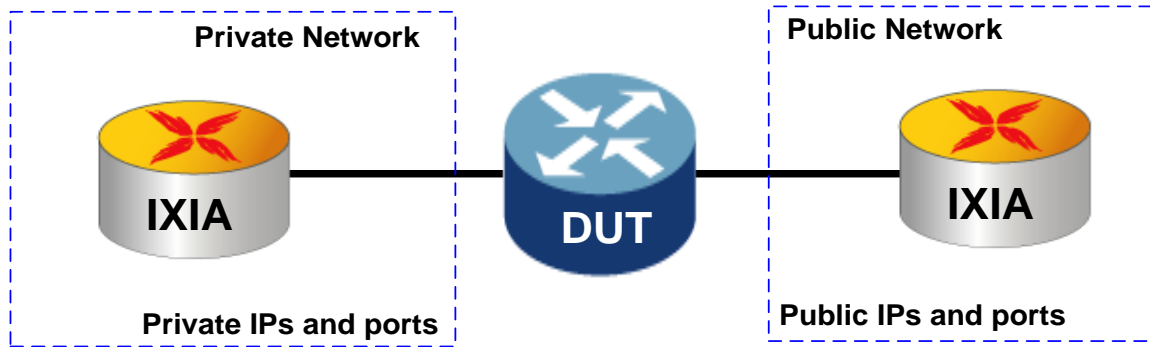


Figure 140. Test Topology

Step-by-Step Instructions

The final IxLoad configuration as a result of these steps is provided on blackbook.ixiacom.com Web site - see *IxLoad 5.10 Voice - SIP NAT.crf*. To import a Compressed Repository File (crf) in IxLoad, use the command **Import** under the **File** menu.

The step-by-step instructions highlight how to set the essential parameters of this configuration and explain additional options, which may be used to change the behavior of the test.

Depending on the protocol used (SIP or H.323) start with the configuration steps highlighted in *Determining the maximum call setup rates* (SIP example) and *Determining the maximum number of concurrent calls* (H.323 example). This example assumes the configuration instructions starting with the SIP configuration, because the test objective will require modifications as well. This example starts from the configuration "IxLoad Voice - SIP Call Setup Rate.rxf" resulted at the end of the **Test Case: Determining the Max Call Setup Rate for SIP-Based Devices and Systems**.

Test Case: Using VoIP to Measure NAT/PAT Performance

Updating the Test Scenario to Generate Voice and Video Traffic

1. Open the configuration **IxLoad Voice - SIP Call Setup Rate.rxf**.
2. Click the **MakeCall** activity, and then click **Save As** to save the test scenario under the name **SIP_NAT_MakeCall - ReceiveCall - EndCall with RTP.tst**.
3. Save the configuration (main menu/File/Save As) under the name **IxLoad Voice - SIP NAT.rxf**.
4. Select either one of the two activities **Make_Call** or **Receive_Call**.

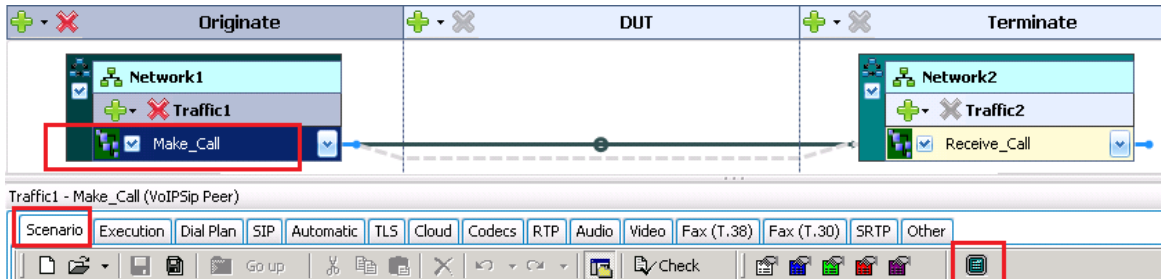


Figure 141. Test Configuration – selection of Scenario Editor

5. The configuration page displays **Test Scenario Editor**.
6. Click **Full Screen** available in the Scenario Editor toolbar.
7. For both scenario channel #0 and scenario channel #1, replace the **Voice Session** script object with the **Multimedia Session** script object; the **Multimedia Session** script object allows both voice and video traffic to be simultaneously sent and received. Leave the parameters of the **Multimedia Session** objects to their defaults; the settings at the Activity level, in Audio and Video tabs, will be used.

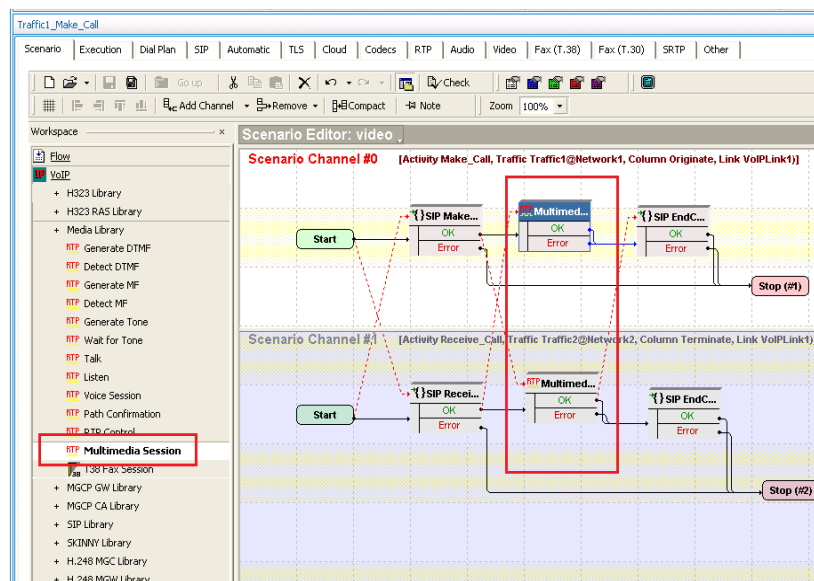


Figure 142. Test Scenario Editor with Multimedia Session

Test Case: Using VoIP to Measure NAT/PAT Performance

8. Select the Multimedia Session from **Scenario Channel #0** and open its properties (**Right Click** on object| **Object Properties ...**) to configure the video clip and audio clip used for multimedia transmission.
 - a. Select the **Play Audio, Clip** and the **Play Video, Video** to be transmitted; all video properties will be retrieved from the selected video clip (for example, CODEC bit rate, profile, level).

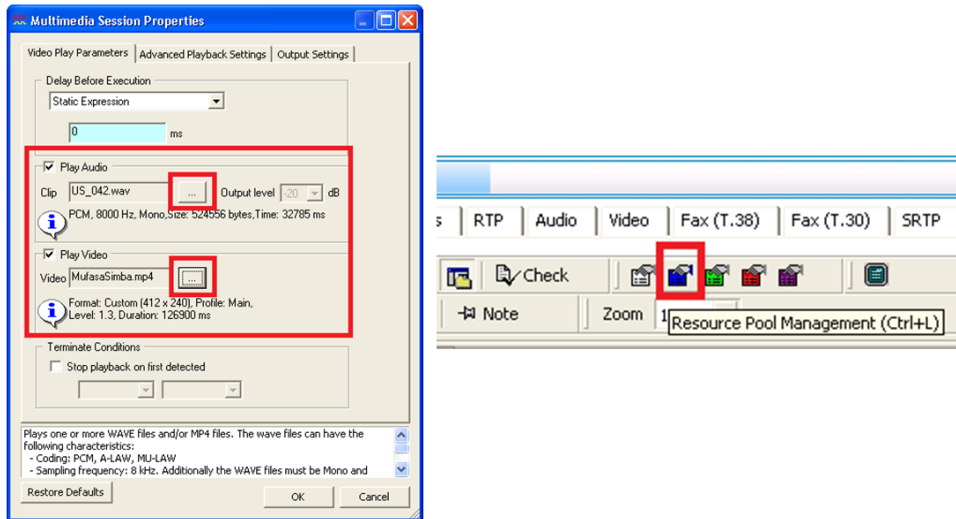


Figure 143. Multimedia Session (left) and Resource Pool Management button (right)

- b. New audio and video sample files can be added by clicking **Resource Pool Management** available in the Scenario Editor toolbar.

Configuring the Settings for Video Traffic

1. Click the **Make_Call** activity.
2. Click the **Video** configuration tab.
3. Enable the video traffic by selecting the **Enable video on this activity** check box.
4. Select the Clip to be played from the list.
5. Set the duration of the call to **3 minutes** using the **Play for** parameter.

Test Case: Using VoIP to Measure NAT/PAT Performance

- To enable calculation of video MOS (V-MOS relative, V-MOS absolute, and V-Factor) select the **Perform MOS** check box to also automatically select the **Calculate One Way Delay** check box.

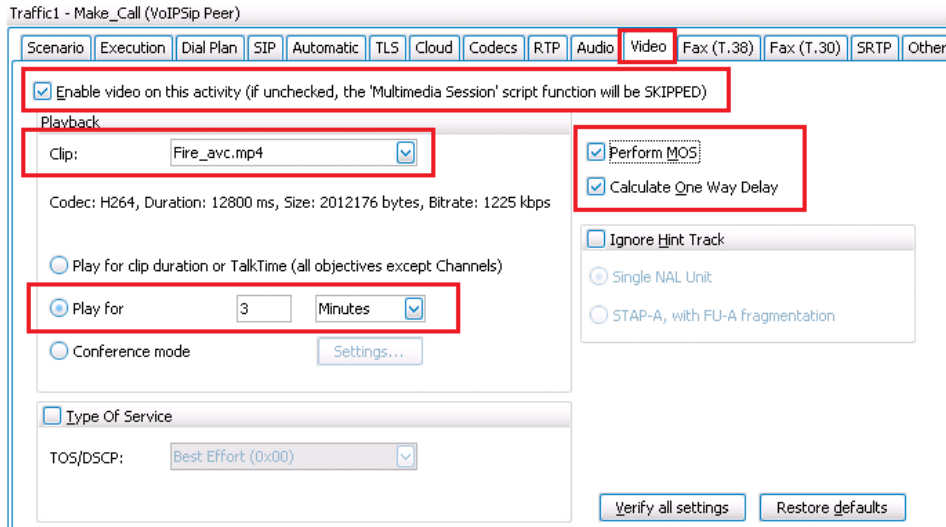


Figure 144. Video settings

Note: if the test objective will remain 'Calls Initiated per Second,' at step 3, select the **Play for Talk Time** option to allow the test objective to calculate and pass the call duration to the script.

- Select the second activity, **Receive_Call**, and repeat steps 2 to 6.
- Save the repository file by using the **File | Save (CTRL+S)** command.

Configuring the Timeline and Objective

- Click **Timeline & Objective** from the **Test Configuration** panel.
- Set the test **Objective Type** to **Channels**.
- Set the test **Objective Value** to the number of concurrent calls desired.
- Under **Timeline**, set the **Ramp Up Value**.
- Set the **Ramp Up Interval** to *1 second*.
- Set the **Sustain Time** to *cover the test duration desired*.

Note: The **Sustain Time** must be higher than the **Talk Time**. It is also good practice to add some extra time (for example, 30 seconds) as a buffer for any delays that may occur during the call setup and end call phases. Hence sustain time should have a value higher than 3 min + 30 sec = 3.30 min

Test Variables

Test Tool Variables

Table 43.

Parameter Name	Current Value	Additional Options
Type of traffic	Video	Audio, Video, T38, Audio/Video RTCP
IP Mapping rules for signaling and media	N to 1	1 to 1, 1 to N, N to 1, and N to N
Number of users available	User defined	
Number of active calls	User defined	
Call Rate	User defined	
Call Duration	User defined	
Audio/Video CODEC (type, packet size & frequency)	G.711u H264	G.711, G.729, G.723, G.726, iLBC, AMR
Mix with data protocols (for example, FTP, HTTP, Telnet)	Not included	Any combination of data protocols supported by IxLoad
VoIP Protocol used	SIP	SIP, H323, SKINNY, MGCP, H248
Mix of call flows		successfully calls, canceled calls, unanswered calls, busy calls
Mix of call features		call forward, call transferred, call hold/retrieve

DUT Test Variables

Table 44.

Parameter Name	Current Value	Additional Options
NAT type, translation type	User defined	
Number of public IP addresses and port numbers	User defined	-
Number of private IP addresses and port numbers	User defined	

Results Analysis

The DUT shall be monitored for:

- Memory size, memory allocation/de-allocation issues while:
 - Translations are added/deleted to/from the NAT table
 - Translations are continuously added and sessions are kept active

Test Case: Using VoIP to Measure NAT/PAT Performance

- CPU usage
- Size of NAT table

The following questions provide guidelines on how to recognize specific problems during or at the end of the test execution:

1. Has the test objective been achieved? Check the **Call Rates** view.

Table 45. Call Rate statistics

Statistic Name	Value	Questions
Calls Attempted per Second		1. Have the calls been attempted continuously at a constant call rate during the Sustain Time?
Calls Connected per Second		2. How do the Calls Attempted rate and the Calls Connected rate compare to each other?

2. Have any call failures been reported? Check the **Calls** view.

Table 46. Call statistics

Statistic Name	Value	Questions
Calls Attempted		1. Have any call attempts failed? Compare: a. Calls Attempted and Calls Received, with b. Calls Attempted and Calls Connected.
Calls Connected		
Calls Received		
Calls Answered		
End Calls Received		
End Calls Completed		

3. Have any scenario loop failures been reported? Check the **Loops** statistics view.

Table 47. Statistics highlighting the pass/fail result based on call flow execution

Statistic Name	Value	Questions
Total Loops		1. Are the Successful Loops and Total Loops values equal?
Successful Loops		2. Have any Failed Loops, Aborted Loops or Warning Loops been reported? Note: failed/aborted and warning loops highlights failures at the scenario level.
Failed Loops		
Aborted Loops		
Warning Loops		

Test Case: Using VoIP to Measure NAT/PAT Performance

4. Has the QoS for signaling met the expected quality? Check the **Call Times** and **Delays** statistic views. Use the maximum value reported.

Table 48. : Statistics used to determine the QoS for the SIP signaling

Statistic Name	Value (max /avg/min)	Questions
Call Setup Time		<ol style="list-style-type: none"> 1. Is the maximum Call Setup Time less than 4 seconds? 2. Is the maximum End Call Time less than 2 seconds? 3. Is the maximum Media Delay (Tx or Rx) less than 4 seconds? 4. Is the maximum Post Dial Delay less than 2 seconds? 5. Is the maximum Post Pickup Delay less than 2 seconds? 6. For all the stats listed in this table, compare their value distribution in time.
End Call Time		
Talk Time		
Media Delay TX/RX		
Post Dial Delay		
Post Pickup Delay		
<p>Note: Another important factor in establishing the quality of the signaling is the number of retransmissions. IxLoad counts those using the <i>SIP Retransmitted Msgs</i> statistic, located under the SIP Messages view.</p>		

5. Has the QoS for media met the expected quality? Check the **RTP MOS RTP QoS**, **RTP Advanced QoS**, **RTP Jitter Distribution**, **RTP Consecutive Lost Datagram Distribution**, and **RTP Streams** statistic views.

Table 49. MOS statistics

Statistic Name	Questions
RTP MOS Best RTP MOS Worst	<ol style="list-style-type: none"> 1. How do the last values reported by the RTP MOS Best and RTP MOS Worst compare with each other? 2. How does the RTP MOS Worst score compare with the max theoretical score for the CODEC used?
RTP MOS Instant (Best/Avg/Worst)	<ol style="list-style-type: none"> 1. Are any times without an instantaneous MOS value? 2. How frequent are the changes in the instantaneous MOS values?
RTP MOS Per Call (Best/Avg/Worst)	<ol style="list-style-type: none"> 1. How do the MOS per Call statistics compare with the RTP MOS Best and RTP MOS Worst statistics?

Test Case: Using VoIP to Measure NAT/PAT Performance

Table 50. Basic RTP QoS statistics (see RTP QoS and RTP Advanced QoS statistics views)

Statistic Name	Questions
RTP Packets Sent RTP Packets Received RTP Packets Lost	1. Are there any differences between RTP Packets Sent and RTP Packets Received? 2. Does the difference match the value of RTP Lost Packets?
RTP One Way Delay [us]	1. Is the One Way Delay higher than 100 ms?
RTP Delay Variation Jitter [us] RTP Interarrival Jitter [us]	1. What is the max Delay Variation Jitter? 2. What is the max Interarrival Jitter?

Table 51. RTP Jitter distribution statistics

Statistic Name	Value(s)	Questions
Packets with Delay Variation Jitter up to 1 ms		1. Assuming Jitter was reported, what is the distribution of the Delay Variation Jitter values?
Packets with Delay Variation Jitter up to 3 ms		
Packets with Delay Variation Jitter up to 5 ms		
Packets with Delay Variation Jitter up to 10 ms		
Packets with Delay Variation Jitter up to 20 ms		
Packets with Delay Variation Jitter up to 40 ms		
Packets with Delay Variation Jitter over 40 ms		

Table 52. Distribution of RTP Consecutive Lost Packets

Statistic Name	Value(s)	Questions
Consecutive Loss of One Packet Sequence		1. Assuming that packet loss was reported, what is the distribution of the lost RTP packets?
Consecutive Loss of Two or Three Packet Sequences		
Consecutive Loss of Four or Five Packet Sequences		
Consecutive Loss of Six to Ten Packet Sequences		
Consecutive Loss of Eleven or More Packet Sequence		

Test Case: Using VoIP to Measure NAT/PAT Performance

Table 53. RTP Streams

Statistic Name	Value(s)	Questions
Concurrent RTP Streams		1. Assuming that packet loss was reported, what is the distribution of the lost RTP Packets?
Concurrent RTP Streams (max)		
Number of calls with incoming RTP packets		2. Are any calls without RTP? 3. Are any calls with RTP?
Number of calls without incoming RTP packets		4. Does this number match the number of Calls Connected * 2?

Troubleshooting and Diagnostics

The following table summarizes some of the common issues that may be encountered when running a call rate test.

Table 54.

Issue	Troubleshooting Solution
The <i>Incoming RTP throughput</i> is not constant during the Sustain Time or has lower values than the <i>Outgoing RTP Throughput</i>	Check the reported <i>RTP Packets Lost</i> , <i>RTP Consecutive Packets Lost</i> , and compare the RTP Packets Sent with RTP Packets Received. This issue may be caused by a large number of packets being dropped or by calls without RTP packets. Endpoints connected in calls without receiving any media are counted using the <i>Calls without RTP packets</i> statistic.
All calls were connected but the expected maximum throughput is not reached	Verify that requests to disconnect the calls are not received from the DUT during the <i>Talk Time</i> .
The <i>incoming throughput</i> and <i>outgoing throughput</i> values are lower or higher than expected	Verify the <i>Codec Distribution</i> and <i>RTP Packet Distribution</i> statistics – these may be used to confirm that the calls negotiated the desired audio/video CODEC. Problems may be fixed by correcting the DUT configuration or by setting the same CODEC on both calling and receiving activities.

Conclusions

This test methodology provided a guideline for determining the maximum number of voice or video calls that can be established through a device with NAT functionality enabled by emulating clients in the private network and the server in the public network. The mapping of IP addresses and port numbers used in the private network, correlated with the traffic type, traffic volume (number of concurrent streams) or combination of voice and data traffic can lead to QoS degradation or the absence of media.

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

Overview

Even though VoIP is largely adopted, there are still a lot of telephony systems in production that use legacy technology. The bridge between VoIP and PSTN is done by dedicated gateways that convert the signaling and media between the two environments. There is a large variety of gateways depending on type of signaling protocols used in both VoIP and PSTN networks.

The VoIP to PSTN gateway does the physical and protocol conversion of both signaling and media traffic. The basic SIP to ISDN call flow is as follows:

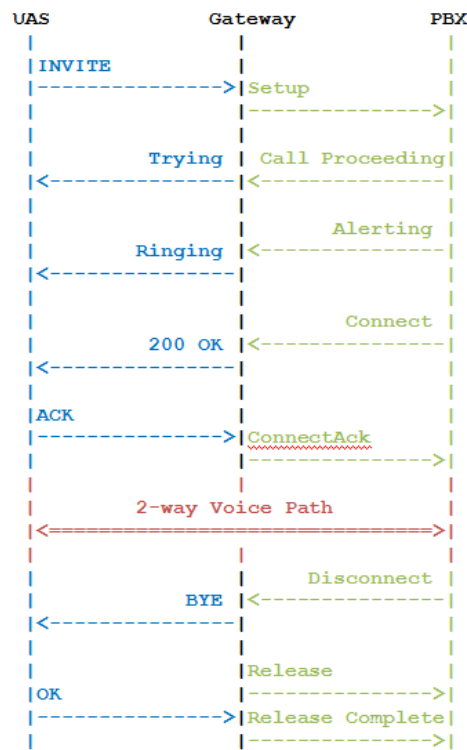


Figure 145. Basic SIP to ISDN call flow

Objective

The goal of this test methodology is to determine the capacity represented in the number of concurrent calls that can be handle by a VOIP to PSTN gateway.

IxLoad simulates SIP User Agents on Ethernet interface and PSTN PBX on TDM interface. The detailed steps below will show how to configure a test in IxLoad to originate calls from SIP and terminate on E1/PRI to test a getaway. **E1** is a format of Time Division Multiplex (**TDM**) technique providing 2.048 Mbps communication link divided into 32 time slots of 64 kbps each. E1 is used in most parts of the world while a different format, T1 (1.544 Mbps divided in 24 time

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

slots) is used in North America. The 24 time slots of a T1 link, or 32 of the **E1** can be used to transfer signaling, data or both of them. Channel Associated Signaling (CAS) and Integrated Services Digital Network (ISDN) are two of the standards defining how the signaling and media is sent over the TDM links.

Setup

The example below shows the step to originate 30 concurrent calls from SIP and terminate them on an E1 span set to ISDN; media will be sent in both directions on the established calls.

The example uses one port of an Application Load Module (for example, Acceleron XP) and one E1/T1 port of one of the Telephony Boards (944-0002 TOB-E1/T1-02, 944-0003 TOB-E1/T1-04 or 944-0004 TOB-E1/T1-08).

The test setup topology is shown in the next figure:

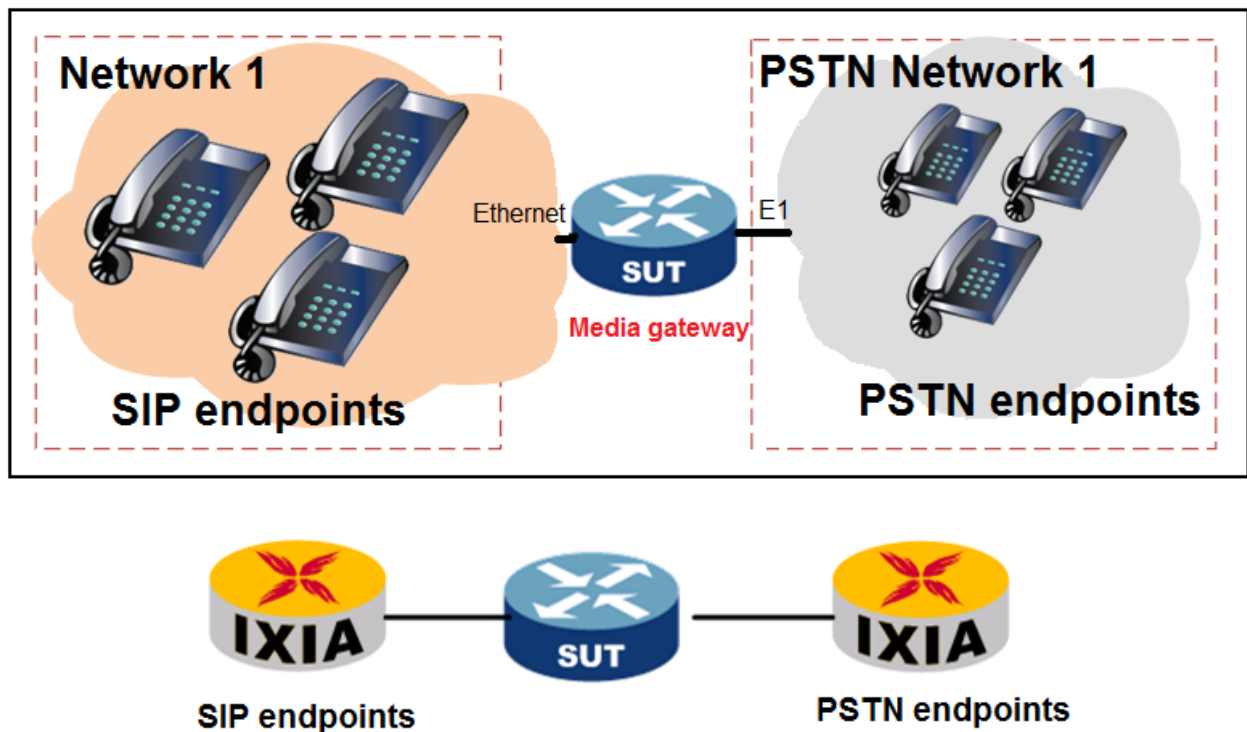


Figure 146. SIP to PSTN test setup

Note: This configuration requires a real gateway; the test configuration cannot be executed in back to back by connecting Acceleron and TOB interfaces together.

The gateway is configured to route the SIP calls received on the IP address 192.168.1.100 port 5060 to its E1 interface.

Step-by-Step Instructions

The final IxLoad configuration as a result of these steps is provided on the blackbook.ixiacom.com Web site - see *IxLoad 5.10 Voice - SIP to PSTN.crf*. To import a Compressed Repository File (crf) in IxLoad, use the command **Import** under the **File** menu.

The step-by-step instructions highlight how to set the essential parameters of this configuration and explain additional options, which may be used to change the behavior of the test.

Create the Network for SIP Activity

1. Open the IxLoad GUI.
2. Add a NetTraffic activity and select the network **Network1**.
3. Set the network parameters as shown in the following table:

Table 55. Summary of *Network 1* parameters

IP Type	Count	Address	Mask	Increment	Gateway	Gateway Increment	MSS (RX)
IPv4	30	192.168.1.101	24	0.0.0.1	0.0.0.0	0.0.0.0	1460

Add and Configure the SIP Peer Activity

1. Add a **VoIPSIP Peer** activity.
2. Rename the new added activity from VoIPSipPeer1 to SIPMakeCall.
3. Edit the **Scenario** by adding the **SIP Make Call - Complete** procedure, the **Voice Session** function, and the **End Call Initiate** procedure.

TIP: an existing test scenario may be loaded instead of creating it from scratch. For example, you may load *VS_022_DUT_SIPv4 MakeCall - EndCall with RTP - 33s.tst*, provided with the product.

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

- Click the **Dial Plan** tab and configure the **Destination** phone numbers to user defined values: *717444[0001-0030]*; to modify the user defined values, the **Override phone numbers for destination activity** check box has to be selected.

The screenshot shows the 'Dial Plan' configuration window for SIP activity. The 'Dial Plan' tab is selected. The 'Source' section has 'IPs' set to 'The source IP addresses are taken from the associated Network (see Traffic - Network mapping tables in the test)'. The 'Phone numbers' section has 'Phone book entry' set to '<None>' and 'User defined' set to '160[00000000-]'. The 'Destination' section has 'IPs' set to 'None'. The 'Override phone numbers from destination activity' checkbox is checked. The 'Phone book entry' is set to '<None>' and the 'User defined' field is set to '717444[0001-0030]'. The 'Use Tel URI parameters' checkbox is unchecked, with the value 'phone-context=example.com'.

Figure 147. : Dial plan for the SIP activity

Note: Same dial plan has to be configured on the gateway.

- Click the **SIP** tab to configure the destination of SIP messages. Select the **Use external server** check box and set the **Server address** to *192.168.1.100* and **Domain name or local IP** to *voice.ixiacom.com*

The screenshot shows the 'SIP' configuration window for SIP activity. The 'SIP' tab is selected. The 'Enable signaling on this activity' checkbox is checked. The 'SIP Port' is set to '[5060-]'. The 'Use external server' checkbox is checked. The 'Server address' is set to '192.168.1.100', the 'Server port' is set to '5060', and the 'Domain name or local IP' is set to 'voice.ixiacom.com'. The 'Outbound proxy', 'Registrar server', and 'Auto register simulated user agents' checkboxes are unchecked. The 'Override registrar' checkbox is unchecked, with the value 'IP:PORT'. The 'Construction of SIP messages' section has 'Override default contact settings' checked, with the 'Edit Contact ...' button. The 'Override default destination domain name or host:por' checkbox is unchecked, with the 'Domain name or Host:Port' field. The 'Use Tel URI scheme for Source' and 'Use Tel URI scheme for Destination' checkboxes are unchecked.

Figure 148. SIP Server settings

Note: The server address and domain name has to match the ones configured on the gateway.

- Click the **Audio** tab to enable media and set the call duration. Select the **Enable audio on this activity** check box, click the **Play for** option, set **Play for** to *50 seconds*, and select the **Perform MOS** check box. With these settings, the calls will have the duration of 50 seconds.

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

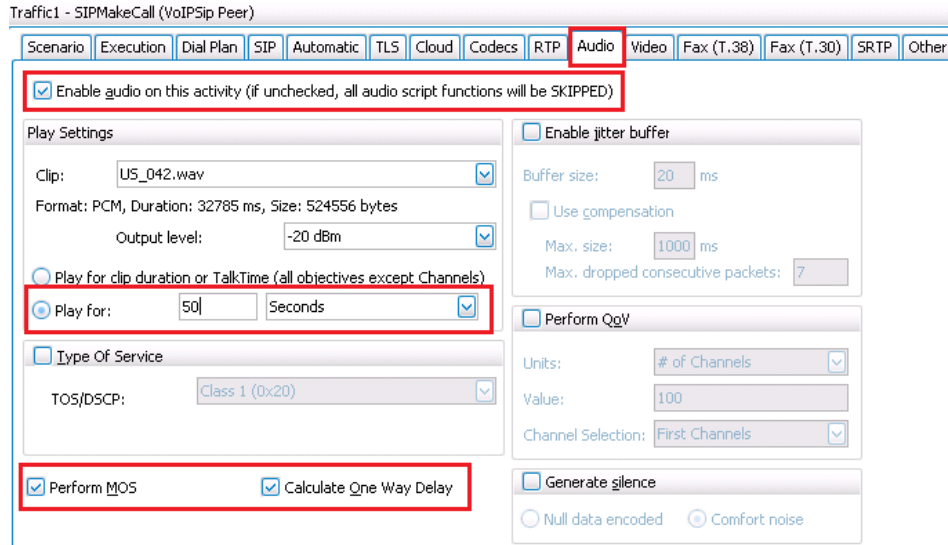


Figure 149. Audio settings for SIP activity

Add and Configure a PSTN Net Traffic

1. Click **Add Traffic Flow Element** under **Terminate**, and then click **PSTN Net Traffic**.

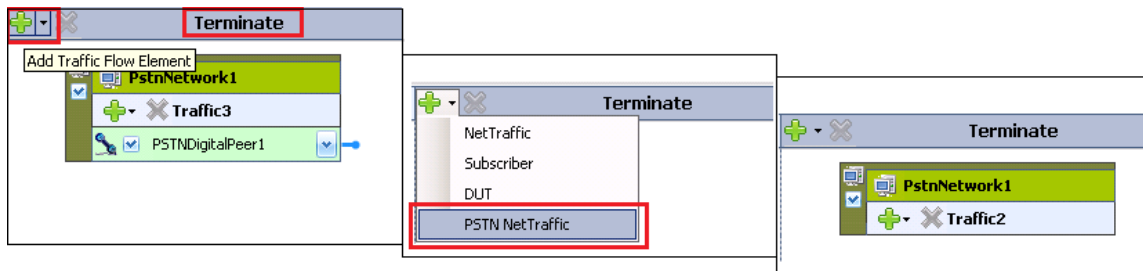


Figure 150. Add a PSTN Net Traffic

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

2. Click **PstnNetwork1** and edit the network parameters:

Table 56.

Parameter	Value	Notes											
Type	<i>E1</i>	The type is selectable between E1 or T1 Important: All board spans of the same board must be configured in the same way, either T1 or E1; mixed configurations of one board spans are not allowed.											
Framing Format	<i>G704</i>	The T1/E1 framing format can be either of the following: <ul style="list-style-type: none"> For T1: D4 (Superframe, default) or ESF (Extended Super Frame). For E1: G.704 without CRC4 (default) and G.704 with CRC4. 											
Line encoding	<i>HDB3</i>	The line encoding can be: <ul style="list-style-type: none"> For T1: AMI (no zero suppression mechanism, data is encoded in the bipolar Alternate Mark Inversion format) or B8ZS (Bipolar Eight Zero Suppression). For E1: AMI (Alternate Mark Inversion - standard line coding with no zero code suppression) and HDB3 (High Density Bipolar 3 code that uses patterns of bipolar violations to replace sequences of 4 zero data bits to maintain ones density on clear channel transmission) 											
Signaling	<i>ISDN</i>	The signaling can be one of the following: <ul style="list-style-type: none"> T1 CAS: In-band signaling, 24 channels/span T1 ISDN PRI: Out-of-band signaling, 23 channels/span T1 NCC: No signaling E1 CAS: In-band signaling, 30 channels/span E1 ISDN PRI: Out-of-band signaling, 30 channels/span E1 NCC: No signaling, 30 channels/span 											
Protocol	<i>QSIG</i>	There are several variants of ISDN and CAS depending on the type of interface. The following protocols are supported in the current version of IxLoad: <table border="1" data-bbox="787 1514 1289 1892"> <thead> <tr> <th>Interface Type</th> <th>Signaling</th> <th>Protocol</th> </tr> </thead> <tbody> <tr> <td rowspan="2">E1</td> <td>ISDN PRI</td> <td>QSIG VN6 KHT KOR TWN AUS</td> </tr> <tr> <td>CAS</td> <td>MFC R2</td> </tr> <tr> <td>T1</td> <td>ISDN PRI</td> <td>4ESS 5ESS DMS NT2</td> </tr> </tbody> </table>	Interface Type	Signaling	Protocol	E1	ISDN PRI	QSIG VN6 KHT KOR TWN AUS	CAS	MFC R2	T1	ISDN PRI	4ESS 5ESS DMS NT2
Interface Type	Signaling	Protocol											
E1	ISDN PRI	QSIG VN6 KHT KOR TWN AUS											
	CAS	MFC R2											
T1	ISDN PRI	4ESS 5ESS DMS NT2											

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

Parameter	Value	Notes				
		<table border="1"> <tr> <td></td> <td><i>NTT</i> <i>QSIG</i> <i>DSS1</i></td> </tr> <tr> <td>CAS</td> <td><i>E&M</i></td> </tr> </table>		<i>NTT</i> <i>QSIG</i> <i>DSS1</i>	CAS	<i>E&M</i>
	<i>NTT</i> <i>QSIG</i> <i>DSS1</i>					
CAS	<i>E&M</i>					
Variants	<i>TE</i>	The variant type depends on the Signaling settings: <ul style="list-style-type: none"> For T1/E1 ISDN PRI: NT (network termination), TE (terminal equipment) For T1 CAS: FGB, FGD, Immediate For E1 CAS: None 				
Publish Statistics	<i>Enabled</i>	If selected, statistics for the range are computed and displayed in the StatViewer component of IxLoad.				

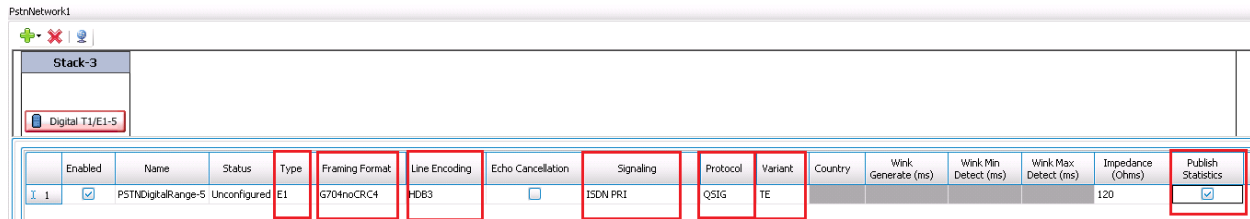


Figure 151. PSTN Network Settings


Note: All the parameters are consistent with the parameters set on the DUT; it has the following settings:

Table 57.

Parameter	Value	Notes
Type	<i>E1</i>	
Framing Format	<i>G704</i>	
Line encoding	<i>HDB3</i>	
Signaling	<i>ISDN</i>	
Protocol	<i>QSIG</i>	
Variants	<i>NT</i>	If one device is set to Terminal the connected device has to be set to Network.
Publish Statistics	<i>Enabled</i>	

Note: Multiple ranges can be configured for the same PSTN Network; this allows creating multiple PSTN activities on the same Net Traffic, each activity with its own network parameters.

Add a PSTN Activity

1. In the **PstnNetwork1 Traffic2**, click the  button to add an activity.

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

2. Choose the **Voice > PSTNDigital Peer** activity type; the activity **PSTNDigitalPeer1** will be added.

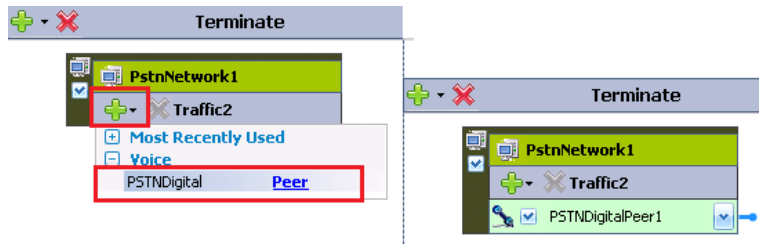


Figure 152.

Edit the Call flow for PSTNDigitalPeer1

Similar to the advanced VoIP protocols (for example, SIP Peer), the call flow for PSTN activity is defined by one channel of a test scenario. The same test scenario may contain VoIP and PSTN channels; in this way, the actions on one activity may be synchronized with the actions on the other activity.

This example uses a single test scenario for SIP Make Call activity and for PSTN Receive call activity. The scenario for SIP Make Call was created at the **Step 3 of Add and Configure the SIP Peer Activity**. The steps to add the second test scenario channel to the existing one, and associate it with the PSTN activity are as follows:

1. Click the **SIPMakeCall** activity.
2. Click the **Scenario** tab.

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

3. Click **Add Channel**. A new channel is added to the test scenario, and a window to map the new created channel to an activity appears.

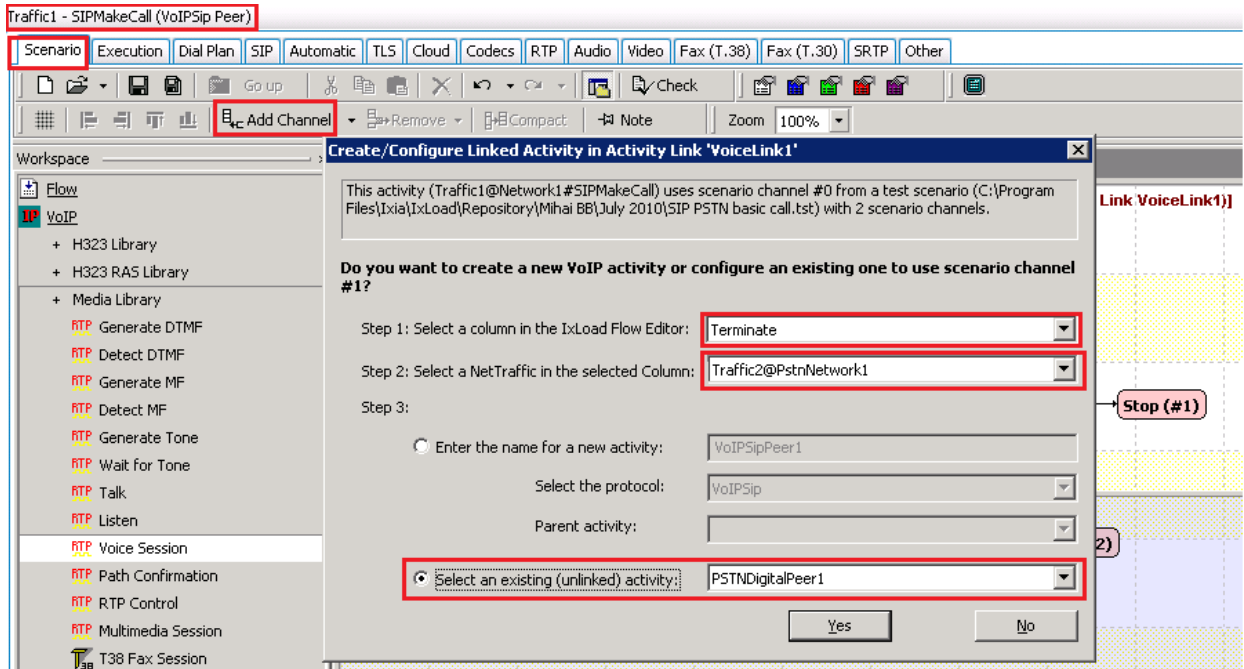


Figure 153. Add a new test scenario channel and map it to an activity

4. In the **Create/Configure Linked Activity in Activity Link 'VoiceLink1'** window, click **Terminate** at step1, **Traffic2@PstnNetwork1** at step 2, and **Select an existing (unlinked) activity PSTNDigitalPeer1** at step3. Close the window by clicking **Yes**.
5. Click the **PSTNDigitalPeer1** activity.

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

6. In the **Scenario** tab, add and connect the T1/E1 script objects **Make Call**, **Voice Session**, and **End Call**. Add triggers to between the **SIPMakeCall** activity and **PSTNDigitalPeer1** activity.

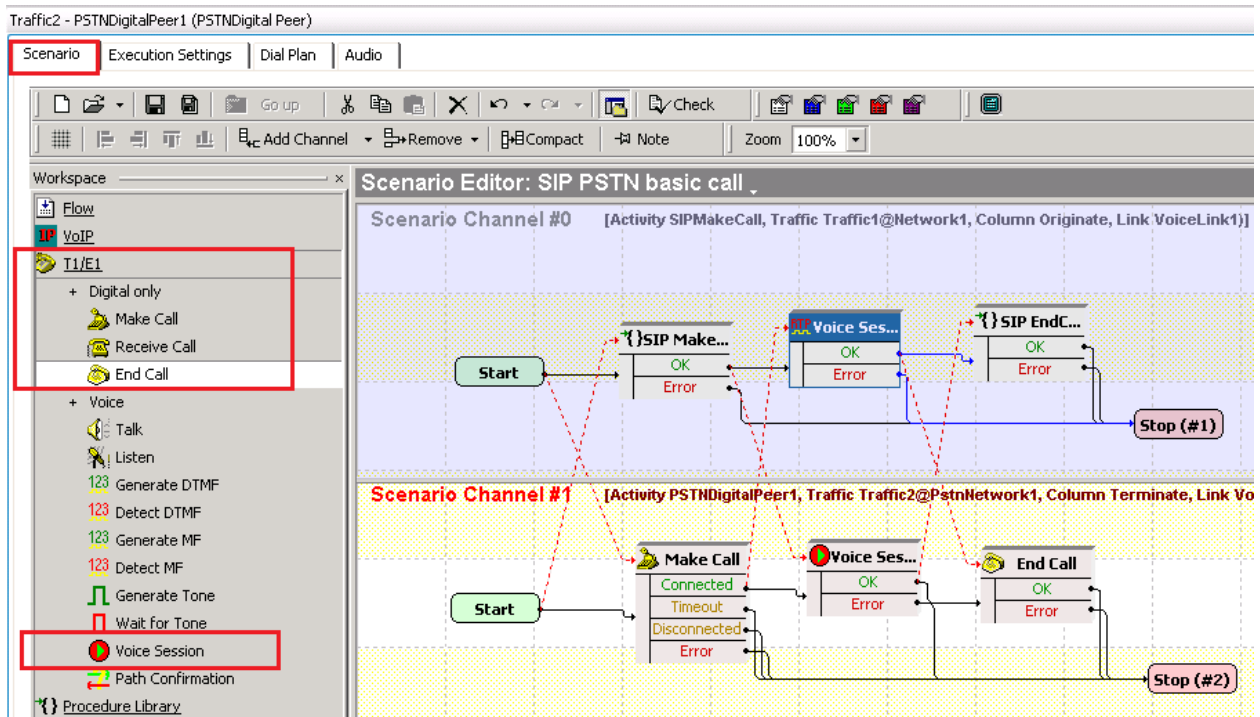


Figure 154. SIP / PSTN test scenario

Edit the Dial Plan for PSTNDigitalPeer1

1. Click the **Dial Plan** tab.
2. Edit the **Source Phone Number/Specify** field to: `717444[0001-0030]`; this is the same sequence defined for the SIP Make Call activity as Destination Phone Numbers.

Note: Depending on the distribution group defined on the DUT, IxLoad may be configured for other matching schemes of incoming calls: DNIS to receive, ANI to receive.

Set Audio Parameters for PSTNDigitalPeer1

The Audio parameters (the clip to be played and the played duration) may be set in the test script objects or at the Activity level. To control the audio parameters from the test script objects, the **Overwrite playback activity settings** check box has to be selected in the **Voice Session** function (or whatever media function is used). It is preferable to use the settings at the Activity level to allow control of the audio parameters from the automation scripts.

1. Click the **Audio** tab.
2. Select the **Play for** check box.
3. Set **Play for** to *50 seconds*.

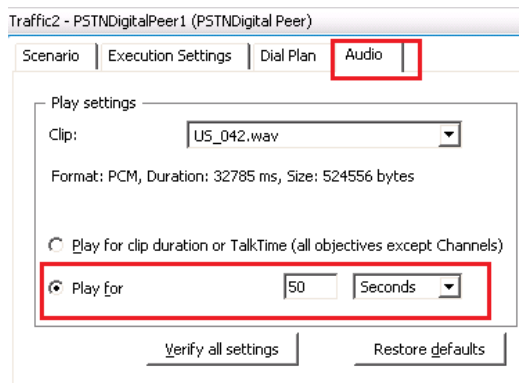



Figure 155. Audio settings for PSTNDigitalPeer

Configuring the Timeline and Objective

1. Click **Timeline & Objective** from the test configuration panel.
2. Set the test **Objective Type** to **Channels**. This is equivalent with concurrent active calls.
3. Set the test **Objective Value** to *30*. This value may be increased if multiple spans are available; the number of IP addresses for SIP activity and the dial plan range have to be extended.
4. On the **Timeline** tab, set the **Ramp Up Value** to *10*.
5. Set the **Ramp Up Interval** to *1 second*.
6. Set the **Sustain Time** to *10 minutes*.

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

Assigning Interfaces to the NetTraffics

1. Click **Part Assignments** from the test configuration panel.
2. Add a chassis by clicking **Add Chassis** .
3. Assign a port of one Application Load Module to the SIP NetTraffic. In this example, the first port of second card 1.2.1 (an XMV4) is used for VoIP traffic.
4. Assign one port of **Adapter Modules for Telephony Boards** (ADM01 TOB) to the PSTN NetTraffic. In this example, the *Card 7* is used for PSTN traffic. One Adapter Module for Telephony Boards appears as having a single port; indeed one ADM01 TOB has a single processor that controls all the E1/T1 spans. A single adaptor card can accommodate one or two telephony modules, each with 2, 4, or 8 spans. That means a single ADM01 TOB card may have at least 2 E1/T1 spans and at most 16 spans. The association of a specific span to a PSTN activity is done in the next step.

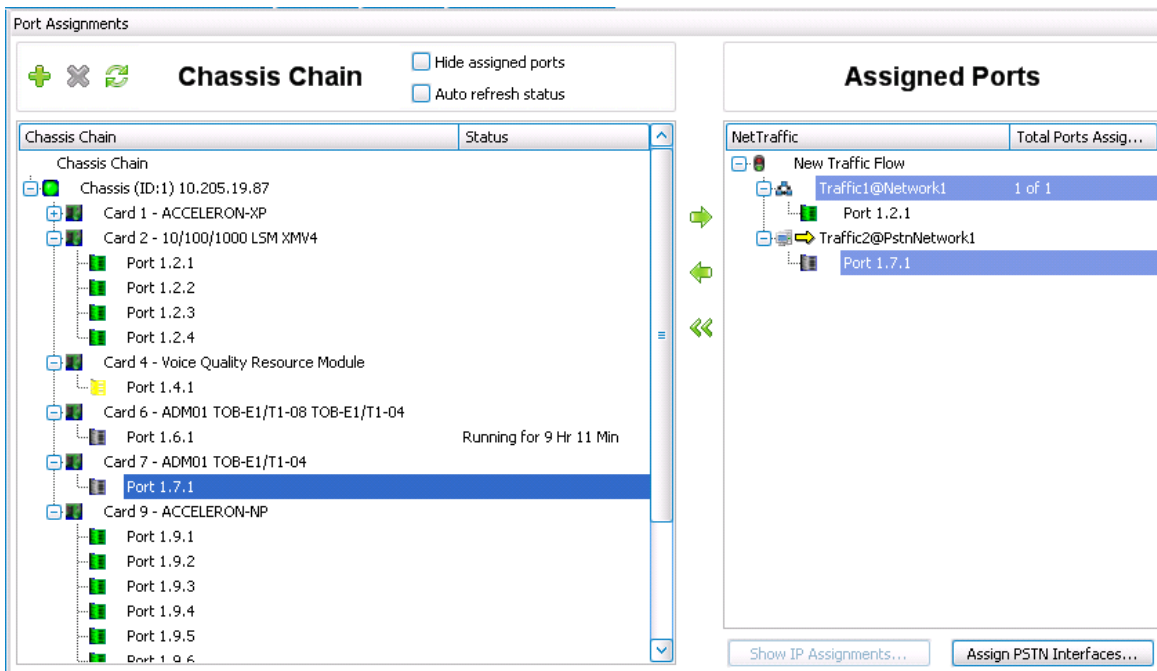


Figure 156. Assigning interfaces to VoIP and PSTN NetTraffics

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

Assigning PSTN Interfaces

1. In the **Port Assignment** page, click **Assign PSTN Interfaces**.
2. Select the desired span in the **Available Ports** pane and associate it with the **PSTN Network Range**. In this example, Span#3 is used.

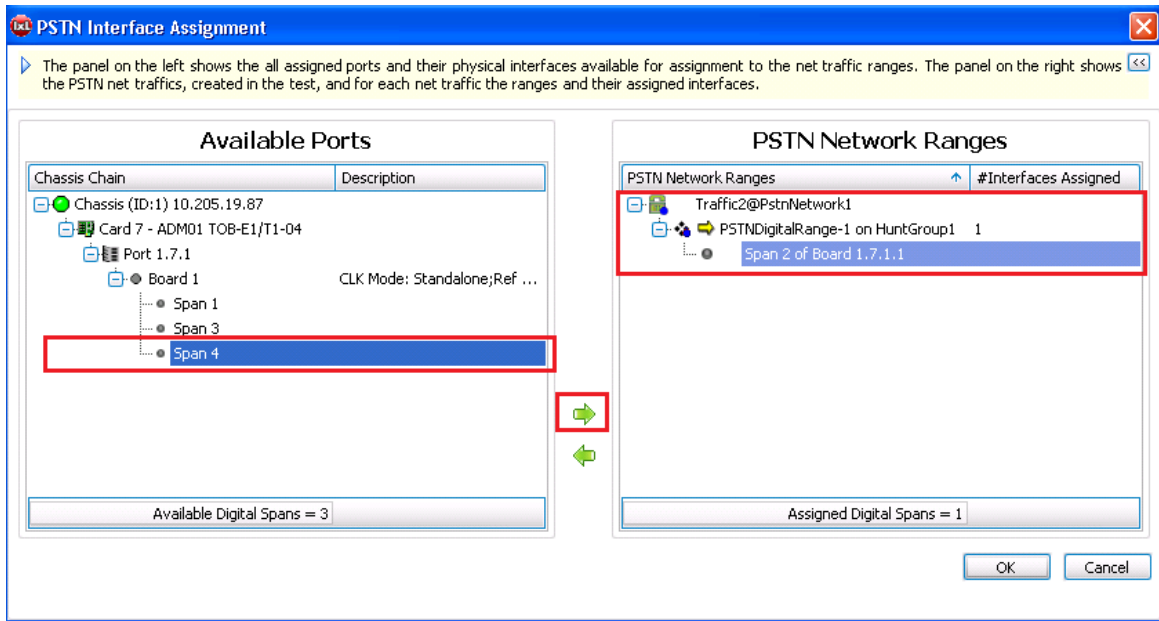


Figure 157. Associate a span to a PSTN Network Range

Configuring PSTN Board Clock Settings

For T1/E1 boards, configuring the clock settings is important. With respect to the ECTF H.110 specification, the telephony boards supported by IxLoad can be configured to use group A clock lines only (fallback to group B is not supported) configured as either standalone (that is, decoupled from the CT bus), master, or slave.

For the current release, the clock settings apply at the ADM carrier board level only (it is not possible to synchronize the clocks of telephony modules from different adaptor modules)

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

To configure PSTN board clock settings, do the following:

1. In the Assigned Ports or in the PSTN Interfaces Assignment window, right-click an assigned ADM board and click Clock Settings. In the **Clock Settings** window that appears, all PSTN boards installed in the ADM carrier board are displayed.

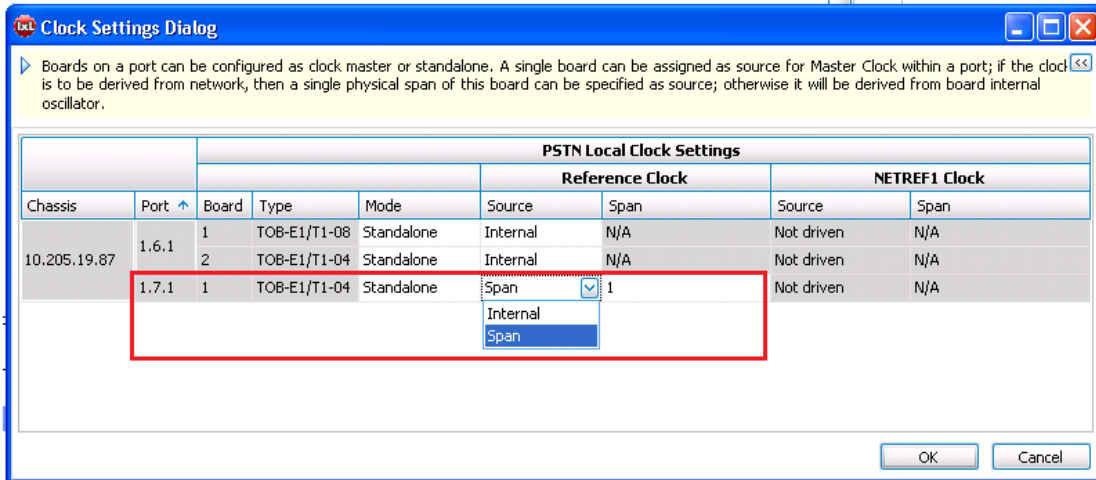


Figure 158. Configuring the PSTN board clock setting

2. Set the **Source Reference Clock** to Internal. The DUT is configured to synchronize its clock to the Span, so both IxLoad and DUT are synchronized to the clock generated by IxLoad.

Running the Test

1. Save the configuration.
2. Set Global settings (optional step). There are several parameters that control the PSTN interfaces behavior. These parameters are usually not changed between devices under test. Their values can be changed in the Global Settings, that can be accessed by clicking **Global Settings** in the scenario tab of a VoIP or T1/E1 activity:

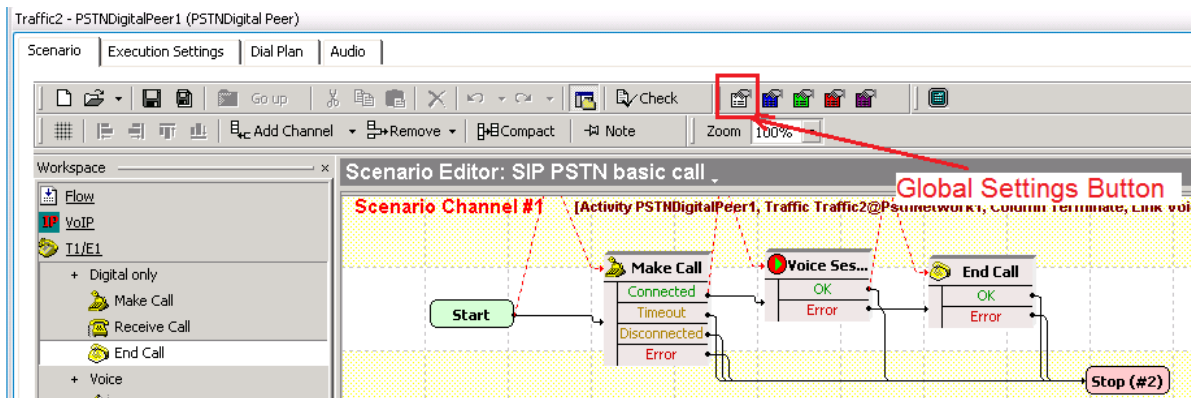


Figure 159.

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

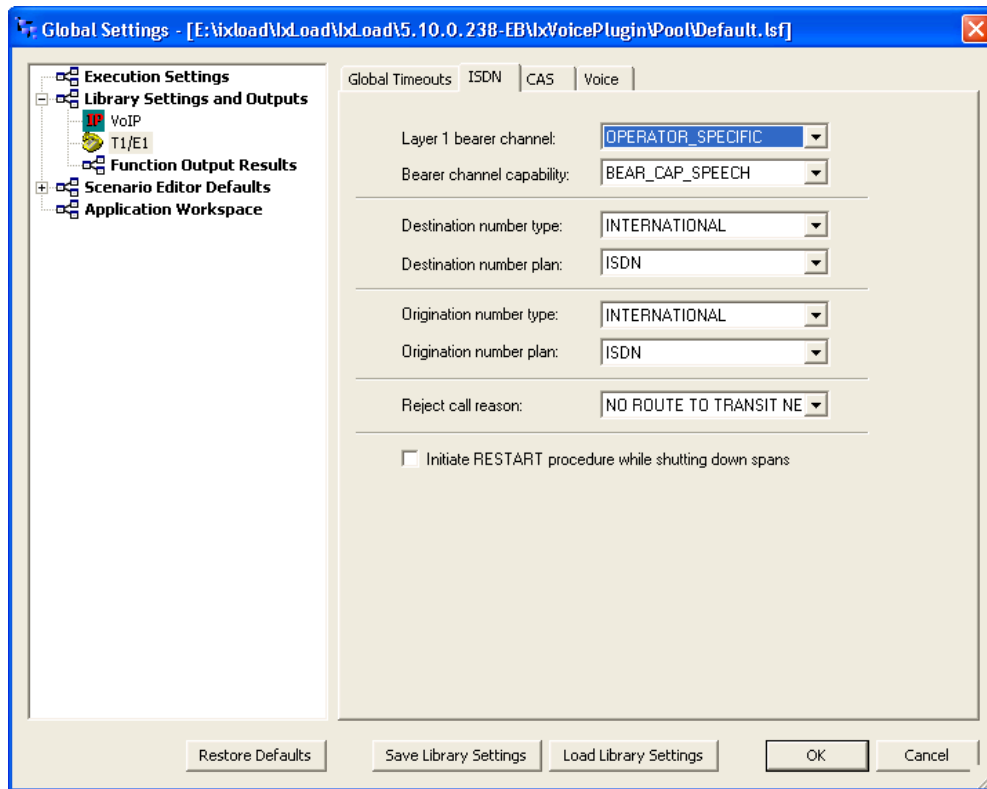


Figure 160. Global Settings for Voice Activities

3. Start the test by clicking **Run**.

Results Analysis

The following questions provide guidelines on how to recognize specific problems during or at the end of the test execution:

1. Has the test objective been achieved? Check the **Calls** view.

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

Table 58. Calls statistics

Statistic Name	Value	Questions
SIP Attempted Calls SIP Connected Calls PSTN Received Calls PSTN Answered Calls	300 Same value for all stats	<ol style="list-style-type: none"> 1. Is the number of Attempted calls equal with the numbers of Connected calls? Are all the originated calls connected? 2. Is the number of Attempted calls equal with the number of received and Answered calls?

Note: If the number of Connected Calls is equal with the number of Received calls but less than the number of Attempted calls, then the DUT cannot route all the calls. The reason may be:

- the miss-configuration of some of the spans if many spans are used
- the mismatch of source and destination dial-plan
- lack of resources on DUT to route all the calls.

It is possible, on some DUTs, to observe the number the number of Attempted Calls equal with the number of Connected calls, but different then the number of Received/Answered Calls. In this case, the DUT completes the SIP call legs, but fails to set the PSTN calls, showing a functionality error.

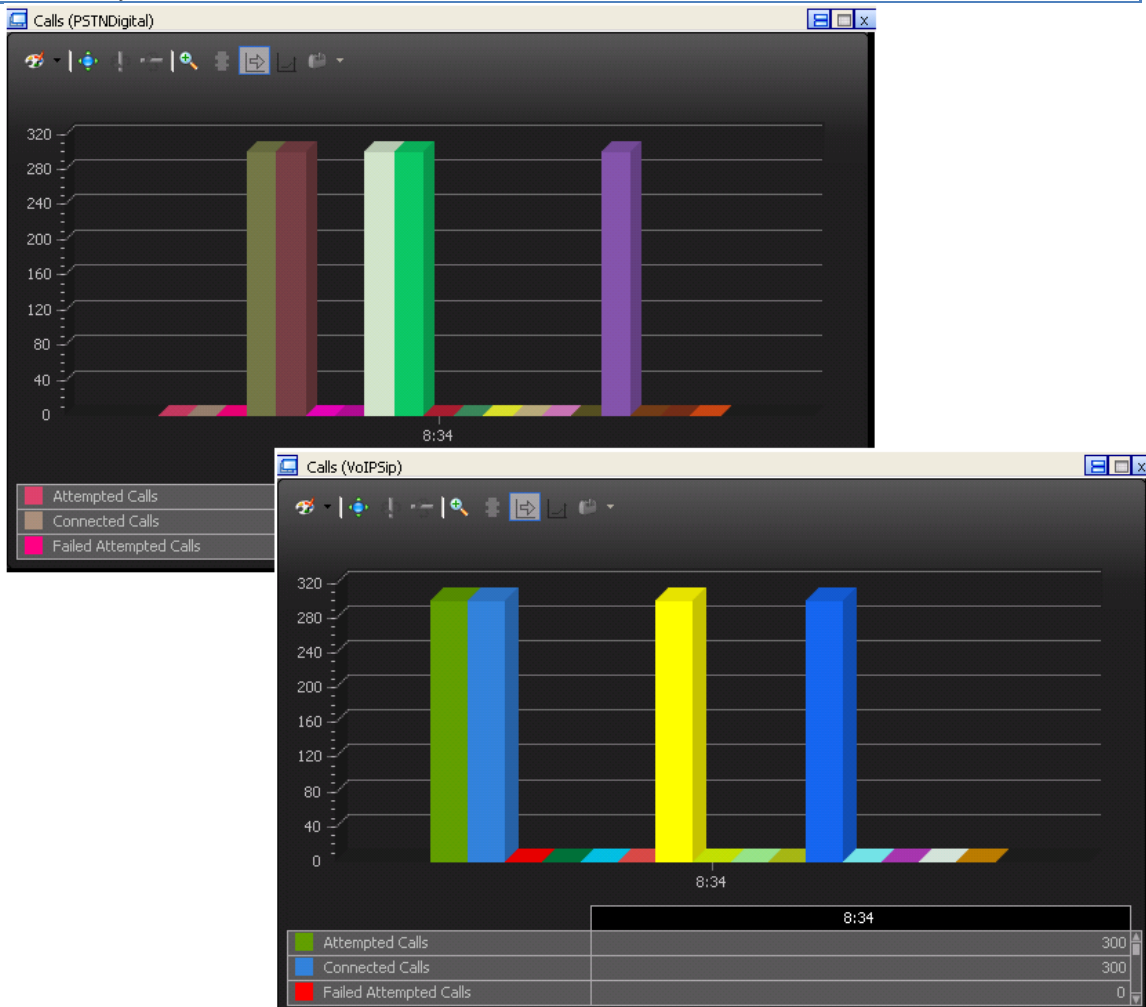


Figure 161. SIP and PSTN Calls Statistics

Test Case: Determining the Capacity of a VoIP to PSTN Gateway

2. Have any call failures been reported? Check the **Calls** view.

Table 59. Call failure statistics

Statistic Name	Value	Questions
SIP Failed Attempted Calls PSTN Rejected Calls	0	1. Is any called fail or rejected?
<p>Note: If there are calls failures on the SIP side, the SIP Message stats may be checked to find the reason of failures (5xx responses means the internal DUT error, 401 or 407 responses means the DUT has the authentication enabled, and the call flow should use also authentication, 404 response means the destination phone number is not configured on the DUT The SIP event viewer shows the errors in the SIP call flow.</p>		

3. Have any scenario loop failures been reported? Check the **Loops** statistics view.

Table 60. Statistics highlighting the pass/fail result based on call flow execution

Statistic Name	Questions
Total Loops	1. Are the Successful Loops and Total Loops values equal?
Successful Loops	
Failed Loops	2. Have any Failed Loops, Aborted Loops or Warning Loops been reported?
Aborted Loops	
Warning Loops	
<p>Note: failed/aborted and warning loops highlights failures at the scenario level.</p>	

Troubleshooting and Diagnostics

The following table summarizes some of the common issues that may be encountered when running a call rate test.

Table 61.

Issue	Troubleshooting Solution
The number of concurrent calls is not sustained for the entire test duration	<p>Check the SIP event viewer; a pattern in the phone numbers failing to connect calls is an indication of miss-configuration of some phones (wrong destination phone number).</p> <p>Check the SIP Retransmissions counter, Call Setup Time, and End Call Time measurements. A high number of retransmissions is an indication that the SUT cannot maintain the load generated. This leads to larger call setup and teardown times, which affects the number of users available to place new calls.</p>

Test Variables

Table 62.

Parameter Name	Current Value	Additional Options
IP Version	IPv4	IPv6
Concurrent Calls	30	Up to 480 concurrent calls using a single adapter card with two eight spans E1/T1 modules (2 x 8 x 30). More than 480 concurrent calls using more than one adapter card and appropriate number of E1/T1 modules.
Direction of calls	SIP to PSTN	PSTN to SIP; the call flow (in test scenario) and the dial plan have to be changed to run this test.
TDM interface type	E1	T1
PSTN Protocol	ISDN / QSIG	ISDN: VN6, KHT, KOR, TWN, AUS CAS / MFC R2

Conclusions

This configuration covered the main parameters of the SIP to PSTN configuration by using a practical example allowing the user to control the concurrent number of active calls running capacity tests. The results section covered the main statistics that may highlight an issue.

Test Case: Determining the Performance of a Session Border Controller

Overview

The challenges and test methodology for **Session Border Controllers** and **Application Layer Gateways** testing were discussed in this Black Book in the section *TEST CASE: DETERMINING THE MAX CALL SETUP RATE FOR SIP-BASED DEVICES AND SYSTEMS*. That use case is focused on originating calls from one network (Private) and terminate them on the other network (Public). In that case, a single call leg is set for each call. There are cases when the calls are established between two endpoints located in the same network while the SIP server is in a different network. In these cases, the DUT handles two call legs for each call: one between the originator of the call and SIP proxy and second between SIP Proxy and the terminator of the call. In addition to the call setup messages, the DUT has to transmit the Registration requests from the SIP User Agents to the SIP Registrar server. The typical SIP call flow for Registration and Basic Call is as follows:



Figure 162. SIP Call flow for Basic Call via a proxy over an SBC

Objective

This test determines the maximum rate of SIP calls that can be established through a DUT between endpoints in the Private network when the SIP Proxy is in a Public network.

Setup

In this test topology, a pair of Ixia ports emulates SIP User Agent in a private network and the third Ixia port emulates the SIP Registrar and Proxy server in the public network. The SIP User Agents register to the emulated SIP Register server and establish calls through the emulated SIP Proxy server over the DUT—an Application Layer Gateway or a Session Border Controller.

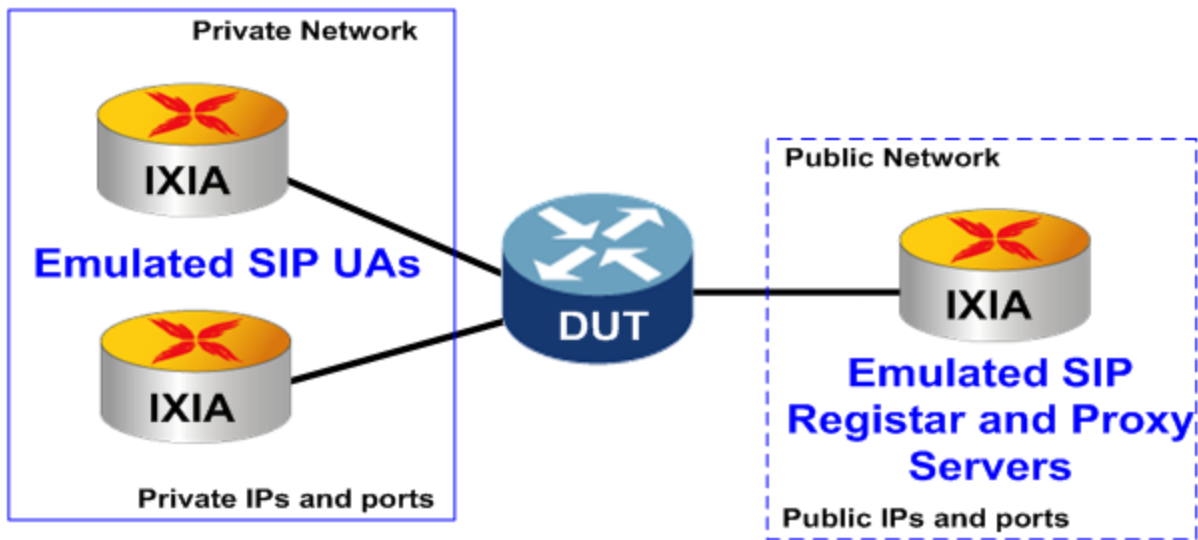


Figure 163. Test Topology for SIP user agents and SIP Server emulation

Step-by-Step Instructions

The final IxLoad configuration as a result of these steps is provided on the blackbook.ixiacom.com Web site - see *IxLoad 5.10 Voice - SIP Back to Back User Agent.crf.crf*. To import a Compressed Repository File (crf) in IxLoad, use the command **Import** under the **File** menu.

The step-by-step instructions highlight how to set the essential parameters of this configuration and explain additional options, which may be used to change the behavior of the test.

Setting the Network Parameters

1. Import the configuration **IxLoad 5.10 Voice - SIP Back to Back User Agent.crf**.

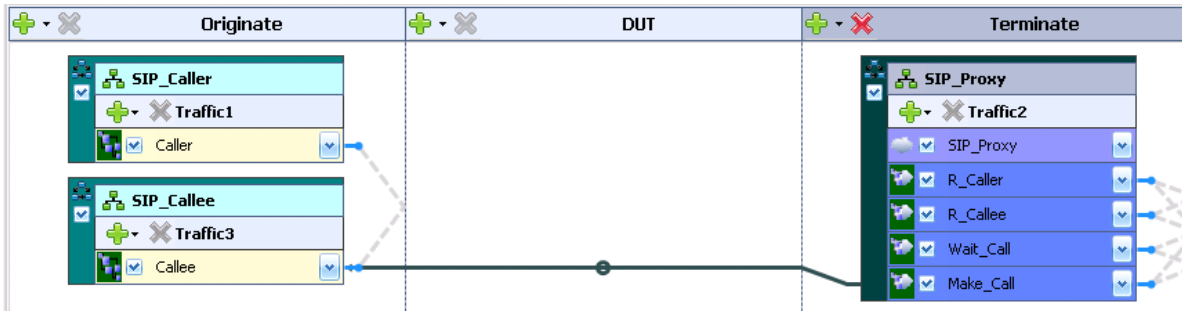


Figure 164. SIP Back to Back User Agent configuration

2. Click the **SIP_Caller** NetTraffic and change the network settings accordingly to the particular test topology. The number of IP addresses should match the number of channels defined in Test Objective (1000 in this example).
3. Click the **SIP_Callee** NetTraffic and change the network settings accordingly to the particular test topology. The number of IP addresses should match the number of channels defined in Test Objective (1000 in this example).
4. Click the **SIP_Proxy** NetTraffic and, for IP-R3, the first IP range, change the network settings accordingly to the particular test topology. This will be the IP address of the emulated SIP server. The same value should be set as the SIP Proxy IP address in the **Caller** and **Callee** activities. The IP addresses of the other ranges under **SIP_Proxy** NetTraffic may remain unchanged while these are internal IP addresses used just to route the messages between the SIP Cloud to the SIP activities. The number of IP addresses for the range IP-R3 should be equal with the number of ports number of Ixia pots used to emulate the SIP Proxy—1 in this case. The number of IP addresses of other ranges should be equal with the number of channels defined in Test Objective (1000 in this example).

Test Case: Determining the Performance of a Session Border Controller

The screenshot shows a network configuration interface with three main sections: **Originate**, **DUT**, and **Terminate**. Under **Originate**, there are **SIP_Caller** and **SIP_Callee** components. Under **Terminate**, there is a **SIP_Proxy** component. The **SIP_Proxy** configuration is expanded to show a table of IP addresses.

Enabled	Name	Status	IP Type	Address	Mask	Increment	Count	Gateway	Gateway Increment	Gateway Increm
<input checked="" type="checkbox"/>	IP-R3	Unconfigured	IPv4	40.40.50.1	16	0.0.0.1	1	0.0.0.0	0.0.0.0	Increment every
<input checked="" type="checkbox"/>	IP-R4	Unconfigured	IPv4	40.40.100.1	16	0.0.0.1	1000	0.0.0.0	0.0.0.0	Increment every
<input checked="" type="checkbox"/>	IP-R5	Unconfigured	IPv4	40.40.110.1	16	0.0.0.1	1000	0.0.0.0	0.0.0.0	Increment every
<input checked="" type="checkbox"/>	IP-R6	Unconfigured	IPv4	40.40.120.1	16	0.0.0.1	1000	0.0.0.0	0.0.0.0	Increment every
<input checked="" type="checkbox"/>	IP-R7	Unconfigured	IPv4	40.40.130.1	16	0.0.0.1	1000	0.0.0.0	0.0.0.0	Increment every

Figure 165. Set the IP address of the emulated SIP Server

- Click **Traffic2** under **SIP_Proxy** NetTraffic and check whether each activity has the SIP traffic mapped to one and only one network range. Check that the **Distribution Group** for the range used by **SIP_Proxy** activity (the cloud) is *Round Robin*.

The screenshot shows the **Traffic2** configuration interface. The **User Source IP Mapping** section is expanded to show a table of network ranges and their mapping to activities.

Network Ranges By Port Distribution Group	Activities & Endpoints
<input checked="" type="checkbox"/> [IP-3] Group1: Consecutive IPs	SIP_Proxy, R_Caller, R_Callee, SIP, RTP, T38, SIP
Network Range IP-R4 in SIP_Proxy (40.40.100.1+1000)	<input checked="" type="checkbox"/> SIP, <input type="checkbox"/> RTP, <input type="checkbox"/> T38, <input type="checkbox"/> SIP
Network Range IP-R5 in SIP_Proxy (40.40.110.1+1000)	<input type="checkbox"/> SIP, <input type="checkbox"/> RTP, <input type="checkbox"/> T38, <input checked="" type="checkbox"/> SIP
Network Range IP-R6 in SIP_Proxy (40.40.120.1+1000)	<input type="checkbox"/> SIP, <input type="checkbox"/> RTP, <input type="checkbox"/> T38, <input checked="" type="checkbox"/> SIP
Network Range IP-R7 in SIP_Proxy (40.40.130.1+1000)	<input type="checkbox"/> SIP, <input type="checkbox"/> RTP, <input type="checkbox"/> T38, <input type="checkbox"/> SIP
<input checked="" type="checkbox"/> [IP-3] Group2: IP Round Robin	<input checked="" type="checkbox"/> SIP, <input type="checkbox"/> RTP, <input type="checkbox"/> T38, <input type="checkbox"/> SIP
Network Range IP-R3 in SIP_Proxy (40.40.50.1+1)	<input checked="" type="checkbox"/> SIP, <input type="checkbox"/> RTP, <input type="checkbox"/> T38, <input type="checkbox"/> SIP

Figure 166. SIP activities mapping to IP ranges for emulated SIP Proxy

Test Case: Determining the Performance of a Session Border Controller

6. Click the **SIP_Caller** NetTraffic and change the network settings accordingly to the particular test topology. The number of IP addresses should match the number of channels defined in Test Objective (1000 in this example).
7. Click the **SIP_Callee** NetTraffic and change the network settings accordingly to the particular test topology. The number of IP addresses should match the number of channels defined in Test Objective (1000 in this example).
8. Click the **SIP_Proxy** NetTraffic and, for IP-R3, the first IP range, change the network settings accordingly to the particular test topology. This will be the IP address of the emulated SIP server. The same value should be set as the SIP Proxy IP address in the **Caller** and **Callee** activities. The IP addresses of the other ranges under **SIP_Proxy** NetTraffic may remain unchanged while these are internal IP addresses used just to route the messages between the SIP Cloud to the SIP activities. The number of IP addresses for the range IP-R3 should be equal with the number of ports number of Ixia pots used to emulate the SIP Proxy—1 in this case. The number of IP addresses of other ranges should be equal with the number of channels defined in Test Objective (1000 in this example).

The screenshot shows a network configuration interface with three main sections: **Originate**, **DUT**, and **Terminate**. The **Originate** section contains **SIP_Caller** and **SIP_Callee** components. The **Terminate** section contains **SIP_Proxy**, **R_Caller**, **R_Callee**, **Wait_Call**, and **Make_Call** components. The **SIP_Proxy** component is highlighted with a red box. Below the components is a **Stack-2** configuration area with tabs for **Filter-3**, **TCP-3**, **Settings-3**, **GratARP-3**, and **DNS-3**. The **Ethernet-3** interface is also visible. At the bottom is a table for IP configuration with a red box around the first row (IP-R3).

	Enabled	Name	Status	IP Type	Address	Mask	Increment	Count	Gateway	Gateway Increment	Gateway Increm
▶ 1	<input checked="" type="checkbox"/>	IP-R3	Unconfigured	IPv4	40.40.50.1	16	0.0.0.1	1	0.0.0.0	0.0.0.0	Increment every
2	<input checked="" type="checkbox"/>	IP-R4	Unconfigured	IPv4	40.40.100.1	16	0.0.0.1	1000	0.0.0.0	0.0.0.0	Increment every
3	<input checked="" type="checkbox"/>	IP-R5	Unconfigured	IPv4	40.40.110.1	16	0.0.0.1	1000	0.0.0.0	0.0.0.0	Increment every
4	<input checked="" type="checkbox"/>	IP-R6	Unconfigured	IPv4	40.40.120.1	16	0.0.0.1	1000	0.0.0.0	0.0.0.0	Increment every
5	<input checked="" type="checkbox"/>	IP-R7	Unconfigured	IPv4	40.40.130.1	16	0.0.0.1	1000	0.0.0.0	0.0.0.0	Increment every

Figure 167. Set the IP address of the emulated SIP Server

Test Case: Determining the Performance of a Session Border Controller

- Click **Traffic2** under **SIP_Proxy** NetTraffic and check whether each activity has the SIP traffic mapped to one and only one network range. Check that the **Distribution Group** for the range used by **SIP_Proxy** activity (the cloud) is *Round Robin*.

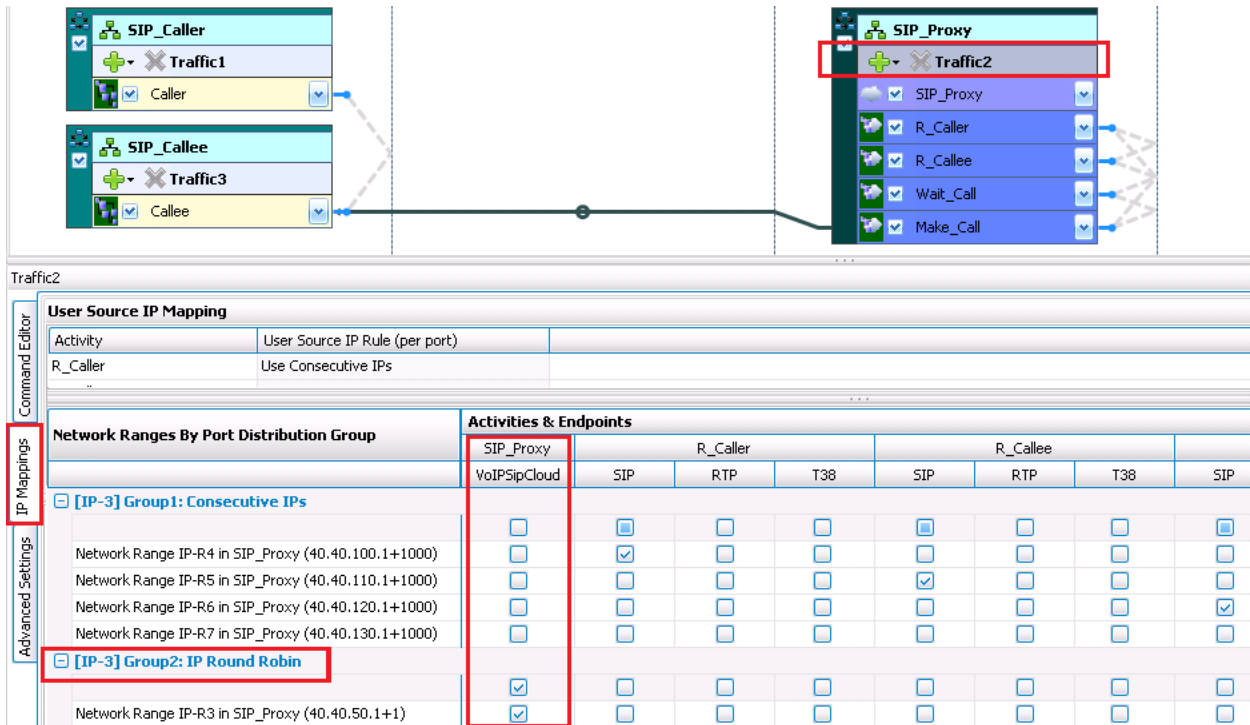


Figure 168. SIP activities mapping to IP ranges for emulated SIP Proxy

Setting the SIP_Caller Activity Parameters

- Click the **Caller** activity under **SIP_Caller** NetTraffic.
- Click the **Scenario** tab, and check whether the call flow is the required one. In this example, the call flow is: Register, Make Call, Play Audio Streams, and Initiate end call. Refer the Black Book sections describing how to create and change the call flow in the Scenario Editor or the IxLoad VoIP User Guide for instructions of how to modify the call flow.
- Click the **Dial Plan** tab and set the **Source Phone Numbers** *5551[0000-]* in this example.
- Set the **Destination IP address** to the IP address of the SIP Proxy server *40.40.50.1:5060* in this example.
- Set the **Destination Phone Numbers** *5552[0000-]* in this example.
- Click the **SIP** tab and set the **Server Address** to the IP address of the SIP Proxy server or the IP address of the DUT (private side) depending on the type of DUT.
- Click the **Audio** tab and ensure that the **Enable audio on this activity** check box is selected; this activity will play and receive audio on the established calls.

Setting the SIP_Caller Activity Parameters

1. Click the **Callee** activity under **SIP_Callee** NetTraffic.
2. Click the **Scenario** tab, and check whether the call flow is the required one. In this example, the call flow is: Register, Receive Call, Play Audio Streams, and Wait End call. Refer the Black Book sections describing how to create and change the call flow in the Scenario Editor or the IxLoad VoIP User Guide for instructions of how to modify the call flow.
3. Click the **Dial Plan** tab and set the **Source Phone Numbers** *5552[0000-]* in this example, this sequence should match the one specified as **Destination Phone Numbers** for the **SIP_Caller** activity.
4. Click the **SIP** tab and set the **Server Address** to the IP address of the SIP Proxy server or the IP address of the DUT (private side) depending on the type of DUT.
5. Click the **Audio** tab and ensure that the **Enable audio on this activity** check box is selected; this activity will play and receive audio on the established calls.

Setting the SIP_Proxy Activity Parameters

SIP_Proxy is an activity of type *SIP Cloud*. It listens on the specified port (5060 in this example) and dispatches the incoming SIP messages to one of the SIP Peer activities mapped to it.

1. Click the **SIP_Proxy** activity under **SIP_Proxy** NetTraffic.
2. In the **Settings** tab, set the **IP Address** to the network range IP-R3 – the one with **Distribution Rule** *Round Robin* – see the section “Setting the SIP_Caller activity parameters”. The UDP port is *5060* – change it if needed.

traffic2 - SIP_Proxy (VoIPSipCloud)

Settings Preview Cloud Traffic

IP Preference
 Only IPv4 Only IPv6

Simulated SIP Servers					
Name	IP Address	Range Type	IPv4/6	Port	Domain name
sip_server#1	Network Range IP-R3 in SIP_Proxy (40.40.50.1+1]	IP	IPv4	5060	ixload.com

Figure 169. SIP_Proxy activity (the Cloud) settings

Setting the R_Caller Activity Parameters

To emulate the Registrar server, the IxLoad conjunction has to contain a SIP Peer activity waiting for SIP Register messages and send the proper response. In addition, this activity extracts the SIP_Contact information from the incoming Register message and passes it to the activity that handle the call setup. There are two activities for Registrar emulation, one for each set of SIP User Agent: one for SIP Callers and one for SIP Callees. Each activity has its dial plan match the dial plan of the SIP User Agents.

1. Click the **R_Caller** activity under **SIP_Proxy** NetTraffic.
2. Check the Scenario; it contains a procedure waiting for SIP Register and sending the response 200 Ok. It can be modified to emulate the Register with authentication enabled by adding script objects (functions) to send 401 Unauthorized and wait for the second Register.
3. Click the **Dial Plan** tab and set the **Source Phone Numbers** `5551[0000-]` in this example; this sequence should match the one specified as **Source Phone Numbers** for the **SIP_Caller** activity; this is the criteria for matching the incoming SIP Register messages to this activity.
4. Click the **Cloud** tab and ensure that the **Enable SIP Cloud simulation using setting from** check box is selected.
5. Check whether the dispatching rule in **Override default dispatching rules** has the proper sequence of phone numbers in the **Formula for dispatching** field – `5551[0000-]` in this example.

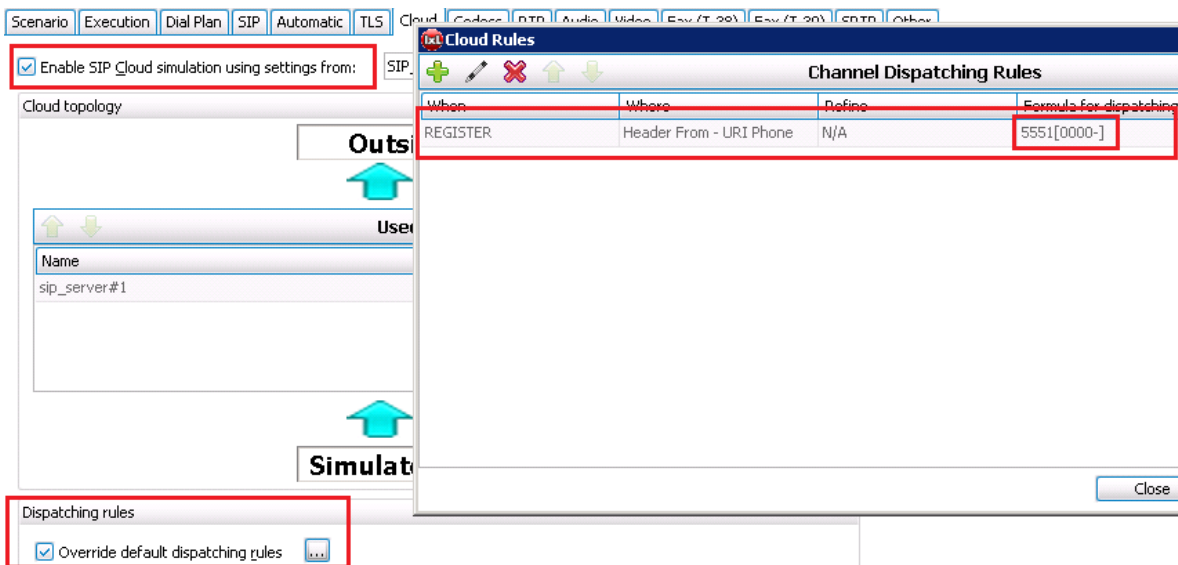


Figure 170. Cloud settings for R_Caller activity

Setting the R_Callee Activity Parameters

The activity handling the registrations of SIP Callee user agents is R-Callee.

1. Click the **R_Callee** activity under **SIP_Proxy** NetTraffic
2. Check the Scenario; it contains a procedure waiting for SIP Register and sending the response 200 Ok. It can be modified to emulate the Register with authentication enabled by adding script objects (functions) to send 401 Unauthorized and wait for the second Register.
3. Click the **Dial Plan** tab and set the **Source Phone Numbers** *5552[0000-]* in this example; this sequence should match the one specified as **Source Phone Numbers** for the **SIP_Caller** activity; this is the criteria for matching the incoming SIP Register messages to this activity.
4. Click the **Cloud** tab and ensure that the **Enable SIP Cloud simulation using setting from** check box is selected.
5. Check whether the dispatching rule in **Override default dispatching rules** has the proper sequence of phone numbers in the **Formula for dispatching** field – *5552[0000-]* in this example.

Setting the Wait_Call Activity Parameters

The calls between SIP User Agents **Caller** and **Callee** via the SIP Proxy, are composed by two call legs: one between **Caller** to Proxy and one from Proxy to **Callee**.

The SIP proxy server is emulated by two activities behind the Cloud: **Wait_Call** and **Make_Call**. **Wait_Call** terminates the first call leg from **Caller** and **Make_Call** originates the second call leg to **Callee**.

The two activities on the SIP Proxy (**Wait_Call** and **Make_Call**) communicate between them to synchronize the call flow and to pass parameters from one call leg to other, for example, destination phone number and SDP information.

1. Click the **Wait_Call** activity under **SIP_Proxy** NetTraffic.
2. Check the Scenario; it contains a set of procedures to receive the call, send the proper responses, and handle the end call sequence.
3. Click the **Dial Plan** tab and set the **Source Phone Numbers** *5552[0000-]* in this example; this sequence should match the one specified as **Destination Phone Numbers** for the **SIP_Caller** activity; this is the criteria for matching the incoming SIP Register messages to this activity.
4. Click the **Cloud** tab and ensure that the **Enable SIP Cloud simulation using setting from** check box is selected.
5. Check whether the dispatching rule in **Override default dispatching rules** has the proper sequence of phone numbers in the **Formula for dispatching** field – *5552[0000-]* in this example.

Setting the Wait_Call Activity Parameters

1. Click the **Make_Call** activity under **SIP_Proxy** NetTraffic.
2. Check the Scenario; it contains a set of procedures to originate the call and handle the end call sequence.
3. Click the **Dial Plan** tab and set the **Destination Phone Numbers** *5551[0000-]* in this example; this sequence should match the one specified as **Source Phone Numbers** for the **SIP_Callee** activity; this is the criteria for matching the incoming SIP Register messages to this activity.
4. Click the **Cloud** tab and ensure that the **Enable SIP Cloud simulation using setting from** check box is selected.
5. Check whether the dispatching rule in **Override default dispatching rules** has the proper sequence of phone numbers in the **Formula for dispatching** field – *5551[0000-]* in this example.

Setting the Timeline and Objective

This configuration contains two Activity Links: one for SIP User Agents Caller/Callee and one for SIP Registrar and Proxy Server. This Activity Links are independent, but the test objective for the server should be high enough to accommodate the level of traffic required by the test objective of the Caller/Callee Activity Link. In this example, the proxy is configured to handle 1,000 calls while the Caller/Callee Activity Link has a test objective of 100 Calls per Second with a maximum of 1.000 concurrent calls. For 1,000 calls, there are needed 2,000 endpoints (1,000 endpoints originating and 1,000 terminating calls). In this configuration (as in the majority of the IxLoad Voice configurations), the Objective Type *Channels* is equivalent with endpoint.

1. Click **Timeline & Objective** from the test configuration panel.
2. For the **VoIPLink2**, set the test **Objective Type** to **Channels**.
3. Set the test **Objective Value** to *1,000*.
4. On the **Timeline** tab, set the **Ramp Up Value** to *1,000*. This means all the resources on the emulated SIP Registrar and Proxy server become active in the first second of the test execution.
5. Set the **Sustain Time** to *1 hour*. The **Sustain Time** of this activity should be longer or at least equal with the sustain time of the Caller/Callee Activity Link to ensure that all the registration requests and calls initiated by the emulated SIP User Agents are properly answered by the server.
6. For the **VoIPLink1**, set the test objective to *Loops Initiated Per Second*. For this call flow, this is equivalent with Calls per Second, while the test scenario contains one call.
7. Set the test **Objective Value** to *1,000*.
8. On the **Timeline** tab, set the **Ramp Up Value** to *100*.

Test Case: Determining the Performance of a Session Border Controller

9. Set the **Sustain Time** to 1 hour.
10. In the **Custom Parameters** tab, select the **Specify Number of Channel** check box and set its value to 1,000.

Execute the Test

Map the ports and run the test.

Test Variables

Table 63.

Parameter Name	Current Value	Additional Options
IP Type	IPv4	IPv6
Type of traffic	Audio	Video, T38, Audio/Video RTCP
IP Mapping rules for User agents	N to 1	1 to 1, 1 to N, N to 1, and N to N
Number of users available	User defined	
Number of active calls	User defined	
Call Rate	User defined	
Call Duration	User defined	
Audio CODEC (type, packet size & frequency)	G.711u	G.711a, G.729, G.723, G.726, iLBC, AMR, any other codec using Custom Codec
Mix with data protocols (for example, FTP, HTTP, Telnet)	Not included	Any combination of data protocols supported by IxLoad
Mix of call flows		successfully calls, canceled calls, unanswered calls, busy calls
Mix of call features		call forward, call transferred, call hold/retrieve

Results Analysis

The DUT shall be monitored for the following:

- Memory size, memory allocation/de-allocation issues while:
 - Translations are added/deleted to/from the NAT table
 - Translations are continuously added and sessions are kept active
- CPU usage
- Size of NAT table

The following questions provide guidelines on how to recognize specific problems during or at the end of the test execution:

1. Has the test objective been achieved? Check the **Call Rates** view.

Test Case: Determining the Performance of a Session Border Controller

Table 64. Call Rate statistics

Statistic Name	Value	Questions
Calls Attempted per Second		1. Have the calls been attempted continuously at a constant call rate during the Sustain Time?
Calls Connected per Second		2. How do the Calls Attempted rate and the Calls Connected rate compare to each other?

2. Have any call failures been reported? Check the **Calls** view.

Table 65. Call statistics

Statistic Name	Questions
Calls Attempted	1. Have any call attempts failed? Compare: a. Calls Attempted and Calls Received, with b. Calls Attempted and Calls Connected.
Calls Connected	
Calls Received	
Calls Answered	
End Calls Received	
End Calls Completed	

3. Have any scenario loop failures been reported? Check the **Loops** statistics view.

Table 66. Statistics highlighting the pass/fail result based on call flow execution

Statistic Name	Questions
Total Loops	1. Are the Successful Loops and Total Loops values equal? 2. Have any Failed Loops, Aborted Loops or Warning Loops been reported? Note: failed/aborted and warning loops highlights failures at the scenario level.
Successful Loops	
Failed Loops	
Aborted Loops	
Warning Loops	

4. Have all messages received by cloud been dispatched correctly?

Table 67. Dispatched messages stats

Statistic Name	Questions
Undispatched Messages under SIP Cloud view	1. Is any SIP message received and not dispatched? Note: If there are Undispatched messages, the dispatching rules do not cover all the cases. The Dispatching Rules set on each activity should be checked; typically, the error is in mismatching of the phone numbers.

Test Case: Determining the Performance of a Session Border Controller

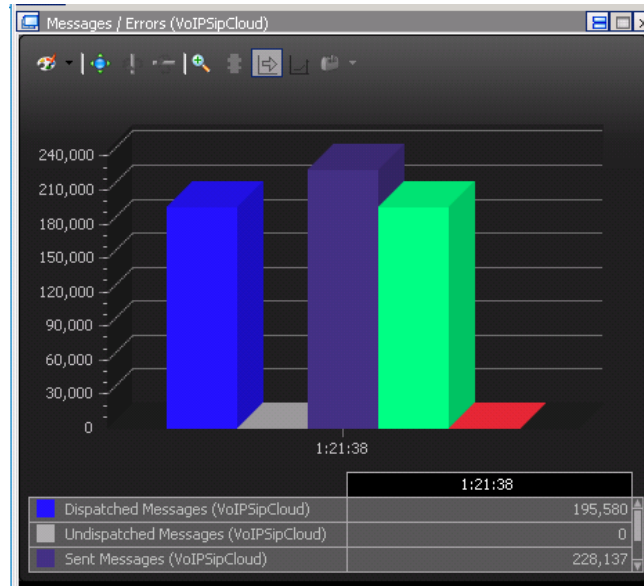


Figure 171. Dispatching SIP Messages Stats

- Has the QoS for signaling met the expected quality? Check the Call **Times** and **Delays** statistic views. Use the maximum value reported.

Table 68. Statistics used to determine the QoS for the SIP signaling

Statistic Name	Questions
Call Setup Time	1. Is the maximum Call Setup Time less than 4 seconds?
End Call Time	2. Is the maximum End Call Time less than 2 seconds?
Talk Time	3. Is the maximum Media Delay (Tx or Rx) less than 4 seconds?
Media Delay TX/RX	4. Is the maximum Post Dial Delay less than 2 seconds?
Post Dial Delay	5. Is the maximum Post Pickup Delay less than 2 seconds?
Post Pickup Delay	6. For all the stats listed in this table, compare their value distribution in time.

- In some cases, the DUT is not involved in the media path. In other cases, for example, when the SIP User agents are in different networks, the DUT handles the media as well. In this case, the following quality of media indicators should be checked: **RTP MOS RTP QoS**, **RTP Advanced QoS**, **RTP Jitter Distribution**, **RTP Consecutive Lost Datagram Distribution**, and **RTP Streams** statistic views.

Table 69. MOS statistics

Statistic Name	Questions
RTP MOS Best RTP MOS Worst	1. How do the last values reported by the RTP MOS Best and RTP MOS Worst compare with each other?
	2. How does the RTP MOS Worst score compare with the max theoretical score for the CODEC used?

Test Case: Determining the Performance of a Session Border Controller

RTP MOS Instant (Best/Avg/Worst)	<ol style="list-style-type: none"> 3. Are any times without an instantaneous MOS value? 4. How frequent are the changes in the instantaneous MOS values?
RTP MOS Per Call (Best/Avg/Worst)	<ol style="list-style-type: none"> 5. How do the MOS per Call statistics compare with the RTP MOS Best and RTP MOS Worst statistics?

Table 70. Basic RTP QoS statistics (see RTP QoS and RTP Advanced QoS statistics views)

Statistic Name	Questions
RTP Packets Sent RTP Packets Received RTP Packets Lost	<ol style="list-style-type: none"> 1. Are there any differences between RTP Packets Sent and RTP Packets Received? 2. Does the difference match the value of RTP Lost Packets?
RTP One Way Delay [us]	<ol style="list-style-type: none"> 3. Is the One Way Delay higher than 100 ms?
RTP Delay Variation Jitter [us] RTP Interarrival Jitter [us]	<ol style="list-style-type: none"> 4. What is the max Delay Variation Jitter? 5. What is the max Interarrival Jitter?

Table 71. RTP Jitter distribution statistics

Statistic Name	Questions
Packets with Delay Variation Jitter up to 1 ms	<ol style="list-style-type: none"> 1. Assuming Jitter was reported, what is the distribution of the Delay Variation Jitter values?
Packets with Delay Variation Jitter up to 3 ms	
Packets with Delay Variation Jitter up to 5 ms	
Packets with Delay Variation Jitter up to 10 ms	
Packets with Delay Variation Jitter up to 20 ms	
Packets with Delay Variation Jitter up to 40 ms	
Packets with Delay Variation Jitter over 40 ms	

Table 72.

Test Case: Determining the Performance of a Session Border Controller

Table 73. Distribution of RTP Consecutive Lost Packets

Statistic Name	Questions
Consecutive Loss of One Packet Sequence	1. Assuming that packet loss was reported, what is the distribution of the lost RTP packets?
Consecutive Loss of Two or Three Packet Sequences	
Consecutive Loss of Four or Five Packet Sequences	
Consecutive Loss of Six to Ten Packet Sequences	
Consecutive Loss of Eleven or More Packet Sequence	

Table 74. RTP Streams

Statistic Name	Questions
Concurrent RTP Streams	1. Assuming that packet loss was reported, what is the distribution of the lost RTP Packets?
Concurrent RTP Streams (max)	
Number of calls with incoming RTP packets	2. Are any calls without RTP? 3. Are any calls with RTP? 4. Does this number match the number of Calls Connected * 2?
Number of calls without incoming RTP packets	

Conclusions

This test methodology provided details of how to emulate with IxLoad Voice a SIP Registrar and Proxy Server to act as a Back to Back user agent. This configuration is presented in the context of measuring the performances of a Session Border Controller. The most challenging part of the configuration is setting correctly the dispatching rules for each activity.

Test Case: Measuring Quality of Experience for Voice Calls in LTE

Overview

With the migration of mobile networks to all IP networks defined by the LTE specification, there is a need to migrate the voice and SMS services as well. Today, there are several options for carrying voice over LTE, using different technologies:

- **CSFB, Circuit Switched Fall Back:** The circuit switched fallback, CSFB option for providing voice over LTE has been standardized under 3GPP specification 23.272. Essentially LTE CSFB uses a variety of processes and network elements to enable the circuit to fall back to the 2G or 3G connection (GSM, UMTS, CDMA2000 1x) before a circuit switched call is initiated. The specification also allows for SMS to be carried as this is essential for very many set-up procedures for cellular telecommunications. To achieve this, the handset uses an interface known as SGs which allows messages to be sent over an LTE channel.
- **SV-LTE - simultaneous voice LTE:** SV-LTE allows running packet switched LTE services simultaneously with a circuit switched voice service. SV-LTE facility provides the facilities of CSFB at the same time as running a packet switched data service. This is an option that many operators will opt for. However it has the disadvantage that it requires two radios to run at the same time within the handset. This has a serious impact on battery life.
- **VoLGA, Voice over LTE via GAN:** The VoLGA standard was based on the existing 3GPP Generic Access Network (GAN) standard, and the goal was to enable LTE users to receive a consistent set of voice, SMS (and other circuit-switched) services as they transition between GSM, UMTS, and LTE access networks.
- **Voice over LTE, VoLTE (initially called One Voice):** The Voice over LTE, VoLTE aims for providing voice over an LTE system utilizes IMS enabling it to become part of a rich media solution.

One additional approach which is not initiated by operators is the usage of Over-the-top (OTT) content services, using applications like Skype and Google Talk to provide LTE voice service. However, handing the LTE voice service over completely to the OTT actors is expected to not receive too much support in the telecom industry while the voice call service is, and will still be, the main revenue source for the mobile operators.

The typical topology for VoLTE is shown in the VoLTE Topology. The SIP registration and call control messages are sent from the User Endpoint (UE) over the default bearer in EPC to the Proxy Call Session Control Function (P-CSCF), the entry point in the IMS domain. In some networks an Session Border Controller (SBC) is used for this function. The Serving Call Session Control Function (S-CSCF) is the central node of the signaling plane. It is a SIP server that communicates to the Home Subscriber Server to download the users' profiles. S-CSCF controls over the Mx or Mg interfaces the Media Server and Media Gateway for voice routing.

Test Case: Measuring Quality of Experience for Voice Calls in LTE

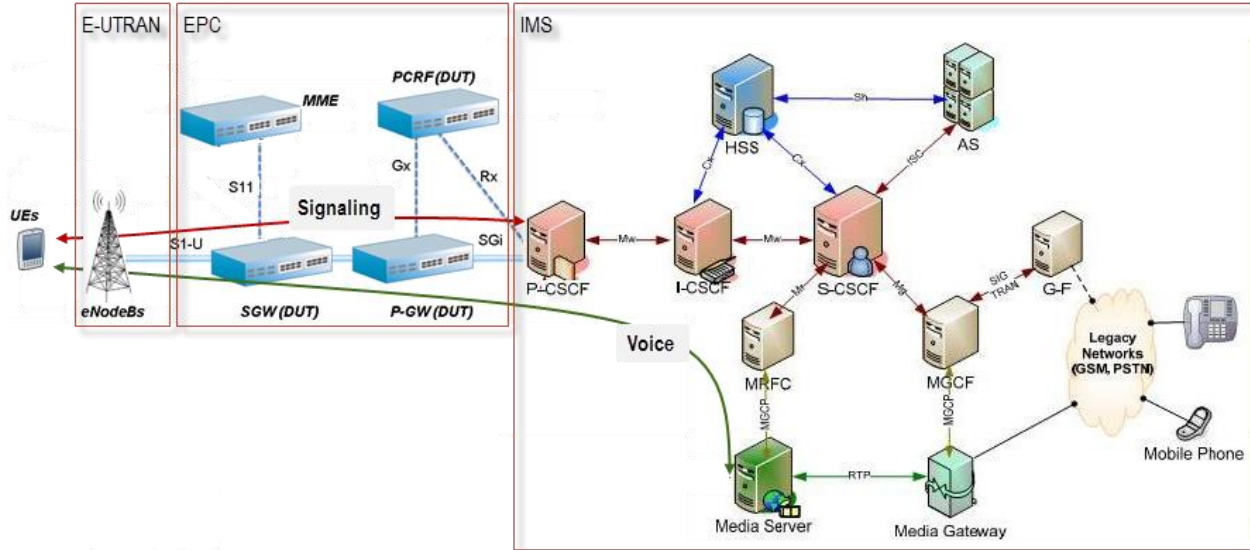


Figure 172. VoLTE Topology

To assure a good quality of voice, a dedicated bearer with high QoS is used for conversational speech in the EPC domain. The allocation of the dedicated bearer is requested by the P-CSCF to the Policy and Charging Rules Function (PCRF) over the Rx interface (this is a Diameter interface).

SMS-over-IP is also a functionality specified by VoLTE. The UE submits a short message via a SIP MESSAGE request that follows the same path to the S-CSCF; from here, depending on the user profile (obtained by S-CSCF from HSS) the SIP request is sent to the IP-SM-GW (IP Short Message Gateway); for simplicity this server is not represented in the topology shown in VoLTE Topology figure. The submission report is sent by the IP-SM-GW to the UE as a SIP MESSAGE Request. The SMS submit and submission report requests use the same SIP Method, but with different content-body.

Objective

The goal of this test methodology is to determine the capacity of the EPC to handle a specific volume of calls without degradation of voice quality.

Setup

The EPC isolation test configuration will be used for this test, where IxLoad will emulate:

- the User Endpoints over eNodeBs and MME (the left side of the topology diagram in the below figure,
- the IMS network; that is the P-CSCF and all the devices behind it

Voice calls will be originated from the UEs (eNodeB /MME) and terminated by user agents behind in the IMS network.

Test Case: Measuring Quality of Experience for Voice Calls in LTE

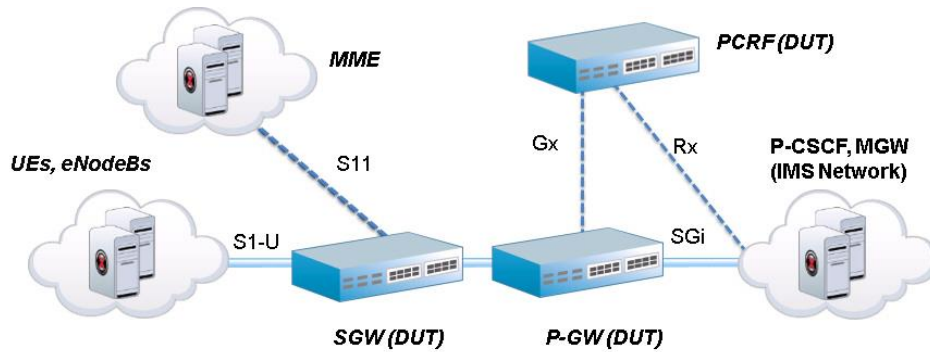


Figure 173. EPC Isolation Test Configuration

Step-by-Step Instructions

The step-by-step instructions highlight how to set the essential parameters of this configuration and explain additional options, which may be used to change the behavior of the test.

The final IxLoad configuration as a result of these steps is provided on the blackbook.ixiacom.com Web site - see *VoIP-VoLTE-IxLoad6.0.crf*. The configuration consists in two files: the test repository, a file with .rx extension, and the test scenario for the SIP call flow, a file with the .tst extension. These two files are archived in a single .crf file. To import a Compressed Repository File (.crf) in IxLoad, use the command **Import** under the **File** menu.



Figure 174. Option to Import a Compressed Repository File into IxLoad

Create the Network Traffic for UEs

1. Open the IxLoad GUI
2. Add the Originate **Net Traffic**

Test Case: Measuring Quality of Experience for Voice Calls in LTE

The menu is context sensitive; to access the Add Net Traffic button you have to select first the Networks and Traffic in the navigation pane.

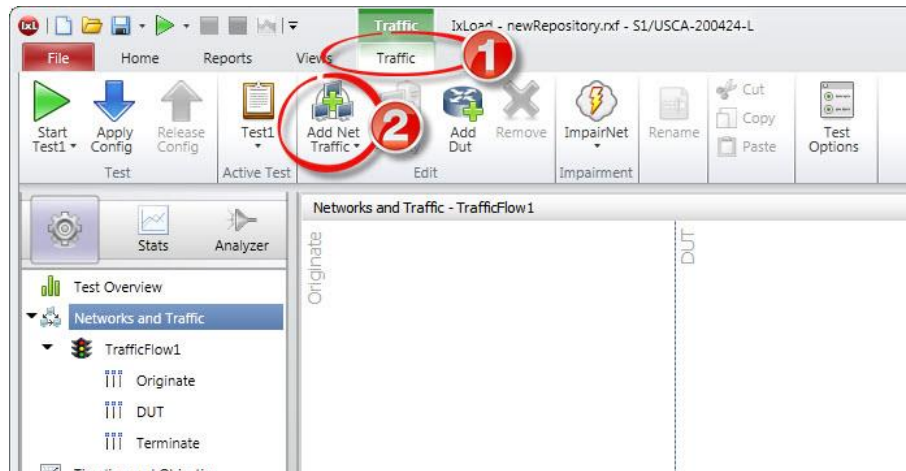


Figure 175. Add Net Traffic

3. Add the **MME/eNB S11/S1-U** Stack Manager interface to the Originate Net Traffic.

By default the Net Traffic added in step 2 contains a range of IPv4 addresses. The intention in this configuration is to emulate User Endpoints over MME and eNodeB that are connected through S11 and S1-U interfaces to the S-Gateway. These interfaces are provided by the MME/eNB S11/S1-U stack manager component. Select the **Network1**, right click on the **IP-1**, select **Add above** and then **MME/eNB S11/S1-u**

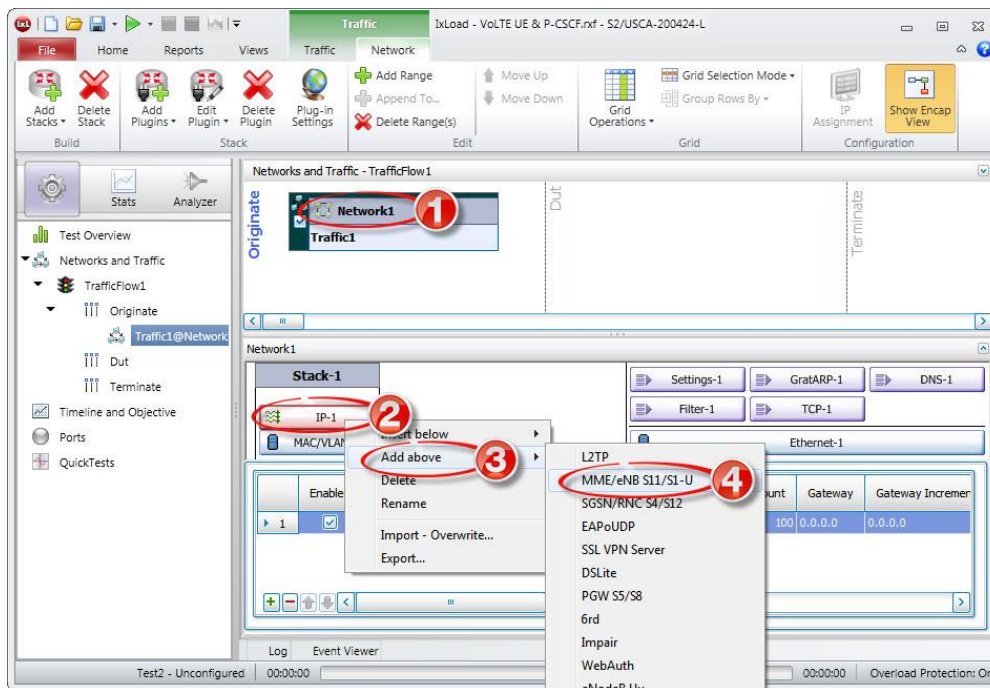
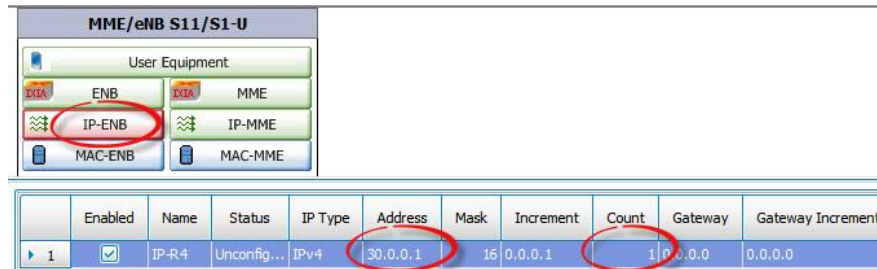


Figure 176. Add MME/eNB S11/S1-U Stack

Test Case: Measuring Quality of Experience for Voice Calls in LTE

4. Configure the IP addresses for the emulated eNodeB and emulated MME:
 - a. One eNodeB with the IP address 30.0.0.1 is emulated by this configuration. Select the **IP-ENB** element and set the **Address** and the **Count**

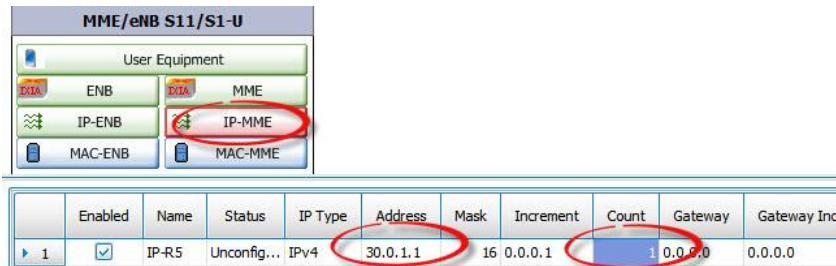


The screenshot shows the configuration interface for MME/eNB S11/S1-U. Under the 'User Equipment' section, the 'IP-ENB' element is selected and circled in red. Below this, a table lists the configuration for the selected element.

	Enabled	Name	Status	IP Type	Address	Mask	Increment	Count	Gateway	Gateway Increment
1	<input checked="" type="checkbox"/>	IP-R4	Unconfig...	IPv4	30.0.0.1	16	0.0.0.1	1	0.0.0.0	0.0.0.0

Figure 177. Setting the eNodeB IP address

- b. One MME with the IP address 30.0.1.1 is emulated by this configuration. Select the **IP-MME** element and set the **Address** and the **Count**



The screenshot shows the configuration interface for MME/eNB S11/S1-U. Under the 'User Equipment' section, the 'IP-MME' element is selected and circled in red. Below this, a table lists the configuration for the selected element.

	Enabled	Name	Status	IP Type	Address	Mask	Increment	Count	Gateway	Gateway Inc
1	<input checked="" type="checkbox"/>	IP-R5	Unconfig...	IPv4	30.0.1.1	16	0.0.0.1	1	0.0.0.0	0.0.0.0

Figure 178. Setting the MME IP address

Test Case: Measuring Quality of Experience for Voice Calls in LTE

5. Configure the emulated User Equipment parameters:

- a. the APN (Access Point Name) must match the name configured on the SUT; for this example the APN is set to *apn-1.test.com*
- b. the IP address of the PGW to which the APN refers must be set in the PGW IP; for this case the real PGW has the IP address 22.0.0.1.

Select the User Equipment element and, in the Access Points tab, set the APN name and the PGW IP.

Note: The PGW IP address can be resolved by DNS query if the option **Resolve DNS** is enabled in the MME DNS tab; in that case there is no need to fill the PGW IP field.

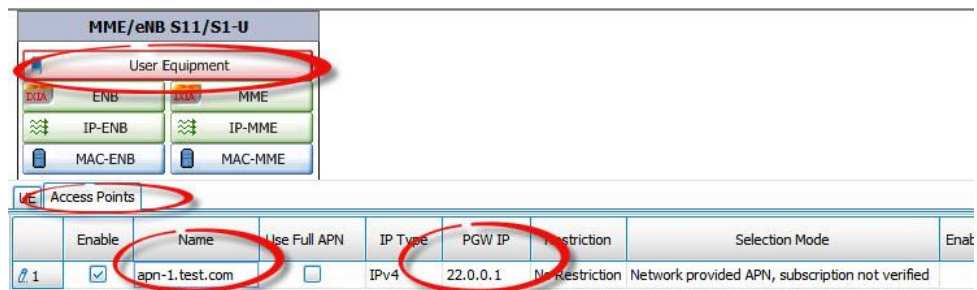


Figure 179. User Equipment Settings

Create the Network Traffic for Emulated P-CSCF

6. Add a Net Traffic on the Terminate side

Click on Terminate in the **Navigation Pane** or in the Terminate column in the **Networks and Traffic** pane; then click on the Add Net Traffic button in the Toolbar, or right click and select Add Net Traffic

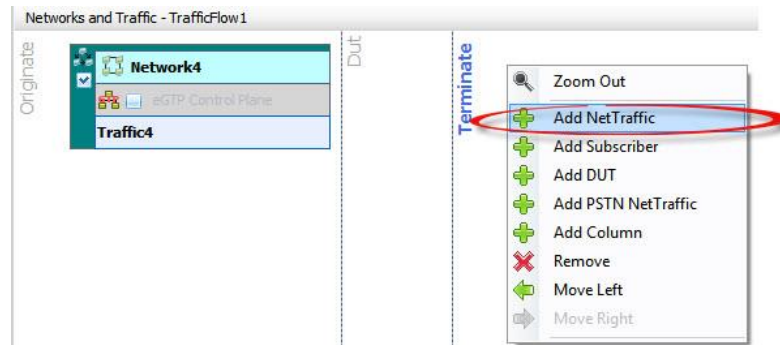


Figure 180. Add Net Traffic by right click on the Network and Traffic pane

Test Case: Measuring Quality of Experience for Voice Calls in LTE

7. Set the IP address of the emulated P-CSCF

One P-CSCF server is emulated in this configuration. Set its IP Address to 22.22.22.1 and the Count to 1

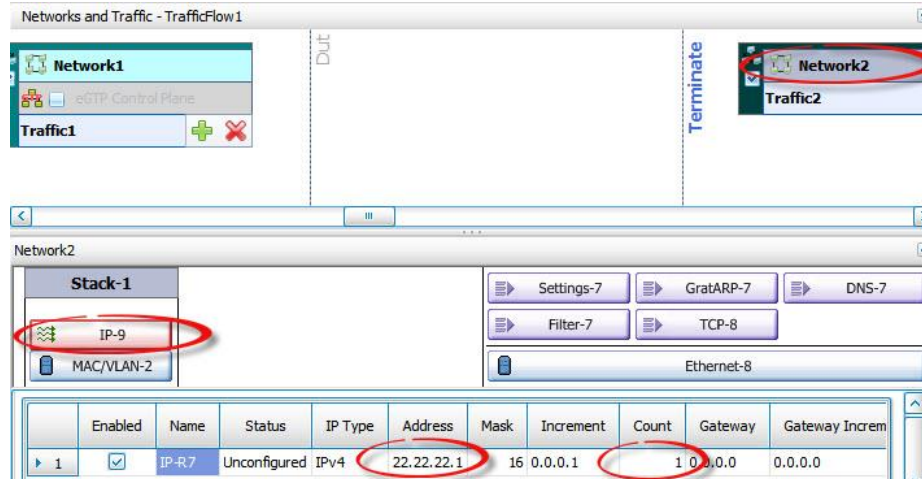


Figure 181. IP Address of the emulated P-CSCF

Add the Emulated P-CSCF

8. Add the SIP Cloud activity

Move the mouse over the Traffic element; a plus sign will be shown; click on it and select the VoIPSIP Cloud activity

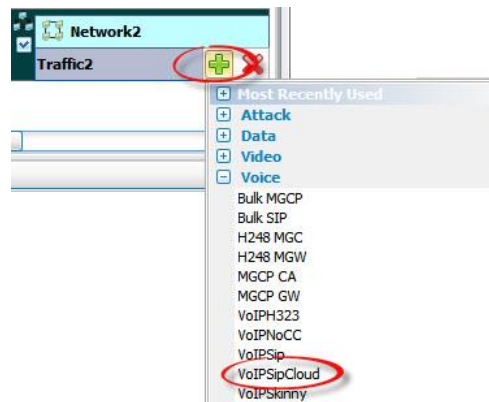


Figure 182. Add a SIP Cloud activity - this is the emulated P-CSCF

- Set the Distribution Group of the IP address range to IP Round Robin

The SIP Cloud requires that the distribution of the network ranges by port to be set to IP Round Robin.

Select the Traffic element in the Net Traffic, select the IP Mappings tab, double click on the Distribution Group and select the IP Round Robin distribution type.

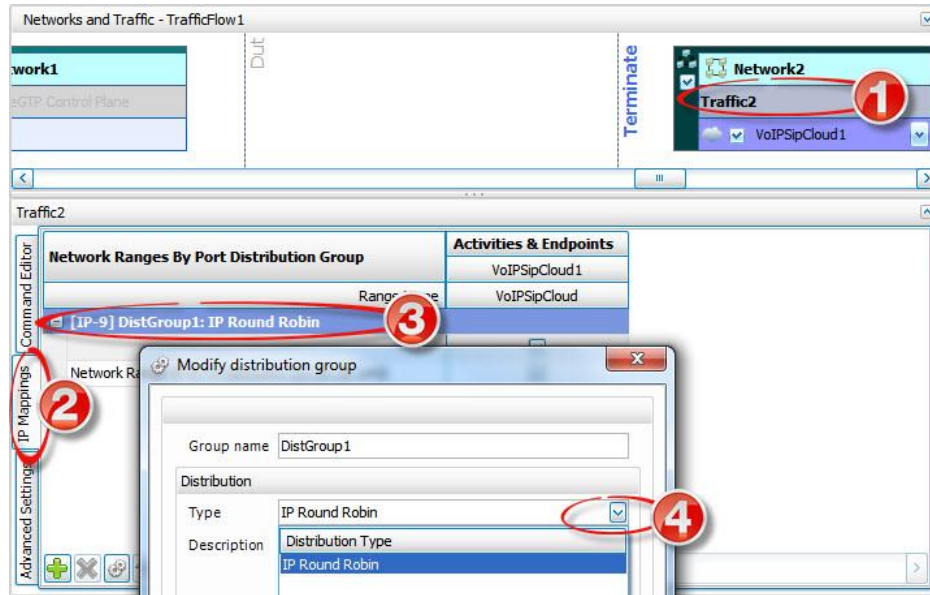


Figure 183. Distribution Type of Network Ranges

Add and Configure the SIP Peer Activities

In this moment the configuration contains all the network elements. The next steps will be to add the definition of the application traffic. The intention of this configuration is to emulate SIP UEs on the eNodeB/MME side that originate calls routed by the emulated P-CSCF to emulated SIP endpoints placed behind the P-CSCF. This will be emulated by adding a SIP activity to originate calls on the **MME/eNB S11/S1-u** network and a SIP activity to receive calls under the **SIP Cloud**.

10. Add a SIP peer activity on the Originate side

Move the mouse over the Traffic element of the Originate network; a plus sign will be shown; click on it and select the VoIPSIP Peer activity.

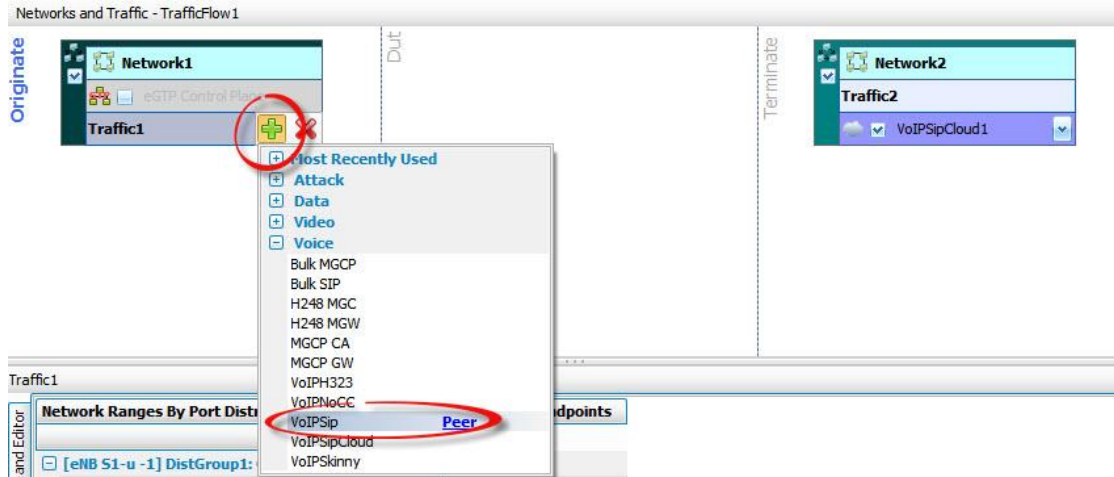


Figure 184. Add the VoIPSIP Peer activity

11. Add a SIP peer activity on the Terminate side

Repeat the operation of adding a SIP peer activity on the terminate Net Traffic



Figure 185. SIP Peer activities on both Originate and Terminate Network Traffics

Test Case: Measuring Quality of Experience for Voice Calls in LTE

12. Use one of SIP Sample message flows as the call flows for VoIPSIPPeer1 and 2 activities.

Drag and drop the lollipop of the VoIPSIPPeer1 over VoIPSIPPeer2. Select the SIP EP – Registration and Call with Voice message flow from the drop down list.



Figure 186. Add SIP Call flow to the SIP activities

13. Set the parameters of VoIPSIPPeer1 activity

Select the VoIPSIPPeer1 activity

a. Set the Dial Plan

In this example we'll emulate 10 UEs with Phone Numbers 1001 to 1010. Select the Dial Plan tab and edit the sequence [1001-1010] in the Source / Phone Numbers field of the Dial Plan. Leave the Destination as the Symbolic Link to the Terminate VoIP activity; the phone numbers defined there will be used in building the SIP Invite requests.

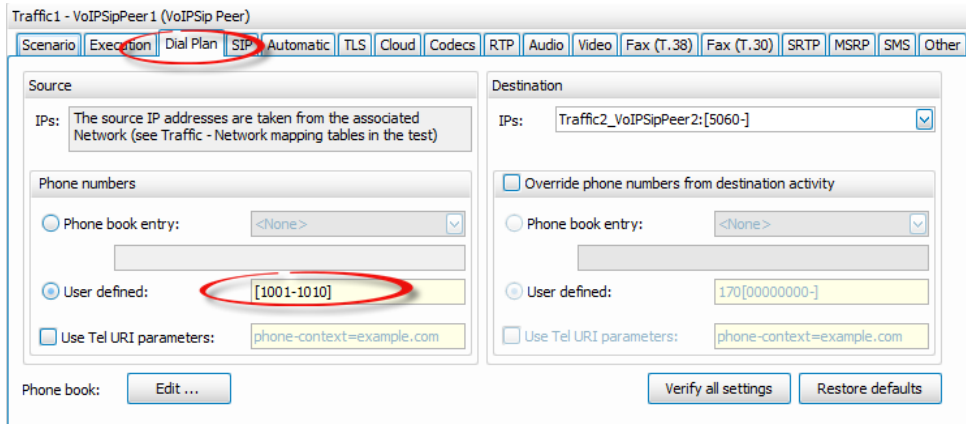


Figure 187. Dial Plan

Test Case: Measuring Quality of Experience for Voice Calls in LTE

b. Set the SIP Server Address

The UE needs to know the IP address of the P-CSCF; it can be configured as an IP address or as a domain name in the SIP Tab. In this case, while the P-CSCF is the entry point to the IMS network, the IP address of the registrar server is the same. The P-CSCF knows how to route the registration and call setup requests, but for the UEs this is transparent; it will send all the SIP requests to the same P-CSCF.

Select the **SIP** Tab, enable **Use External Server** checkbox, set the **Server address** and **Domain name or local IP** to the IP address of the P-CSCF (22.22.22.1 in this example), and enable the **Registrar server** checkbox.

Traffic1 - VoIP SIP Peer 1 (VoIP SIP Peer)

Scenario | Execution | Dial Plan | **SIP** | Automatic | TLS | Cloud | Codecs | RTP | Audio | Video | Fax (T.38) | Fax (T.30) | SRTP | MSRP | SMS | Other

Enable signaling on this activity SIP Port: [5060-]

(if unchecked, all SIP script functions will be SKIPPED)

Transport settings

Maximum message size on UDP: 1024

Override transport specified in scenario: UDP Only

TCP send immediate

Enable FQDN resolution

Authentication UAC

User name: user0[001-010]

Password: 12345

AKA authentication settings

Select configuration: <None>

Edit configurations...

Type Of Service

TOS/DSCP: Best Effort (0x00)

Use external server

Server address: 22.22.22.1

Server port: 5060

Domain name or local IP: 22.22.22.1

Outbound proxy

Registrar server

Auto register simulated user agents

Override registrar IP:PORT

Construction of SIP messages

Override default contact settings Edit Contact ...

Override default destination domain name or host:port

Domain name or Host:Port:

Use Tel URI scheme for Source

Use Tel URI scheme for Destination

Transfer address: Edit ...

Verify all settings Restore defaults

VoLTE settings

Enable Call Control of dedicated bearer

Figure 188. Set the SIP Server IP address

Test Case: Measuring Quality of Experience for Voice Calls in LTE

c. Set the Authentication credentials

In the real deployments the SIP requests for registration or call setup are authenticated by the server. The UE has to use the proper credentials to have access to the service. Several authentication methods are defined for SIP, IxLoad supporting all of them. In this example we'll use digest MD5, that requests each user to have a username and a password. We will use the sequence user0001 to user0010 for the username and the same password 12345 for all users; it is possible also to use a sequence for the password also. In cases when the phone numbers, usernames or passwords of the emulated UEs are not in sequence, a phonebook can be used. In the SIP tab edit the Username and Password fields.

The screenshot shows the configuration window for a VoIP SIP Peer in IxLoad. The 'SIP' tab is selected and highlighted with a red circle. The 'Authentication UAC' section contains the following fields:

- User name:** user0[001-010] (highlighted with a red oval)
- Password:** 12345 (highlighted with a red oval)

Other visible settings include:

- Enable signaling on this activity:** checked
- SIP Port:** [5060-]
- Use external server:** checked
- Server address:** 22.22.22.1
- Server port:** 5060
- Domain name or local IP:** 22.22.22.1
- Registrar server:** checked
- Override registrar:** IP:PORT

Buttons at the bottom include 'Verify all settings', 'Restore defaults', and 'Edit ...'.

Figure 189. Authentication Credentials

Test Case: Measuring Quality of Experience for Voice Calls in LTE

d. Set the Audio Codec to AMR-WB

VoLTE requires AMR codec for the speech communication. This is a multi-rate codec optimized for speech with capability to adapt to variations of network conditions. IxLoad supports both AMR-NB and WB versions.

Select the Codecs tab; in the table of audio codecs, select the first one, and choose AMR-WB codec from the drop down list. You have the option to change the order of preferred codecs. The SDP will be automatically built with the parameters configured in this page.

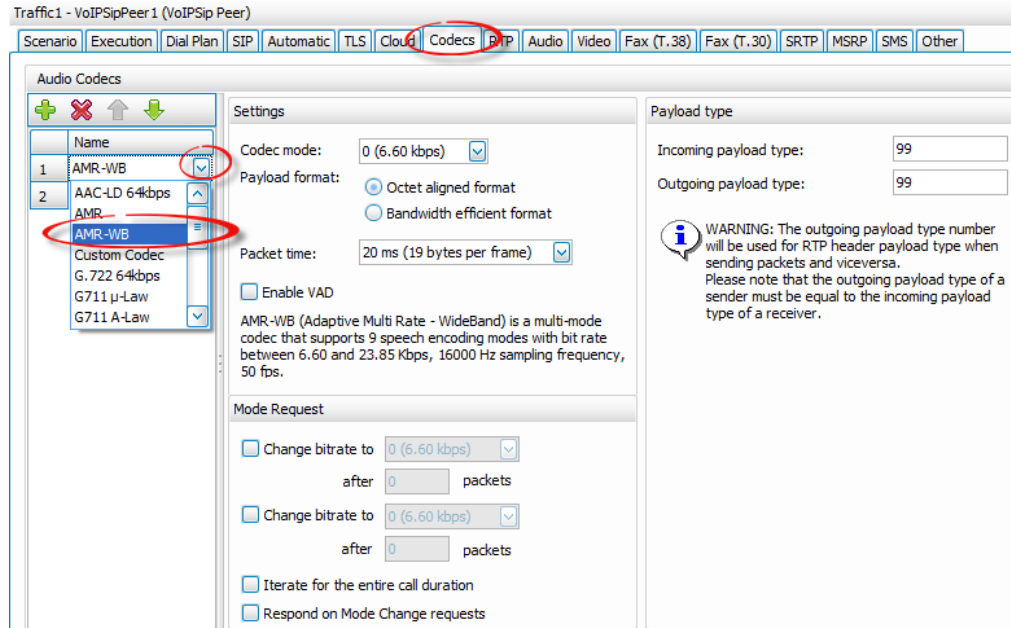


Figure 190. Set the AMR-WB as preferred codec

Test Case: Measuring Quality of Experience for Voice Calls in LTE

e. Set the Audio parameters

To have media during an established SIP call, the call flow must contain a media function and that type of media to be enabled on the activity. The call flow contains the function Voice Session that plays and listen audio clips in the same type; it implements the full duplex audio functionality. On the Audio tab, the Audio Activity has to be enabled. In the same page we can specify the duration of the voice session and implicit the duration of the call (assuming only voice session function is executed during the call).

Select the **Audio** tab, enable the checkbox **Enable audio on this activity**, set the **Play for** to 30 seconds, and enable the **Perform MOS** checkbox.

Traffic1 - VoIP SIP Peer 1 (VoIP SIP Peer)

Scenario Execution Dial Plan SIP Automatic TLS Cloud Codecs RTP Audio Video Fax (T.38) Fax (T.30) SRTP

Enable audio on this activity (if unchecked, all audio script functions will be SKIPPED)

Play Settings

Clip: US_042.wav

Format: PCM, Duration: 32785 ms, Size: 524556 bytes

Output level: -20 dBm

Play for clip duration or TalkTime (all objectives except Channels)

Play for: 30 Seconds

Type Of Service

TOS/DSCP: Class 1 (0x20)

Perform MOS Calculate One Way Delay

Enable jitter buffer

Buffer size: 20 ms

Use compensation

Max. size: 1000 ms

Max. dropped consecutive packets: 7

Perform QoS

Units: # of Channels

Value: 100

Channel Selection: First Channels

Generate silence

Null data encoded Comfort noise

Verify all settings Restore defaults

Figure 191. Audio Settings

14. Set the parameters of VoIPSipCloud1 activity

Select the VoIPSipCloud1 activity

- a. Map the SIP cloud activity to the IP address defined for the Terminate Network

In the Settings tab select the IP address from the list. Only the Network Ranges with the Distribution Group Round Robin are shown.

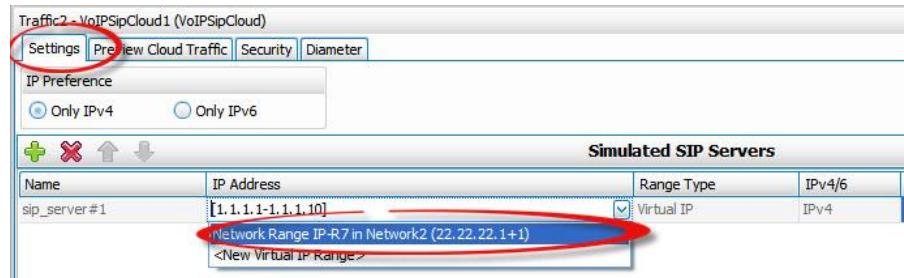


Figure 192. Map the SIP cloud to an IP address

- b. Enable the Rx interface on the emulated P-CSCF

The P-CSCF and the PCRF are connected through the Rx interface – see **Error! Reference source not found.** The interface is used to confirm that call media requests conform to the appropriate policy, open gates and pinholes in the media route, specify the appropriate QoS, request per-flow charging information when needed, and inform the P-CSCF of media-plane events. The Rx interface uses the Diameter protocol and can be emulated by the SIP Cloud module in IxLoad.

Test Case: Measuring Quality of Experience for Voice Calls in LTE

In the **VoIP SIP Cloud1** activity select the **Diameter** tab, and check the **Enable Rx Interface** option

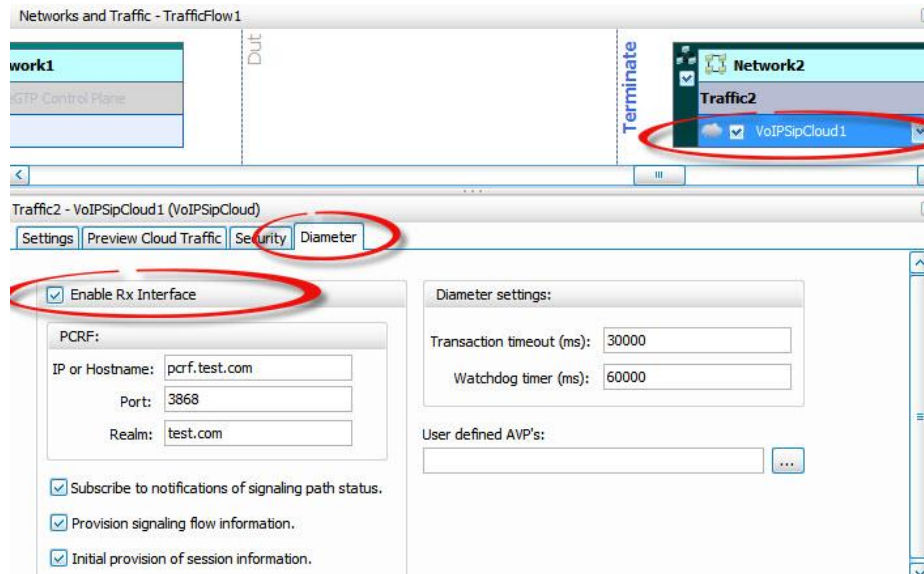


Figure 193. Enable the Rx interface for the emulated P-CSCF

c. Set the PCRF parameters

The P-CSCF needs to communicate with the PCRF; the IP address of the Hostname and the realm of the PCRF are parameters of the Diameter configuration page. In this example the PCRF hostname is pcrf.test.com, and the realm is test.com (these are configuration parameters of the SUT)

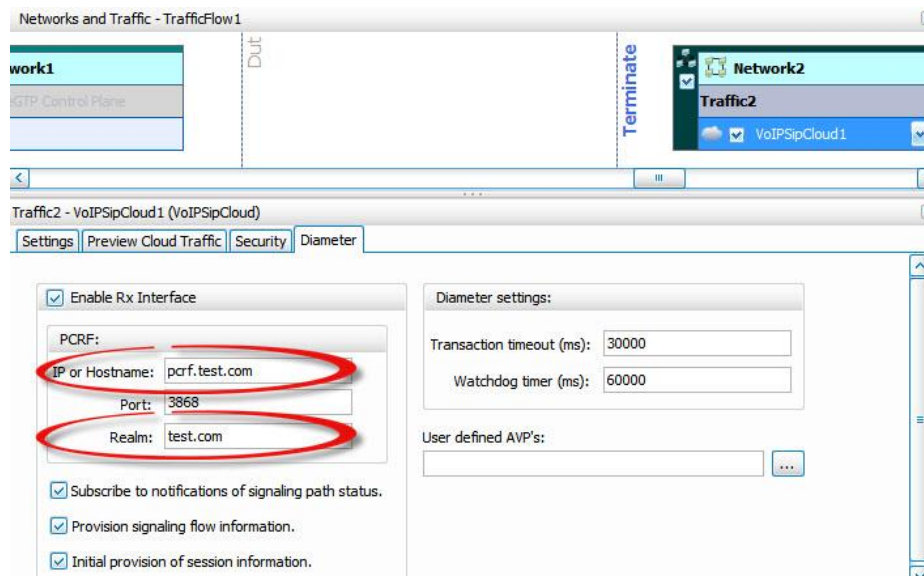


Figure 194. PCRF parameters

Test Case: Measuring Quality of Experience for Voice Calls in LTE

Note: the PCRF implementations have variations in supported AVPs (Attribute Value Pair). To interoperate with a specific PCRF, you may need to use custom AVPs in the Diameter messages on Rx interface. This is possible in IxLoad by loading a file with the description of custom AVPs (the field User defined AVPs allows this).

d. Set the SIP Security parameters on the Server

The SIP Cloud acts as the P-CSCF; a real P-CSCF does the authentication of the SIP requests. The emulated P-CSCF can do the same functionality. In the Security tab, the type of requests that need to be authenticated can be defined, as well as the Authentication algorithm and the credentials for each UE.

Select the **Security** tab, enable **MD5 Authentication Algorithm**, and disable **AKA Authentication**.

Click on Security Profiles Pool... button. Create a new profile; and make sure you set the same values for the Phone Number, Username, and Password as for the emulated UEs (on the originate side)

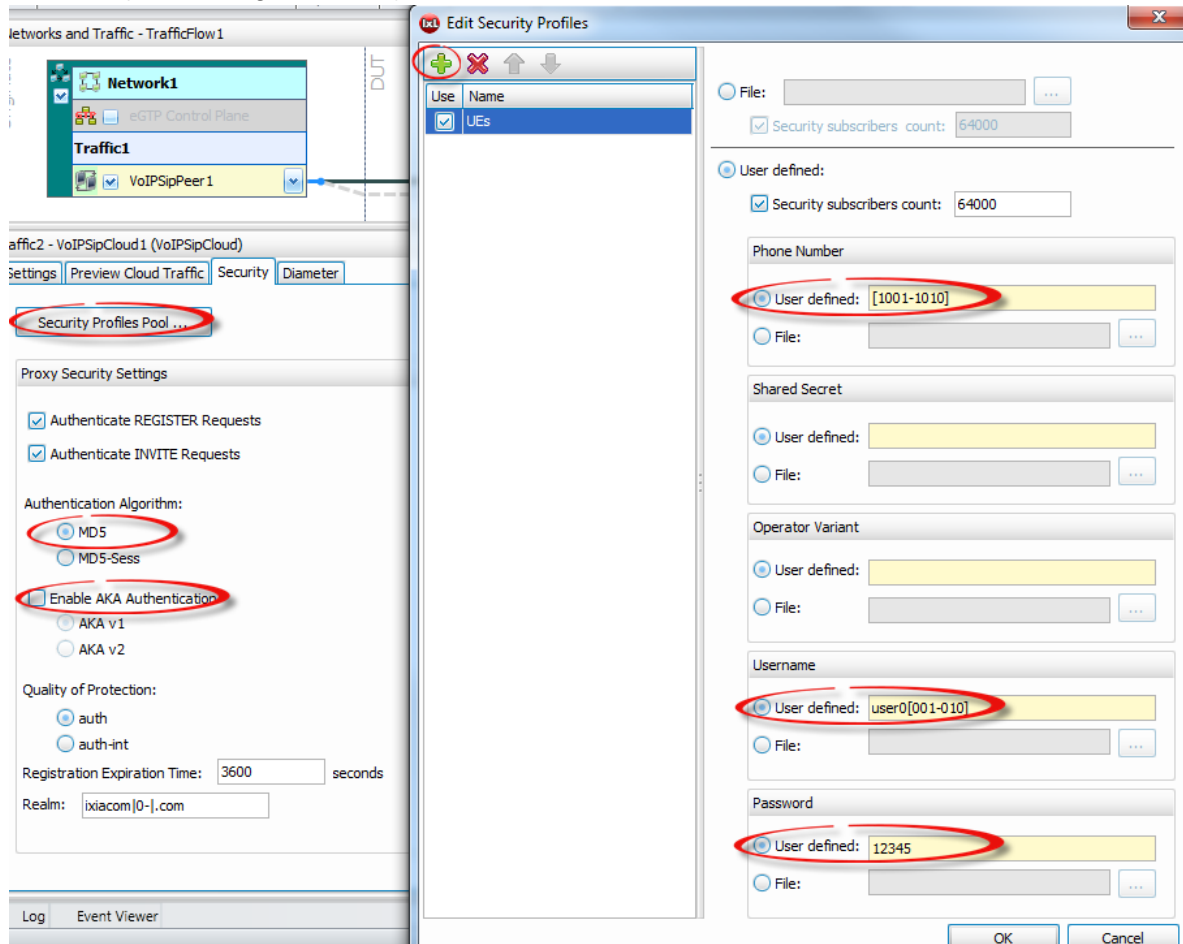


Figure 195. Security profile on the SIP Server

15. Set the parameters of VoIPsIpPeer2 activity

Select the VoIPsIpPeer2 activity

Test Case: Measuring Quality of Experience for Voice Calls in LTE

a. Set the Dial Plan

In this example we'll emulate 10 User Agents with Phone Numbers 2001 to 2010. Select the Dial Plan tab and edit the sequence [2001-2010] in the Source / Phone Numbers field of the Dial Plan. Leave the Destination as none; this activity is only receiving calls, so it does not need destination phone numbers.

Traffic2 - VoIPsipPeer2 (VoIPsip Peer)

Scenario Execution **Dial Plan** SIP Automatic TLS Cloud Codecs RTP Audio Video Fax (T.38) Fax (T.30) SRTP MSRP SMS Other

Source

IPs: The source IP addresses are taken from the associated Network (see Traffic - Network mapping tables in the test)

Phone numbers

Phone book entry: <None>

User defined: [2001-2010]

Use Tel URI parameters: phone-context=example.com

Destination

IPs: None

Override phone numbers from destination activity

Phone book entry: <None>

User defined: 170[00000000-]

Use Tel URI parameters: phone-context=example.com

Phone book: Edit ...

Verify all settings Restore defaults

Figure 196. Dial Plan

b. Enable this activity with the SIP Cloud

The VoIPsipPeer2 activity emulates SIP User Agents in the IMS core, behind the P-CSCF that is emulated by the SIP Cloud activity. That means the all the traffic between EPC and the IMS core goes through the P-CSCF (VoIPsipCloud1 activity) that will receive the SIP Requests, and will forward them to the User Agents behind it.

Test Case: Measuring Quality of Experience for Voice Calls in LTE

Select the **Cloud** tab, enable the **SIP Cloud simulation** checkbox, enable **Virtual IPs** checkbox, and a range of 10 **Virtual IPs** with the starting IP Address 23.23.23.1.

The screenshot shows the configuration page for a VoIP peer activity. The 'Cloud' tab is selected. The 'Enable SIP Cloud simulation' checkbox is checked and circled in red. The 'Used Simulated SIP Servers' table contains one entry: 'sip_server #1' with 'Use server' and 'Keep in route' both checked. The 'Enable Virtual IPs' checkbox is also checked and circled in red. The 'Virtual IPs' table has one entry: '23.23.23.1' with a mask of 16, an increment of 0.0.0.1, a count of 10, and a range type of 'Virtual IP'. The IP address '23.23.23.1' is circled in red.

Name	Use server	Keep in route
sip_server #1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

IP Address	Mask	Increment	Count	Range Type	IPv4/6
23.23.23.1	16	0.0.0.1	10	Virtual IP	IPv4

Figure 197. Associate a SIP Peer Activity with a SIP Cloud activity to emulate SIP User Agents behind a SIP Proxy

There is no need to set the SIP Server IP address or the Credentials for Authentication for this activity as it communicates internally to the emulated P-CSCF.

Test Case: Measuring Quality of Experience for Voice Calls in LTE

c. Set the Audio Codec to AMR-WB

Select the Codecs tab; in the table of audio codecs, select the first one, and choose AMR-WB codec from the drop down list. You have the option to change the order of preferred codecs. The SDP will be automatically built with the parameters configured in this page.

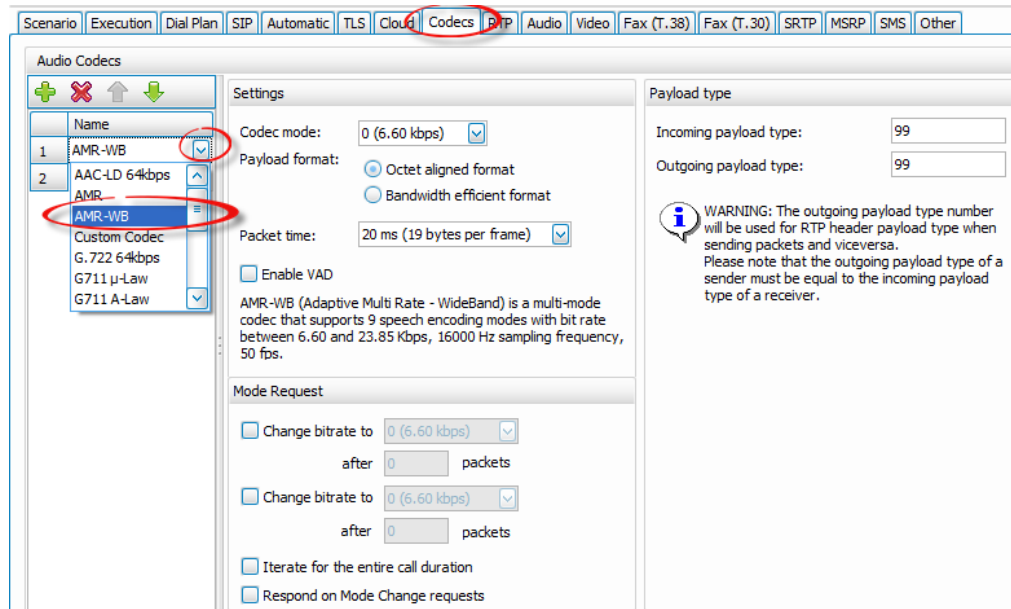


Figure 198. Set the AMR-WB as preferred codec

Test Case: Measuring Quality of Experience for Voice Calls in LTE

d. Set the Audio parameters

As on the originating side, the terminating side must have the Audio settings on to play and receive audio.

Select the **Audio** tab, enable the checkbox **Enable audio on this activity**, set the **Play for** to 30 seconds, and enable the **Perform MOS** checkbox.

The screenshot shows the 'Audio' configuration panel with the following settings:

- Enable audio on this activity (if unchecked, all audio script functions will be SKIPPED)
- Clip: US_042.wav
- Format: PCM, Duration: 32785 ms, Size: 524556 bytes
- Output level: -20 dBm
- Play for clip duration or TalkTime (all objectives except Channels)
- Play for: 30 Seconds
- Type Of Service
- TOS/DSCP: Class 1 (0x20)
- Perform MOS
- Calculate One Way Delay
- Enable jitter buffer
- Buffer size: 20 ms
- Use compensation
- Max. size: 1000 ms
- Max. dropped consecutive packets: 7
- Perform QoV
- Units: # of Channels
- Value: 100
- Channel Selection: First Channels
- Generate silence
- Null data encoded
- Comfort noise

Figure 199. Audio Settings

16. Set the Timeline and Objective

For the capacity test, a test objective type will be set to Channels. The specified number of channels will be concurrently active executing the call flow defined in the activities. Set the Objective Value to 10. If you want to increase the test objective, you will need to:

- Increase the number of Maximum Active UE Count in the User Equipment plugin under Network1
- Extend the sequence of Phone Numbers in the VoIPSIPPeer1 dial plan
- Extend the sequence of User names for the VoIPSipPeer1 Authentication
- Extend the sequence of Phone Numbers in the VoIPSIPPeer2 dial plan
- Increase the number of virtual IP Addresses for VoIPSipPeer2 activity (under Cloud tab).

Set the Sustain Time to 5 min.

17. Ports mapping

Map ports to the TrafficNetworks. You will need a pair of ports, connected to the EPC system (the DUT)

Test Case: Measuring Quality of Experience for Voice Calls in LTE

Running the test

18. Save the configuration

Save the configuration, using the Save button in Quick Access Toll Bar or the option Save under File or the keys Ctrl+S. If it's the first time you save the configuration you will be prompted to enter a name for the rxf configuration and then you'll be prompted to enter a name for the tst file. The .tst file contains the SIP call flow.

19. Run the test

Results Analysis

The EPC, SIP, and RTP stats need to be analyzed in this configuration.

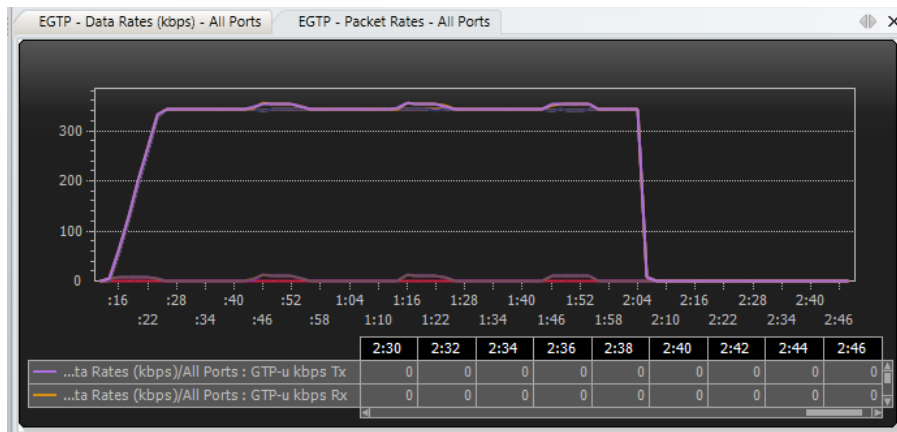


Figure 200. EPC Packet Rates

Test Case: Measuring Quality of Experience for Voice Calls in LTE

The Packet Rate increases when new calls are established, but majority of the packets are RTP.

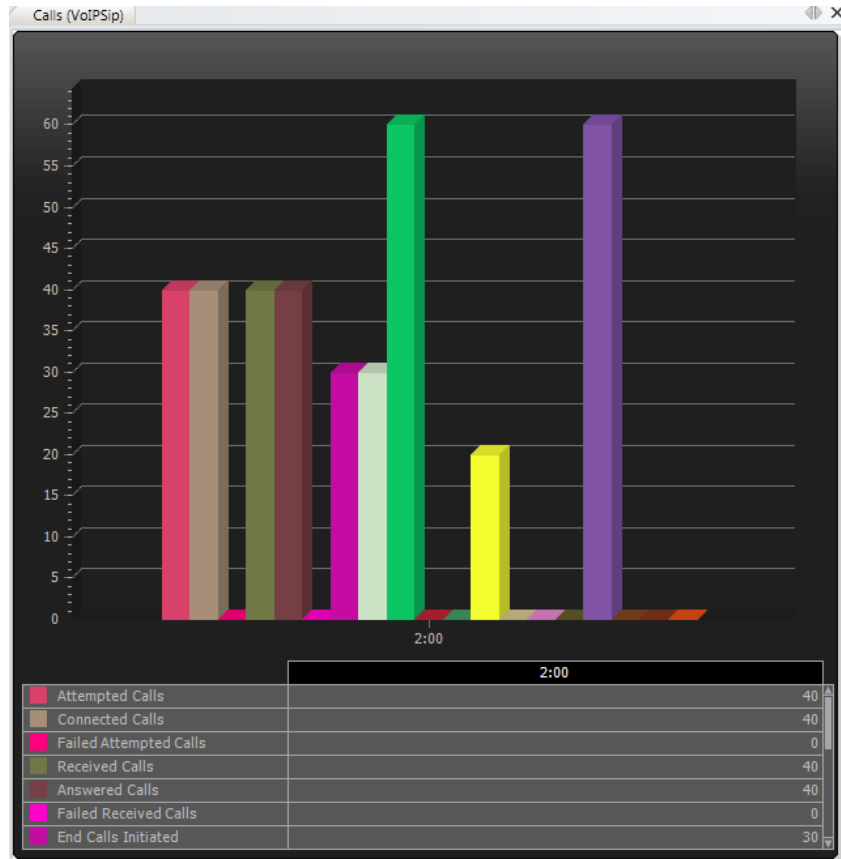


Figure 201. VoIP Calls

The number of Attempted / Connected calls (on the Originating side) and the number of Received / Answered calls (on the Terminating side) must be equal in a successful test. Any difference means the calls cannot be established; the reasons for failure can be identified in the Event Viewer.

RTP MOS (VoIP/Sip)		:40	:42	:44	:46	:48	:50	:52	:54	:56	:58	1:00	1:02	1:04	1:06	1:08
1	MOS Instant (Avg)	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820
2	MOS Instant Best	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820
3	MOS Instant Worst	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820
4	MOS Best	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820
5	MOS Worst	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820
6	MOS Per Call (Avg)					3.820	3.820	3.820	3.820	3.820						
7	MOS Per Call Best					3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820
8	MOS Per Call Worst					3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820	3.820

Figure 202. Quality of Voice

Quality of voice is the final goal of this test; even though the calls can be established, it is important to measure QoE for the voice service. In this case the measure MOS is as expected (for AMR-WB codec mode 0, the expected MOS is 3.82).

Troubleshooting and Diagnostics

When issues are present, a deep down analysis can be done using various features provided by IxLoad:

- Stats drilldown per activity / port
- Event viewer for SIP, SDP, and RTP related issues; when an error occurs on VoIP an error is logged in the Even Viewer window indicating the endpoint ID, the error type and description, and hints to resolve the error.
- Analyzer and Traffic Packet Viewer: you can enable traffic capture per port; to save memory the RTP outbound packets are not captured; to minimize the size of the capture you can apply filters using the tcpdump syntax.

Test Variables

Table 75.

Parameter Name	Current Value	Additional Options
IP Version	IPv4	IPv6
Concurrent Calls	10	Up to 8,000 concurrent active endpoints in calls with audio streams can be emulated by a single 1G port of an Xcellon-Ultra-NP card. To increase the number of channels, you have to allocate enough resources in terms of IP Addresses and Phone Numbers/User Names (see section 16 Set the Timeline and Objective)
Test Objective	Channels	<p>The configuration is created for capacity testing. Other test objectives are available: for testing the rate of call setup supported by the access network, the CPS test objective can be used. Beside the value of Call Per Second Rate, you have to specify the number of emulated endpoints or the call duration (the talk time); these three parameters are correlated:</p> $\text{CPS} = \text{Number_of_Endpoints} / \text{Call_Duration}$ <p>where</p> $\text{Call_Duration} = \text{Talk_Time} + \text{Call_Setup_Time} + \text{Overhead_Time}$ <p>To increase the CPS you have to use more endpoints or reduce the Talk Time.</p>

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

Overview

In the growing global business environment, telepresence has become an important tool for efficient communication between members while assisting time efficiency, decrease in travel costs, and minimizing carbon emission footprint. Telepresence represents a high-definition videoconferencing service, delivering a virtual face-to-face meeting experience without the costs and disadvantage of long-distance travels. The perception is that users across the globe can join a meeting as if they were all in the same room.

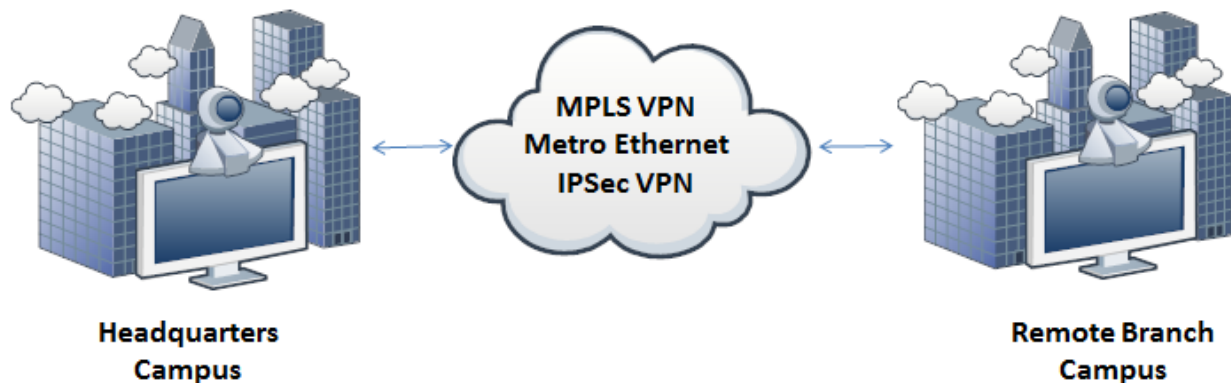


Figure 203. Business-to-business telepresence application

Telepresence from technical aspects represents a combination of technologies to deliver a complex videoconferencing solution. The challenge is to preserve a high quality experience for the entire call duration for an almost realistic user interaction. By default the teleconferencing solution uses one or more high resolution cameras, with additional video channel for presentation content as well as several audio channels originated by all the participating parties.

Network impairments effect acts differently on the telepresence media components, and it's necessary to provide the right level of QoS for the data flows. This chapter will not cover the network configuration aspects, as it will assist in configuring and determine the QoS for telepresence using IxLoad.

Video conferencing has played an important role in bringing additional value to communications by allowing more interaction between multiple participants. Key benefits include an increase in productivity, frequent and fruitful interaction between staff in different locations, a better information flow; and improved decision-making, information-sharing, and knowledge from specialists in different branches all in one delivering a balanced quality of life for the business environment.

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

There are several challenges that this technology brings from technical aspects such as propagation latency, device processing capabilities, and solution convergence in terms of standardization compliance. Starting from the initial point where the media is captured, several enhancements are done to improve the QoE as follows:

- **Advanced /signal Detection:** Voice activated switch that helps determine the media delivery. For example, the speaker or the room with the loudest sound will be the one seen by all the other participants.
- **Echo cancelation:** A reflected source of interference created by the original signal source. The device detects the utterances that re-enter the audio path that comes out of the videoconferencing codec output with some time delay. If untreated, it can result in hearing your own voice with a time lag. There is also a strong reverberation effect that cause severe degradation of speech quality, making it difficult to understand the speaker or even hauling created by feedback.
- **Advanced video processing algorithms:** For high resolution delivery with features such as flicker removal, hue and chroma compensation, or image alteration reduction due to improper lighting or optical focus.
Professional teleconferencing solutions have a multitude of elements that work in synergy to deliver high quality output that allow a comfortable and realistic human interaction.

Telepresence Call Flow

There are several layers involved to establish a telepresence call as there are multiple ways to deliver video conferencing to a single or multiple participants.

The typical layers involved in establishing a call are as follows:

1. **User Interface Layer:** This requires user interaction with the device to program the required action. This can be a graphical application running on the device or computer, a voice responsive control, or a push of a specific pattern on dial pad. This is schedule, set up, or places the call. Account policy is there to reinforce the user access rights to various actions that are allowed to be performed for the profile. This is the primary point of access, and it also delivers various technical details to the other layers involved to establish the call.
2. **Conference Control Layer:** This is mainly responsible for resource allocation and resource reservation, and manages traffic functions like call routing operations. This layer gathers information from the user interface, and allows users to reserve and create meeting rooms, send invitations, or remove users from a conference.
3. **Control Plane Layer:** This layer is responsible for the signal stack. The most common signal is the Session Initiation Protocol and H.323. This offers the functionality to interact with the endpoints in an attempt to establish the call or conference for inbound and outbound, and also negotiate the session parameters that all complies to.
4. **Media Plane Layer:** This has the function to mix and deliver the media streams of video and audio payload to and from the joining endpoints. The main function of this layer is to control the function and delivery of media by management of the Real-Time Transport

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

Protocol and Real-Time Transport Control Protocol suites. It assists in process of negotiation of the payload type, codec in use, video frame rate, video resolution, and additional media channel parameters that the teleconference can have.

From topology deployment point of view, the telepresence application can be categorized as point-to-point call referred to as business-to-business or intra-campus conferencing. This is also called the multipoint videoconferencing or, multi-site meetings. These resources are managed by individual enterprises for own communication purposes or is deployed as a service solution for high-end communication. Multipoint Teleconferencing requires a dedicated network design to preserve the QoS that the platform promises to deliver.

A basic business-to-business call follows the following flow. The flow differs from one vendor to the other since they adjust the call flow depending on the device technical capabilities:

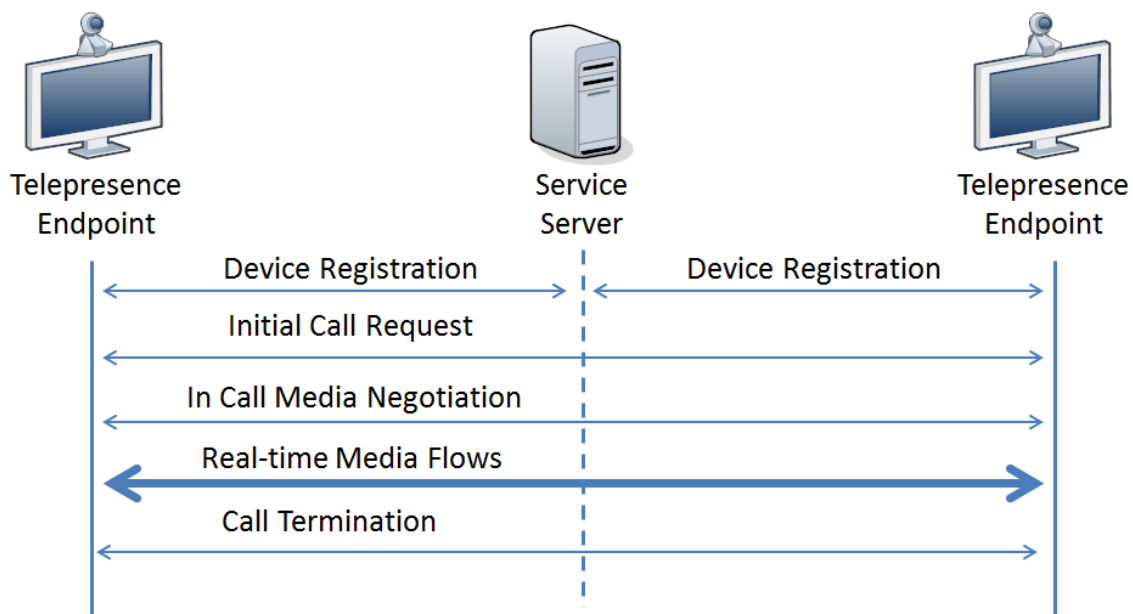


Figure 204. Business-to-business telepresence simplified call flow

The signaling used by the telepresence application is Session Initiation Protocol (SIP). This is a widely deployed signaling protocol for most communication sessions such as voice and video over Internet Protocol.

Endpoints should perform registration to comply with enterprise's security policies. This is done in various ways to allow information exchange only with the trusted entities. The contact header from the SIP REGISTER provides the necessary details for connectivity to the endpoint such as IP address, transport protocol, port number, and endpoint extension.

After successful registration, the endpoint is ready to initiate or receive calls. The user can select a predefined entry and trigger a telepresence in the user interface or dial a known extension, according to the endpoint characteristics. One endpoint sends a SIP INVITE to the other endpoint with details about the required media for transmission. The headers defined in

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

SDP contain data for media negotiation such as codecs in use, codec attributes, and required bandwidth. Depending on the capabilities of the remote party the call determines if it is to continue or new SDP information is to be negotiated.

After the initial contact is established, a second negotiation takes place to enable the real-time video negotiation and establish the data paths required. Both endpoints negotiate as a set of common codec that consider the media capabilities such as, video resolution, video bitrate, and supported list of codecs.

Both endpoints next, start to present the media information received from the other side. The telepresence call is now active. To preserve a feedback mechanism during the call, vendors can implement methods like SIP SUBSCRIBE, SIP OPTIONS, or SIP NOTIFY as update or keep alive mechanisms. The negotiation details are thus enforced by the device capabilities to encode and decode specific formats as well as functionality constraints within the configuration.

A multipoint teleconference flow is similar to the business-to-business call model in that one of the peer endpoints is the telepresence Service Server. All endpoints joining the video conference will call the Service Server; it will relay all signaling and media streams from/to all participants in the conference. From a simplified perspective the multipoint telepresence conferences are not more than several point-to-point calls all landing to the same endpoint, the difference being the dialed number destination is a Service Server. The signaling and media negotiation are otherwise the same and follow similar rules. Depending on the implementation of the Service Server, the SDP negotiation with each endpoint is done at the maximum capabilities of the endpoint or at the level of the least capable endpoint. In the former case the Service Server will do the transcoding; in the latter case all devices will be exchange media at the same level of quality: as an example, if a multipoint telepresence call has negotiated all the endpoints at 1080p resolution with Best quality for video and a new endpoint joins that maximum capabilities are at 720p resolution with Good quality, the SDP information will be negotiated independently to match the announced capabilities.

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

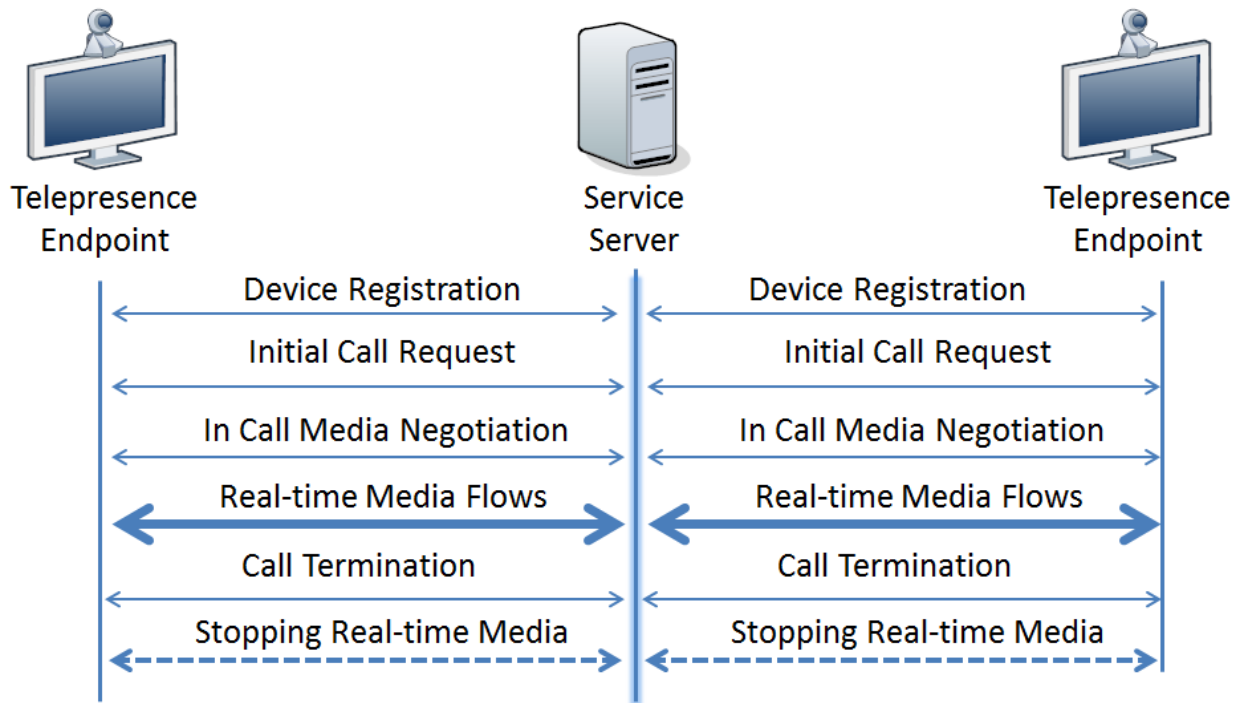


Figure 205. Multipoint telepresence simplified call flow

Typical Cisco's implementation for telepresence call has three cameras and three large high-definition displays to enhance the user experience and several microphones in the room to capture the audio signal. Each stream is coded individual and transmitted to the other side using the negotiated codecs from SDP. The codec type and codec settings will dictate the necessary bandwidth that RTP data generates. For better interaction, there is an additional presentation stream that provides the platform to share content. These aspects are to be considered when designing converged networks since the maximum capacity of the call can affect the inbound or outbound points of the network. The following table provides an example of quality requirements in terms of bandwidth for H.264 video codec and AAC-LD audio codec.

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

Table 76. Telepresence single endpoint bandwidth requirements

Video Quality	Motion quality	Bandwidth per video stream	Bandwidth per microphone	Interactive channels	Compatibility media channels	Aggregated Bandwidth required
1080p	Best	4Mbits	64Kbits	564Kbits	832Kbits	13.6Mbits
1080p	Better	3.5Mbits	64Kbits	564Kbits	832Kbits	12Mbits
1080p	Good	3Mbits	64Kbits	564Kbits	832Kbits	10.5Mbits
720p	Best	2.2Mbits	64Kbits	564Kbits	832Kbits	8.3Mbits
720p	Better	1.5Mbits	64Kbits	564Kbits	832Kbits	6Mbits
720p	Good	1Mbits	64Kbits	564Kbits	832Kbits	4.6Mbits

To understand the call flow that is about to take place, and the requirements for bandwidth, it is useful to capture the SDP information or extract it from endpoints configuration, if available. The expected structure of SDP headers for teleconference call is the following:

- IP address that is going to be used for the media transactions
- The audio codecs supported
 - the preference of use
 - the details about coding time and
 - required bandwidth for each of them
- Video codecs supported
 - the preference of use
 - the details about encoding the media as: frame rate, encoding mode, video resolution,
 - maximum bitrate and the RTCP details that assists real-time media negotiation.
- Optional video codec for presentation channel, that can encode the delivered content to the User Interface
- Optional application interactivity for content delivery.
 These optional channels enhance the audio/visual. The use of such application depends on the technical capabilities of the endpoints.

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

```
m=audio 10000 RTP/AVP 100 102 103 104 105 9 0 101
b=TIAS:64000
a=rtpmap:100 MP4A-LATM/90000
a=fmtp:100 profile-level-id=24;object=23;bitrate=64000
a=rtpmap:102 MP4A-LATM/90000
a=fmtp:102 profile-level-id=24;object=23;bitrate=56000
a=rtpmap:103 MP4A-LATM/90000
a=fmtp:103 profile-level-id=24;object=23;bitrate=48000
a=rtpmap:104 G7221/16000
a=fmtp:104 bitrate=32000
a=rtpmap:105 G7221/16000
a=fmtp:105 bitrate=24000
a=rtpmap:9 G722/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=sendrecv
```

Figure 206. Sample Audio capabilities advertised by single screen endpoint

For the video codecs the options are complex and the SDP information can become quite elaborate. In the example below the endpoint advertise the H.264 as initial primary video codec with alternate options of H.263 codec. The below image has been intentionally truncated to fit the page layout.

```
m=video 20002 RTP/AVP 91 93 94
b=TIAS:5000000
a=rtpmap:91 H264/90000
a=fmtp:91 profile-level-id=428016;max-br=5000;max-mbps=30000;max-fs=3600;//
max-smbps=108000;max-fps=6000;max-rcmd-nalu-size=1382400
a=rtpmap:93 H263-1998/90000
a=fmtp:93 custom=1280,768,3;custom=1280,720,3;custom=1024,768,1;//
custom=1024,576,2;custom=800,600,1;cif4=1;custom=720,480,1;//
custom=640,480,1;custom=512,288,1;cif=1;custom=352,240,1;//
qcif=1;maxbr=7680
a=rtpmap:94 H263/90000
a=fmtp:94 cif4=1;cif=1;qcif=1;maxbr=7680
a=rtcp-fb:* nack pli
a=rtcp-fb:* ccm fir
a=rtcp-fb:* ccm tmmbr
a=sendrecv
a=content:main
a=answer:full
```

Figure 207. Sample Video capabilities advertised by single screen endpoint

The main protocol for video in telepresence is H.264/MPEG-4. This represents the standard for video compression of captured media that requires good quality transmission of high definition video content. The codec is developed by International Telecommunication Union (ITU). It is deployed on a large number of applications and technologies such as high definition content delivery websites, terrestrial or satellite television, Blu-ray Discs, or even online content stores.

The H.264/MPEG Part 10 or, Advanced Video Coding (AVC) is developed to deliver video content at lower bitrate in regards to its predecessors, without any increase in the design

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

complexity or cost of implementation. An additional goal is to allow selection for bit rates scaling for high or low video resolutions and to allow broadcasting of content, disk storage, IP encapsulation, and even telephony transport.

The main features promoted by the codec are:

- Multipicture inter-picture prediction, that allows significant bit rate reduction for repetitive motion content
- Variable block-size motion compensation that allows precise segmentation of moving regions increasing the efficiency over used bitrate
- Multiple motion vectors per macroblock that allow a robust motion detection decoding at the player side
- Motion compensation with weighted prediction
- Lossless macroblock coding that allow representation of specific regions without consuming large amounts of data
- Elaborated algorithms for spatial blocking transform optimization
- Context-adaptive binary arithmetic coding (CABAC) as highly data efficient encoder
- Context-adaptive variable length (CAVLC). This is a less demanding data processing compared to CABAC but more complex in terms of coefficient coding
- Various loss resilience features as Network Abstract Layer, Flexible Macroblock Ordering, or Frame Numbering to better compensate the transmission's channels information losses or errors.

The codec has advanced techniques of encoding and decoding high definition media with a better bit rate efficiency. There still was the need for a more scalable method to deliver the information when endpoint capabilities that demand the same stream are different. Annex G of H.264/MPEG-4 AVC was an answer that allows the encoding of the main streams with one or more subset bitstreams.

The main advantage of H.264-Scalable Video Coding (SVC) is that it uses less bandwidth as it drops specific portions of data from the original stream and displays at reasonable quality the resulted video signal. It delivers the video to the user's display with a drop in spatial resolution (decrease the virtual screen size), decrease in temporal resolution (use a lower number as frame per second), or lower-quality video signal, thus consuming a smaller bandwidth. These scalars can be used independently or in combinations for better converge of the media delivery over impaired network infrastructures.

H.264-SVC can use the base layers for a video stream and construct delivery over multiple enhancement layers for media transmission. It is essential for teleconferencing especially over loss prone and low bandwidth networks like the Internet. The advantage of H.264-SVC is that one encoder can send a single media stream to a multitude of heterogeneous endpoints where independent decoding is performed, depending on the technical capabilities. Several independent tests have shown that in certain situations when H.264-SVC codec is used and the packet loss that a network can introduce is between 20% and 40%, the user can have a decent, intelligible image compared to the packet loss acceptance criteria of H.264 AVC where the loss tolerances is up to 5%.

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

One of the noticeable disadvantages of using H.264-SVC is inter-operability issues. One vendor, for example, can implement specific optimization or algorithms for low bandwidth video delivery that cannot deliver the same results on the device of another vendor. The inter-operability requires large amount of tests, to ensure that most of the cases are covered and the video delivery to the endpoint converges to a good quality.

For most telepresence applications, the video cameras capture and encode a static image with little movement so the overall video rate can drop up to 35% of the per stream average bitrates. Dealing with variable codec such as H.264, the challenge comes when the peak rates exceed the average estimations. This generates a significant traffic over the network and affects the QoS. As a best practice, it is good to over-commission the video bandwidth rate enforcements and allow up to 20% overhead if needed. Another important aspect to be mentioned is the video tendency to send a large number of packets in short burst instead of an even distribution, thus tolerances are accommodated on the entire network infrastructure. Without these optimizations the effect will be a poor image quality received, mostly because of the delay and jitter introduced by passing at a constant rate the peaks of data through the various devices' queues.

As for the Audio media encoding telepresence mandates the use of MPEG-4 Low Delay Audio Coder (AAC-LD), a powerful audio compression format that assists both the perceptual audio coding and the low delay necessary for bidirectional audio communication. The codec enforces a maximum logarithmic delay of 20ms and has a good support for speech and music encoding. There are several bitrates that the encoder can use starting from 32Kbits, but compared with other speech codecs the quality improves with the increase of bitrate.

The challenges of Preserving the QoS for Telepresence

For most of the telepresence deployments the network pass-through is the most significant source of delay and jitter. This is a fixed characteristic since the time that the signal takes to propagate from one point to the other, can increase dramatically when there are bottlenecks of bandwidth in between. For multipoint telepresence calls, the Service Server sitting between the endpoints delivers additional delay due to internal queue processing or load processing. IxLoad assists in measuring the required time for multimedia channels to reach from one end to the other, as part of QoS assessment when measuring the MOS score. The One way delay metrics, measures the time needed from one codec's output to other's codec input with a high precision, for a multitude of current active channels.

A set of tools and techniques are required to preserve high quality for real-time high-definition video and audio over converged IP network. This minimizes the effect of impairments such as bandwidth rate policies, latency, jitter, and packet loss. By design, voice and real-time video should be granted high-priority service, while other non-critical applications may have a lower priority enforced. Several questions are to be asked, before actual telepresence service deployment, which can assist in building the converged network service prioritization requirements:

- Is telepresence the only traffic objective?
- Which additional real-time media services will be deployed over the converged network?
- Which critical applications will share the same network resources?

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

- Are there other types of traffic identified that did not match those above? What are those traffic characteristics and the desired QoS?

The video codecs compensate well with delays up to 200ms from end-to-end, but higher values can trigger changes in bitrate or resolution. However, a 200ms delay has a significant effect on audio quality. Typically, vendors implement methods of informing the user about large latencies and the possible effect on the call quality. If end-to-end delays larger than 400ms are recorded, the call might not be sustained and call termination might be encountered. However, in certain situations like intercontinental calls or satellite teleconferencing, these limits can be increased to allow a sustainable call. This implies a lower rating in user experience index due to the amount of time required for a spoken message to be acknowledged. For these situations, even if technologically the telepresence application will be feasible the user's expectations should be set accordingly.

Besides the effect of delay, delay variation can degrade the quality of the received media. To preserve high quality video, the telepresence maximum jitter should be within 10% of the maximum delay accepted. For example, if the end-to-end delay accepted before quality change is 200ms, the maximum tolerated jitter should not exceed 20ms. Depending on the coding and decoding techniques, different vendors can adjust their thresholds to produce a better user experience and a higher-quality image.

Packet loss is the network impairment that most affects the quality of transmission. As the media information is bandwidth-efficient encoded, any small loss can affect the overall performance. For example, a packet loss of 10% greatly affects the decoder ability to build an image; this should trigger the call to terminate. As a best practice, to preserve the user experience within reasonable ratings, terminate at a packet loss of 1%. The telepresence system should trigger a change to codec bitrate or image resolution in an attempt to lower the active bitrate and packet loss percentage. These values represent observations in certain test configurations and the actual parameter values are adjusted according to specific vendor implementations.

After identifying the sources of network inbound and outbound traffic impairments, the system architect should design and implement corrective actions and optimize configuration to overcome them during day-to-day operation. It is important to measure the user QoE, especially for interactive media applications. Since the early days of telephony, there were static methods to rate the quality of a transmission line. As the digital era overtook the analog signal transmissions the methods had to be adjusted for the same need to measure the user quality index.

The MOS score represents the user-point-of-view metric for QoS over a network as described in the E-model defined by ITU in the G.107 document. As it is considered a subjective way to evaluate the quality of a transmission, ITU has ratified recommendations on how to measure it in the P.800 technical paper. The methods of measurement are generally applicable regardless of the form of degradation the signal is suffering, as it may be packet loss, corrupted payload as an effect of bit errors, various types of noises, propagation delay and delay variations, coding

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

and decoding schemes, as well as the echo or side tone of an audio signal. The score has the lowest quality rated at 1, with a maximum scale value of 5.

To measure the user's rating for audio and video quality for telepresence applications, IxLoad uses the industry's leading library from Telchemy. For each individual channel, if the MOS calculation option is enabled, the agent performs a non-intrusive assessment on quality, displaying a wide range of statistics for channel monitoring and network diagnostics. Additionally, the Media Quality Assessment is performed using the VQmon engine for codec-specific-induced degradations due to variations in speech or video quality that each codec can have.

The VQmon library is running on all Ixia ports assigned for media traffic generation and it collects the required statistics about any network impairment such as packet loss, buffer accumulation, coding and transcoding times, propagation delay, and bandwidth capping to compute the user experience index for audio/video MOS.

The Video Quality Assessment Engine measures individual media streams, delivering ratings based on gathering specific information such as:

- Type of frame and Group of Pictures structure
- Per-frame video quality
- End-to-end, roundtrip, and system delay
- RTCP metrics
- Average and maximum bitrate per frame type
- Content descriptors such as level of detail, motion, panning, frozen, or blank video
- Network-specific quality metrics and much more

Objective

This test methodology helps evaluate the quality of transport of the inter-campus transport network. The simulated application is the videoconferencing between the headquarters and a remote branch over the deployed MPLS network or the METRO Ethernet link. The user experience rating is measured for the audio and video streams played using the MOS rating, for the business-to-business telepresence deployment model.

Setup

The telepresence endpoints are emulated by IxLoad. The transport network between the endpoints represents the SUT target for QoS qualification. The appropriate configuration must be deployed prior to test execution to allow networking access from one side to the other.

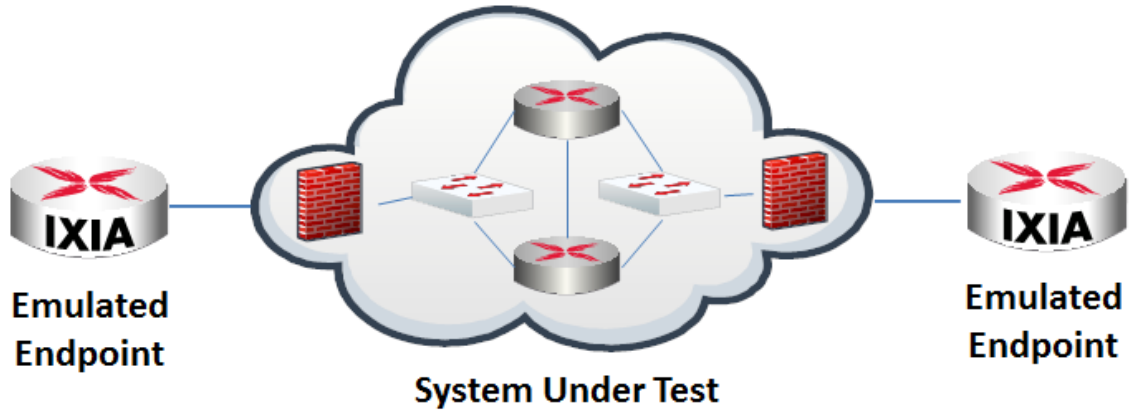


Figure 208. Environment for telepresence application testing

NOTE: IxLoad client should be installed with Media Clips library. This contains all the multimedia files required by the telepresence scenario emulation. Any custom files can be loaded in the Media Library pool that can represent real recordings of events or actions.

Step-by-Step Instructions

The step-by-step instructions highlight how to set the essential parameters of this configuration and explain additional options, which can be used to change the behavior of the test.

1. Start IxLoad.
2. Select **New** from the **File** menu options, and then select **Templates**.

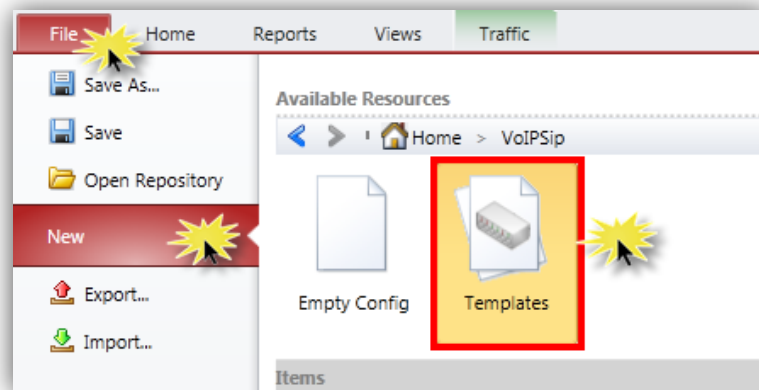


Figure 209. Path to Templates configuration files

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

3. In the new page, go to **VoIPSip** folder.

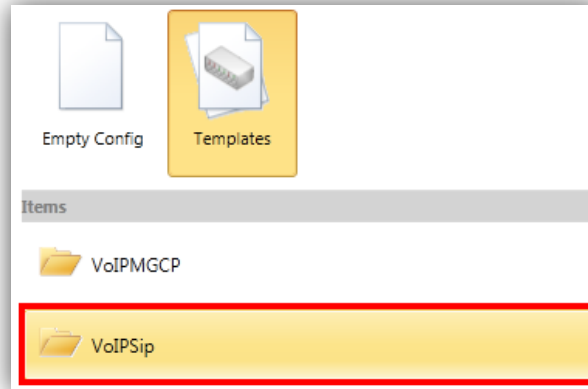


Figure 210. VoIPSip configuration templates

4. Select the **TelepresenceSample.rxf** file from the **Telepresence** folder. The configuration is loaded when the files is accessed.

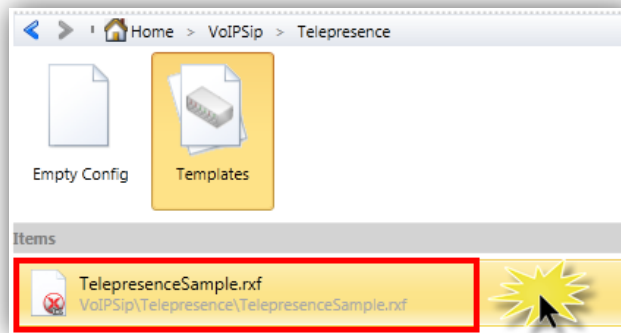


Figure 211. IxLoad telepresence configuration template

5. Access the originating **NetTraffics** to change the network connectivity details, by accessing the **Network1** IP connectivity details.

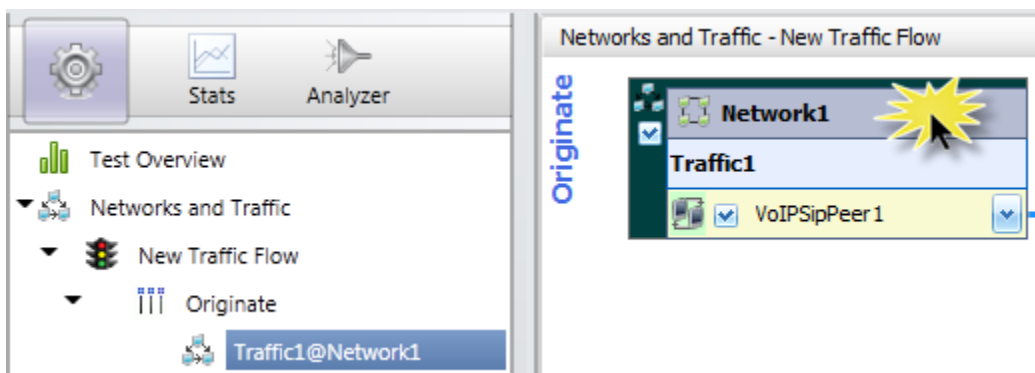



Figure 212. Accessing network configuration details

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

- Configure the necessary IP addresses, subnet mask, IP count for simulation, and default gateway details.

NOTE:

- Several IP address ranges can be configured in parallel to emulate large networks or different office branches by accessing the  button.
- IxLoad offers support for IPv4 and IPv6. Before test execution, verify in the release notes or application datasheet if the configured protocol has adequate support for that IP version.

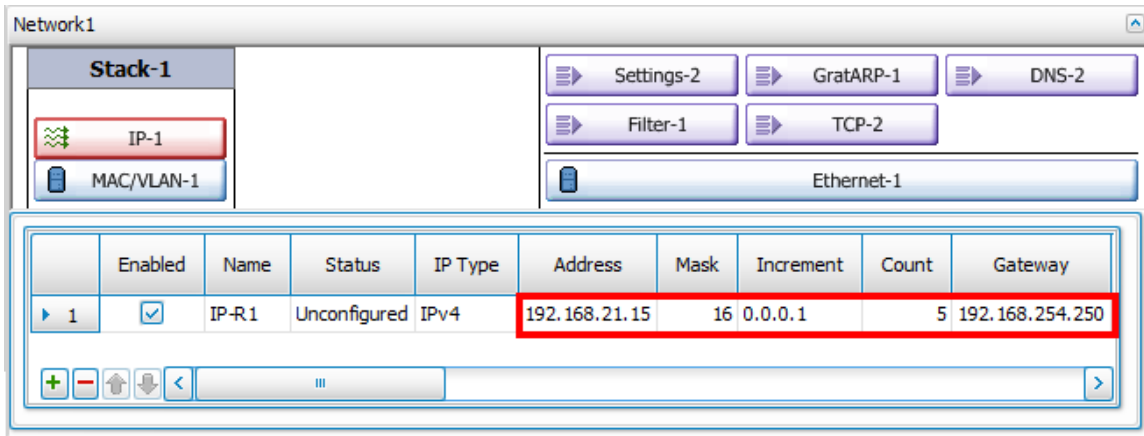


Figure 213. Changing the network connectivity details

Table 77. Template network configuration

Endpoint	IP Address	IP Mask	IP Count	Default Gateway
Originate	2.3.1.1	16 bits	5	0.0.0.0
Terminate	2.3.100.1	16 bits	5	0.0.0.0

- Adjust the network configuration to access the application endpoint emulation on the **Originate** side.
- Click the **VoIPSipPeer1** peer to access the endpoint’s settings.

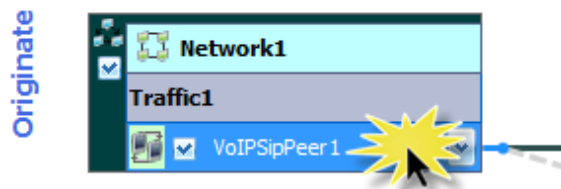


Figure 214. Accessing application endpoint settings

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

9. Select the **Audio** tab and verify the current configuration. To measure the quality of speech for the emulated traffic, enable **Perform MOS** and **Calculate One Way Delay** options. During test execution these will provide real-time statistics for the emulated endpoints as well as aggregated data for the entire call rating.

NOTE: If the option to calculate MOS is enabled, the IxLoad user interface enables the option to measure one-way delay.

The screenshot displays the IxLoad configuration interface for the 'Audio' tab. The 'Audio' tab is highlighted with a red box. The 'Perform MOS' and 'Calculate One Way Delay' checkboxes are also highlighted with red boxes. The interface includes the following sections:

- Enable audio on this activity** (checked): (if unchecked, all audio script functions will be SKIPPED)
- Play Settings**:
 - Clip: UK_31_Telepresence48.mp4
 - Format: AAC-LD, Duration: 40020 ms, Size: 320941 bytes
 - Output level: -20 dBm
 - Play for: 30 Seconds
- Type Of Service**:
 - TOS/DSCP: Class 1 (0x20)
- Enable jitter buffer** (unchecked):
 - Buffer size: 20 ms
 - Use compensation (unchecked)
 - Max. size: 1000 ms
 - Max. dropped consecutive packets: 7
- Perform QoV** (unchecked):
 - Units: # of Channels
 - Value: 100
 - Channel Selection: First Channels
- Generate silence** (unchecked):
 - Null data encoded (unchecked)
 - Comfort noise (checked)
- Buttons: Verify all settings, Restore defaults

Figure 215. Enabling the voice quality measurement

NOTE: Use the template configuration that has the actual telepresence call of an audio stream sent for 30 seconds. The play duration value can be adjusted to better suit the scope and duration of the test.

10. Select the **Video** configuration tab and verify the current configuration. The activity playback is set for **Telepresence** mode. To measure the quality of video for the emulated traffic, enable **Perform MOS** and **Calculate One Way Delay** options. The template configuration emulates a Teleconference session with the duration of 60 seconds. There are three emulated speakers, each talking for 20 seconds. The telepresence speaker rotation scheme can be changed from **Telepresence Settings**, as well as other protocol-specific options.

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

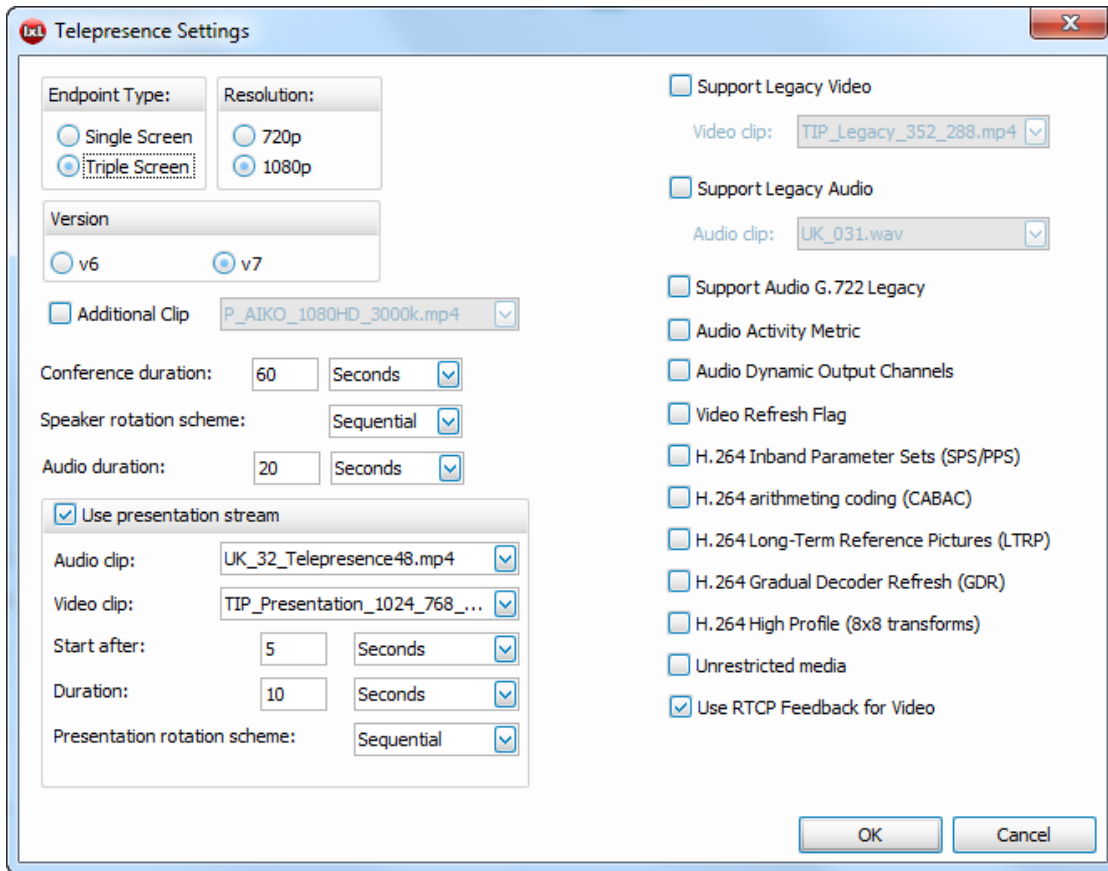


Figure 216. Telepresence settings options

The simulated endpoints act as a triple screen in high definition using the 1080p resolution. This represents intense bandwidth consumption and assists in qualifying the system under test for media delivery. The video stream from the template is H.264-encoded with an average bandwidth of almost 2Mbits. The video file payload can be changed for the **video** configuration tab from the **Playback Clip** option. There are several files encoded on different bandwidths and resolution as part of the IxLoad installation and the user has to option to load custom files in the media library by accessing the **Media Setting** icon on the configuration ribbon.

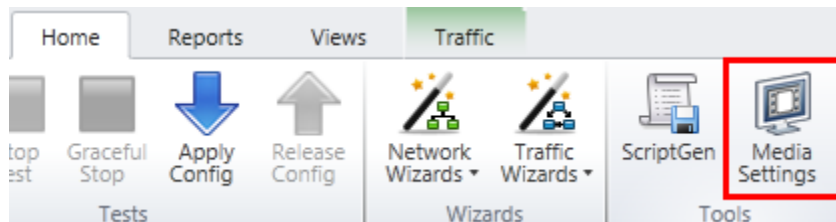


Figure 217. Accessing Media Settings options

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

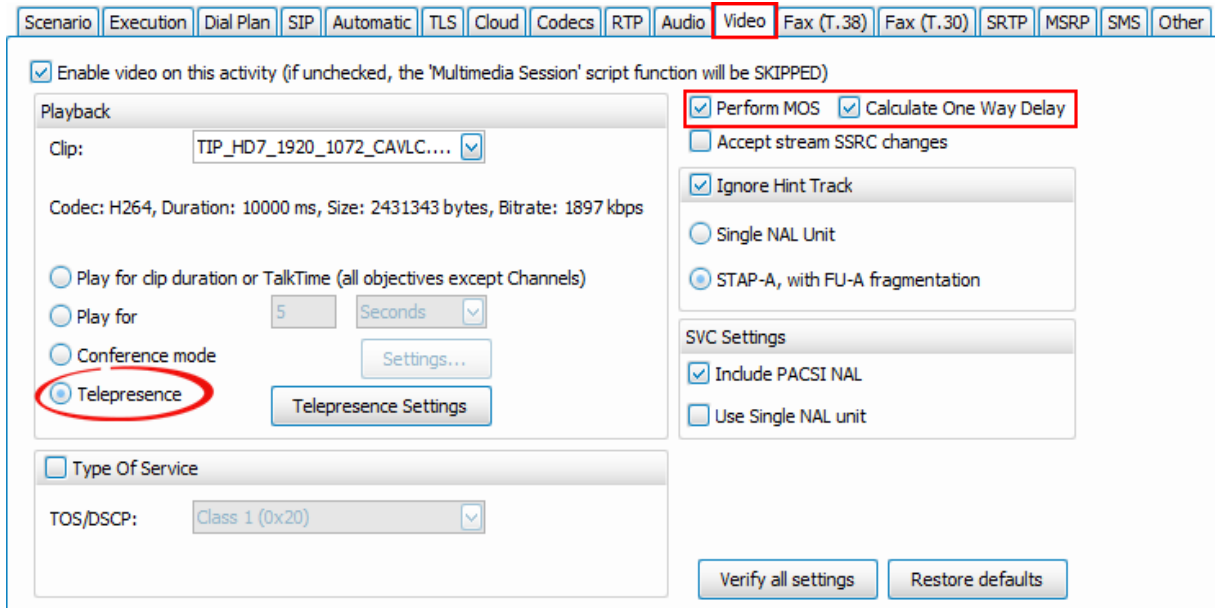



Figure 218. Enabling the voice quality measurement

11. Repeat steps 5 to 9 for the **Terminate** side NetTraffics on the **VoIPSipPeer2** telepresence activity endpoint. The application simulation is now ready for execution using the default scenario activity emulation for triple screens and HD1080p streams.
12. Select the **Timeline and Objective** option from the menu tree, and set the required test objective. For initial testing of network convergence it is suggested to set 1 channel objective and work the way up to measure the maximum performance of the **SUT**.

Network Traffic Mapping	Objective Type	Objective Value	Iteration Time	Total Time
[-] New Traffic Flow				
[-] Activity Links				
[-] VoiceLink1	Channels	1	000:05:11	000:05:11
[-] VoIPSipPeer 1@Network1	Channels	1	000:05:11	000:05:11
[-] VoIPSipPeer 2@Network2	Channels	1	000:05:11	000:05:11

Figure 219. Adjusting the test objective

The test is ready now for execution. It is necessary to assign traffic ports and map them to the right activity emulation. To do so:

13. Select **Ports** menu from 
14. Add the IP address or domain name of the allocated chassis and wait for IxLoad to refresh and display the available port resources.
15. Select the options appropriate for your test configuration and network infrastructure; map them to the corresponding NetTraffics. Once complete, save the file to disk for further use.

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

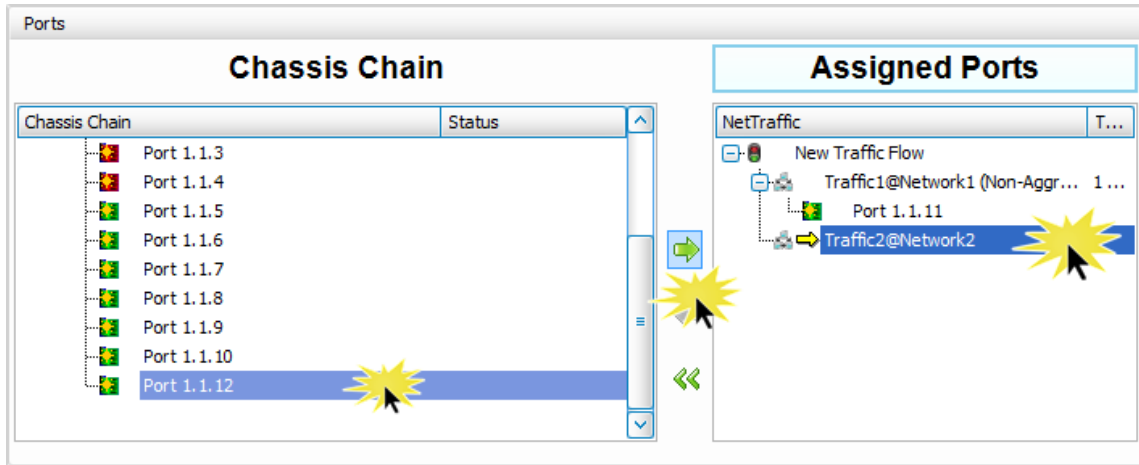



Figure 220. Assigning ports to NetTraffics

16. Start the test execution by triggering the **Start Test**  from the ribbon commands. At this time the **SUT** configuration should have been adjusted accordingly to the scope of test. As necessary, modify QoS policy and firewall rules to allow SIP signaling to pass from one endpoint to another as well as to the RTP and media signaling data.

Results Analysis

The following questions provide guidelines on how to interpret the results during test execution and how to identify issues that can arise during the objective's measurement.

Have any call failures been reported? Check the **Calls (VoIPSip)** view.

Table 78. Reported statistics

Statistic Name	Value	Questions
Calls Attempted		Have any call attempts failed? Compare: Calls Attempted and Calls Received, with Calls Attempted and Calls Connected.
Calls Connected		
Calls Received		
Calls Answered		
End Calls Received		
End Calls Completed		

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

Has the test objective been achieved? Check the **Channels (VoIPSip)** view.

Table 79.

Statistic Name	Value	Questions
Active channels		Have the channels been connected continuously at a constant rate during the Sustain Time?
Successful channels		Are there any failed channels identified?
Failed channels		

What is the quality of the audio streams transmitted? Check the **RTP MOS (VoIPSip)** view.

Table 80.

Statistic Name	Value	Quality	Questions
MOS Best			Are all values above 4?
MOS Worst			Were there any streams with less than 4 score?
MOS per call best			
MOS per call worst			

What is the quality of the video streams transmitted?

Check the **Video RTP Relative MOS (VoIPSip)** view.

Table 81.

Statistic Name	Value	Quality	Questions
Instant Relative MOS_V Average			Are all values above 4?
Instant Relative MOS_V Minimum			Were there any streams with less than 4 score?
Instant Relative MOS_V Maximum			
Completed Relative MOS_V Average			

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

Check the **Video RTP Absolute MOS (VoIPSip)** view.

Table 82.

Statistic Name	Value	Quality	Questions
Instant Absolute MOS_V Average			Are all values above 4?
Instant Absolute MOS_V Minimum			Were there any streams with less than 4 score?
Instant Absolute MOS_V Maximum			
Completed Absolute MOS_V Average			

Reference MOS to user perceived call quality

Table 83.

MOS Value	Quality	Perceived Impairment
5	Excellent	Imperceptible
4	Good	Imperfections can be perceived
3	Fair	Slightly annoying
2	Poor	Nearly impossible to communicate
1	Bad	Impossible to communicate

Test Variables

Use each of the following variables in separate test cases. Use the above test case as a baseline and modify a few parameters in the same test run. You can create various scalability tests to measure the SUT's operating performance under actual telepresence traffic.

Test Case: Measuring Quality of Experience for Multimedia VoIP Calls

Table 84.

Performance Variable	Description
Configure multiple channels objective	Scale up the number of simulated channels to benchmark the maximum capacity that the system under test can handle.
Use single screen model	By default the telepresence scenario uses a triple monitor/camera environment. Change the scenario to single screen emulated endpoint to simplify the test setup and decrease media traffic.
Change the video files	In the Video configuration tab at the emulated endpoint side, modify the playback file to various bitrates and codec modes.
Speaker rotation scheme	From Telepresence Settings configuration screen found in Video tab, modify the speaker duration time and rotation scheme. This converges into different patterns of traffic through the SUT and offers valuable data for the QoS measurement.
Increase test time duration	By increasing the play time for the audio and video streams the emulated call increases the overall duration. This duration should be adjusted also into the Timeline and Objective's Sustain Time .

Troubleshooting and Diagnostics

Table 85.

Issue	Troubleshooting Solution
Calls are not established	Ensure that the SUT configuration is allowing SIP signaling traffic from one endpoint to the other. Verify the Event viewer for error information.
No media traffic is flowing	To ensure that media traffic is flowing through the system, verify the firewall rules to allow RTP and RTCP to pass through. If necessary, enable Analyzer function and capture the SIP conversation. Analyze the SDP information exchanged for additional details on the sockets negotiated for media exchange.
Measured MOS has low-quality rating	To overcome this, verify in the IxLoad statistics the delay, jitter, and packet loss introduced by the SUT. Depending on the observations, adjust the configuration to allow a better connectivity between the emulated endpoints.

Conclusions

This test has offered the methodology to measure the QoS for media delivery in the case of business-to-business telepresence applications. The SUT configuration and performance are determined for an optimum user experience, using the existing configuration template.

Test Case: Telephony Denial of Service

Overview

As a general use, the term “denial of service” represents the attempt of sending malicious traffic towards a destination until the available resources are no longer available for the users of interest or the recipient of the traffic. In telephony, this concept has a very simple example: the destination phone number is called over and over again by an automatic tool, thus preventing any other incoming or outgoing calls. These can look perfectly legitimate from signaling point of view and might represent a prank call or a form of telephone harassment. However, the originator can have a spoofed caller identity, which makes the tracing even more complicated. Spoofed identity, just like in the source IP DoS, amplifies the level of security exposure and typically ends in having blacklisted an entire range of legitimate phone numbers.

These types of organized telephony DoS (TDoS) might be triggered by social networks groups that want their point of view to be heard by the greater public or they react to an action initiated by the target of the attack. These are typically towards financial institutions, governmental agencies, public sector companies, or even political campaign offices or media broadcasting corporations. There has been a world-wide increase of such target-specific telephone attacks, and agencies such as Department of Homeland Security and the FBI are taking actions to prevent and eventually pursue complaints received from individuals or companies. The most noticed forms of TDoS were fraud call for debt collection and extortion scams. The attacker demands a sum of money and, failing to receive this amount, will trigger a flood of streams clogging the system for a long period of time.

A newer threat observed, is the malicious signaling traffic intended to corrupt or compromise telephony systems deployed in organizations or public sector services. The reason is not always obvious, but the effects on the service are severe and usually have a long recovery time. This form of malicious traffic has several levels of impact on the recipient of the attack, including damage to the public image and the immediate financial losses due to increased customer churn. Also, there are the added costs for analysis of the security breach and from the prevention systems deployed after the incident. Usually these are over-architected, to prevent future “worst case scenarios” and the costs penalties are significant. Another long-term cost is the added investment in recovering the company image and replacing lost customers.

In most of the cases, the digital telephony has a low level of security, and as soon as network convergence is obtained for optimum user experience and system capacity, no additional changes are made to avoid disruption of the functional system. Additional levels of voice security enforcements have immediate effect on the capacity or quality and if the effect is significant, those changes are reverted and rarely re-evaluated and enforced. This paradigm of not constantly improving the security of a system that works in production leaves open doors for malicious traffic and may compromise the entire system stability.

Recently the Department of Homeland Security announced that investigations are active for hundreds of service attacks, and a third of them were actually targeting governmental offices and agencies. Most of these attacks are using open source software, reprogrammed to perform floods of calls against a targeted pool of numbers, while others have the capability to malformed the signaling traffic to compromise the recipient device.

Test Case: Telephony Denial of Service

Such attacks have been found and documented over time on security-focused websites or third-party security exploits tools, however more and more of these have a security patch deployed. Such malformed packet attacks typically generate system instability or exploit buffer overflows caused when receiving these frames. Attacks could cause a remote DoS attack by triggering an undefined state of the SIP stack, cause increase of time in the processing of new request, allow unauthorized access to the resources, or even allow remote execution of code. Such attacks include “INVITE of Death” that uses crafted message to handle the content of Via header, TDoS BYE attack that also affects the stability of the system by the use of crafted BYE messages, and slow reply of SIP ACK with crafted SDP.

As an example, an open source SIP proxy server has a serious vulnerability to large messages over TCP that may lead to a system crash. The trigger to such behavior is a remote forged SIP REGISTER over TCP transport. The stack allocates memory for the incoming packet, and if the packet content information is larger than the expected message length, the SIP engine generates a buffer overflow, taking down the entire services for all the serving users. This is a critical issue as the amount of traffic required to take down the service is extremely low, while the effect on the services is severe.

This malformed signaling traffic methodology for TDoS attacks will be treated in this test case, and it will cover a few scenarios that can be achieved with IxLoad as a traffic initiator. The RFC4475 contains a large suite of test cases for SIP torture testing, this document does not intend to cover the entire suite and it addresses a limited set of examples. The scope is to demonstrate the flexibility of the IxLoad test suite, and should be always used in a lab environment for testing purposes.

Typically when a malformed packet is received by the DUT, this should be dropped and no longer processed. Processing invalid information or parameters with invalid values might lead to memory leaks, buffer overflows, or in extreme situations may compromise the application’s capability to process future requests. A more severe case can be when the application allows remote access to the system resources after such manipulation of malicious traffic.

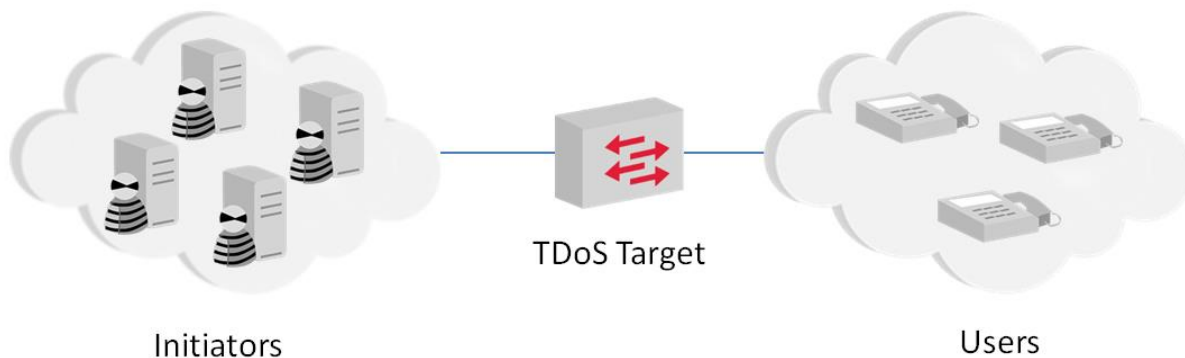


Figure 221. Generic topology for targeted Telephony DoS

Objective

This test demonstrates the steps required to configure IxLoad to send forged SIP packets to test stability of the device or SUT when these are received. Additional test cases can be created by altering the test procedures, the order in which they are transmitted, or the variables used in the example. Please consult the Test Variables section of this test case for more details. During the

Test Case: Telephony Denial of Service

test execution it is recommended to monitor the DUT's log files for debug information that might be triggered by the malicious traffic and monitor the system utilized resources. If the RAM or CPU load is increasing or system stability is compromised, additional verifications should be performed. If the device restarts or stops responding properly, the test is considered failed and immediate actions should be taken to prevent security failures during live operation.

Setup

In this test topology, a single Ixia port emulates the malicious SIP User Agents in a public network initiating traffic towards a SIP Registrar Server as the DUT. The SIP User Agents attempt registration with forged content on the DUT—an application layer gateway or SBC.

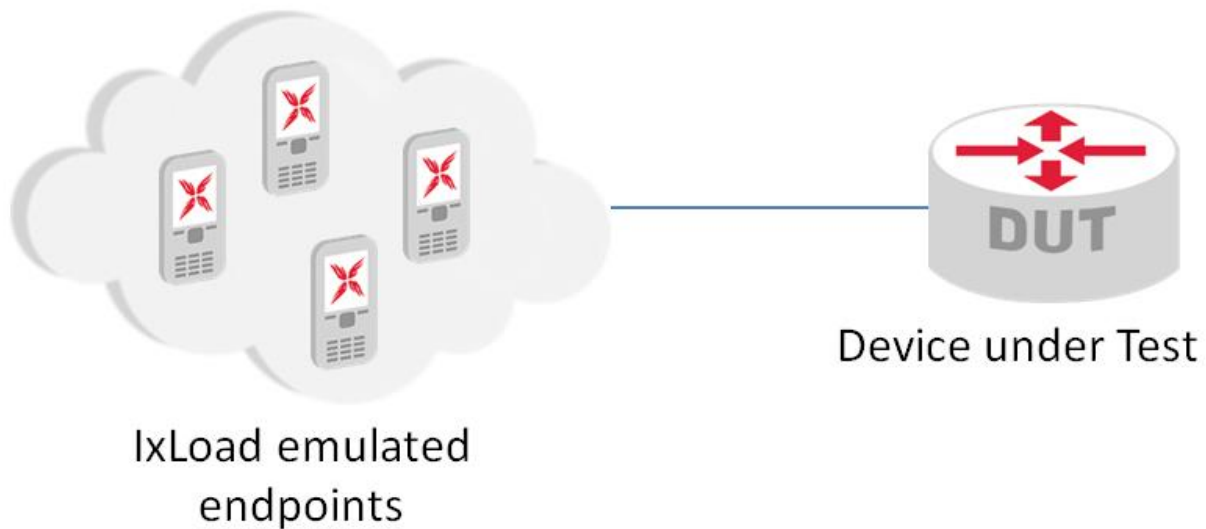


Figure 222. VoIPSIP topology for Telephony DoS emulation

Step-by-Step Instructions

A sample of the configuration as a result of these steps is provided on the Ixia website as *TDoS scenario.crf*. To import the Compressed Repository File (crf) in IxLoad use the command **Import** under the **File** menu. Follow the wizard to save on the local drive the included files.

The step-by-step instructions highlight how to set the essential parameters of this configuration and explain additional options, which may be used to change the behavior of the test or create new test scenarios.

Test Case: Telephony Denial of Service

Setting the Network Parameters

1. Start IxLoad application

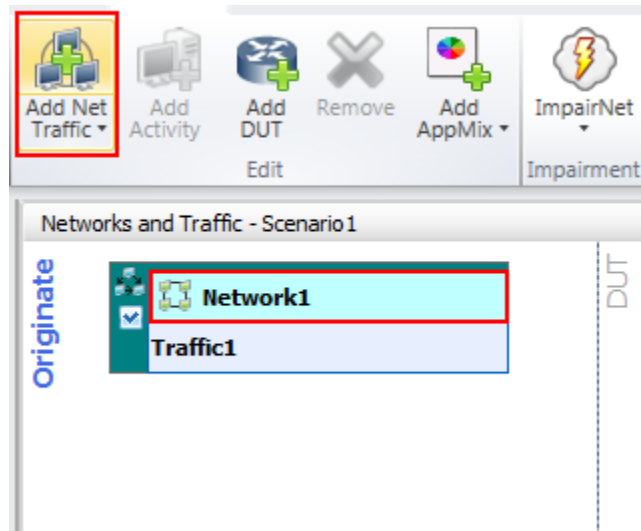



Figure 223. Adding a new originating NetTraffic

Table 86. Sample network configuration

Parameter	Value
IP version	IPv4
Address	20.20.20.1
Mask	24
Increment	0.0.0.1
Count	100
Gateway	20.20.20.254

Test Case: Telephony Denial of Service

2. Add a new **VoIPSIP Peer** activity by clicking the  button. As necessary, expand the **Voice** section from the drop down control window.

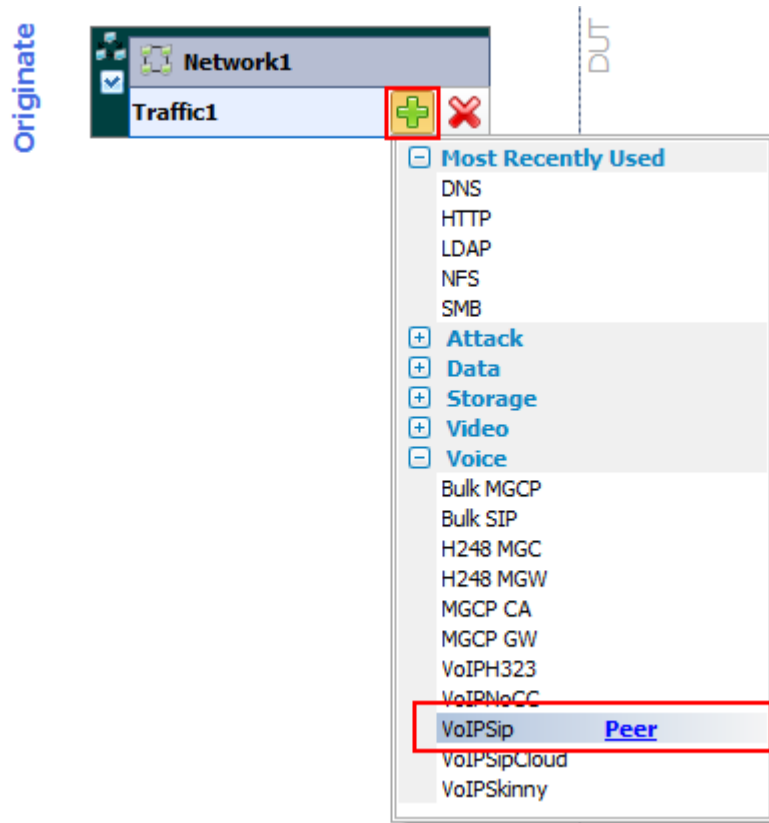


Figure 224. Adding a new VoIPSIP Peer activity

Test Case: Telephony Denial of Service

- Click the **VoIPSIP Peer** activity and select the **Scenario** control tab. This contains the controls for call flow definition and global controls of the emulated channels. As this scenario is focused on the signaling part of the TDoS, the following steps will detail the actions required to send forged SIP messages. This exercise will assist in constructing several REGISTER messages with various headers altered. Similar steps can be followed to construct custom SIP methods such as INVITE, OPTIONS, PING, and others. In addition to custom messages the channel call flow can be manipulated to allow a higher flexibility in terms of SIP endpoint state machine emulation.

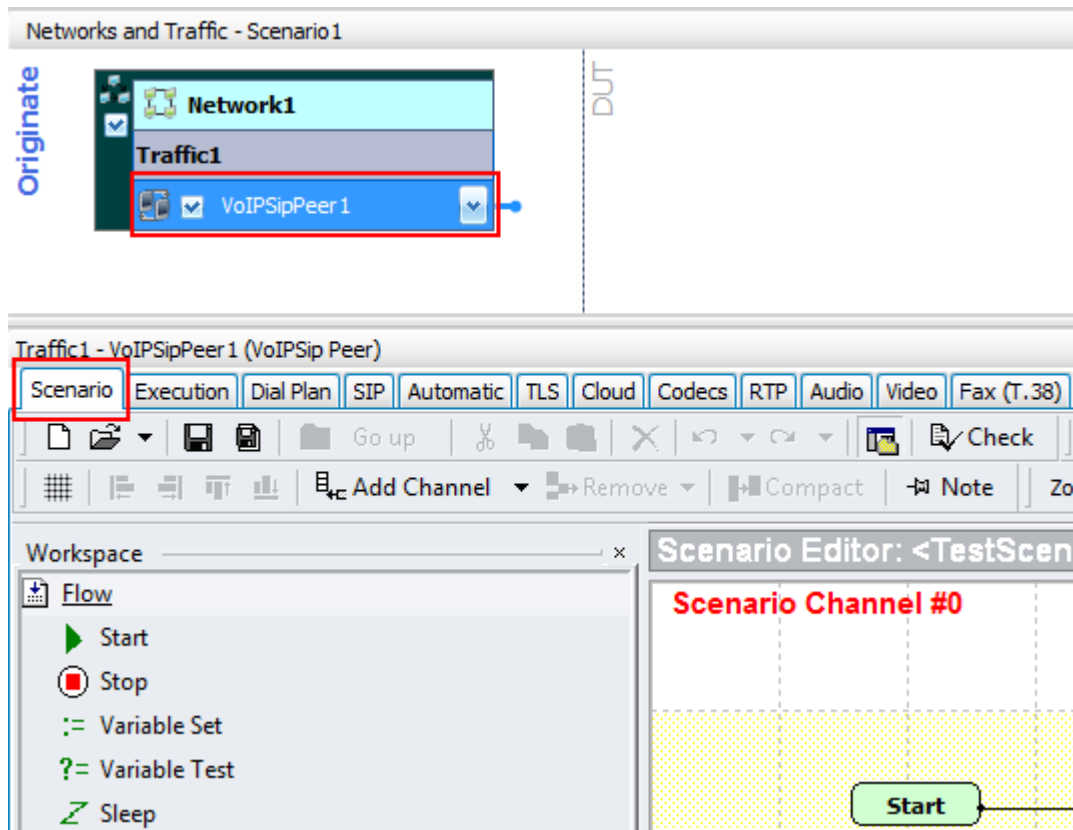


Figure 225. Accessing the Scenario Editor configuration tab

Test Case: Telephony Denial of Service

- From the left side menu of the **Workspace**, navigate to the **Procedure Library** and expand the **SIP** option. This will display all the predefined functions that come with the default installation of IxLoad. They allow a seamless configuration of the endpoint state machine with predefined actions. Navigate through the options up to the **SIP MakeRegistration – Authentication** procedure. Click on it and add the new selection on the right side screen in the **Scenario Editor** screen for **Scenario Channel #0**. Once the procedure has been added as shown below, link it from the **Start** function using the **+** symbol.

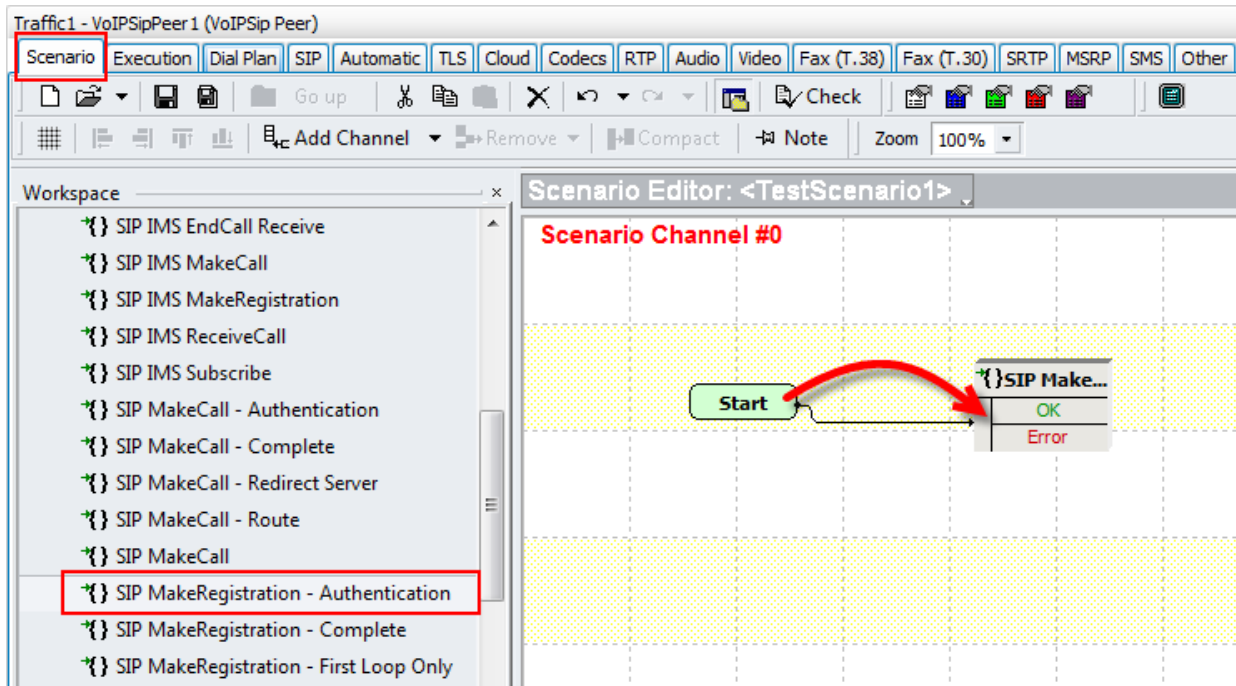


Figure 226. Adding a predefined procedure in the Scenario Editor

Test Case: Telephony Denial of Service

5. Double-click on the **SIP MakeRegistration – Authentication** procedure in the Scenario Editor to access the containing procedures. The following steps will detail the actions needed to modify the SIP signaling headers to alter the proper behavior of an SIP endpoint, and exploit any issue that might cause a security fault of the DUT.
6. This new view will allow access to all the predefined call flow messages. Open the first **Send REGISTER** procedure and modify **Contact** header information to point to 127.0.0.1 as the localhost address. The expected behavior of the DUT is to register the originating phone number in its database as a local resource. Modify the header as: *Contact: < sip:|\$VOIP_Phone|@127.0.0.1 >*. The variables usage is syntax-sensitive so attention should be given when composing the field. An invalid syntax should trigger a SIP parser error on the DUT and this represents another test case.

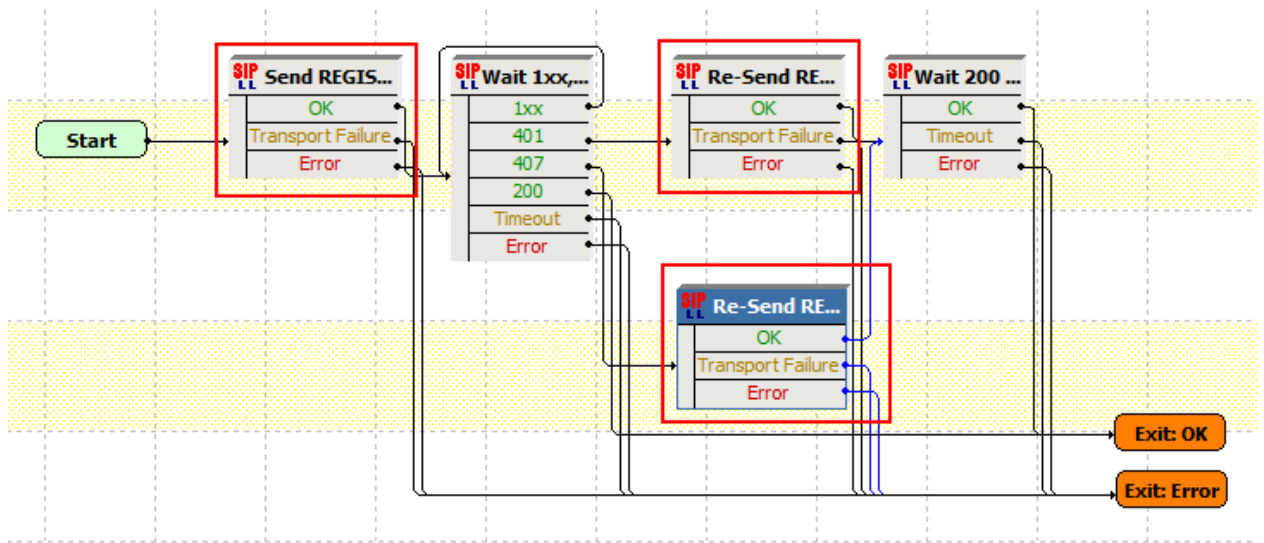


Figure 227. Accessing the lower-level Registration procedures

Test Case: Telephony Denial of Service

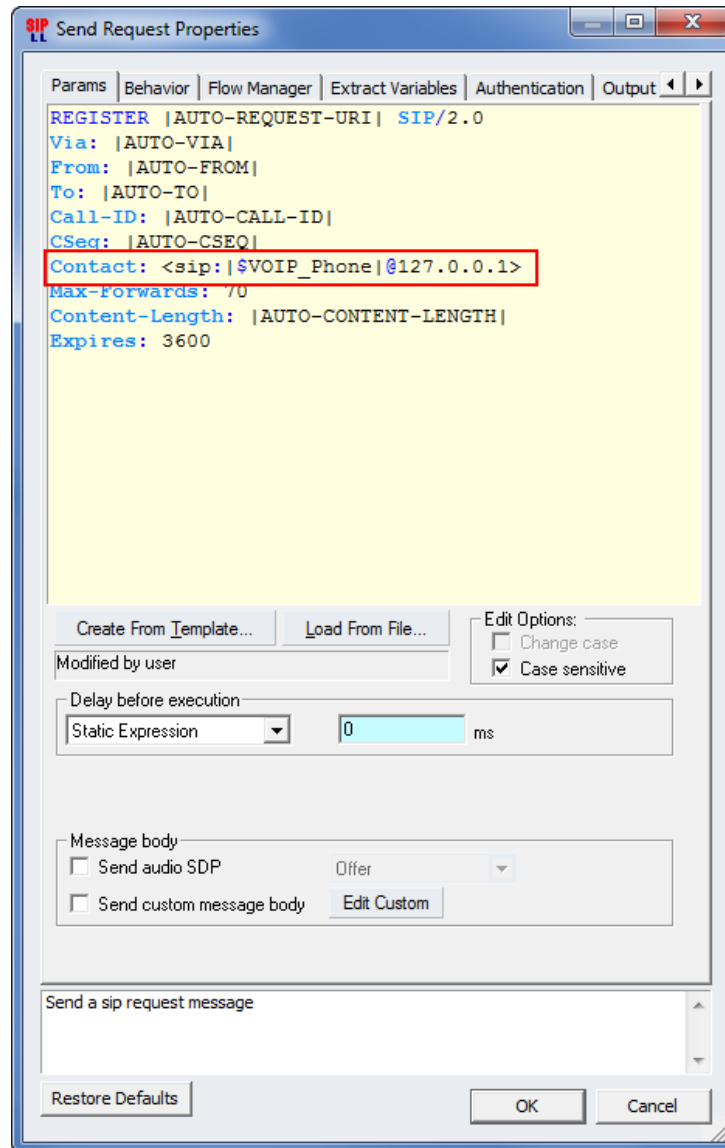


Figure 228. Modifying the Contact header information

Test Case: Telephony Denial of Service

7. In the **Behavior** tab of the **Send Request Properties**, enable the checkbox for **Extended Variables Support**. This enables the endpoint to evaluate the **\$VOIP_Phone** as a variable and construct the message using the phone numbers configured in the **Dial Plan** configuration tab. This checkbox should be enabled whenever variables are used and their values should be used in the SIP message.

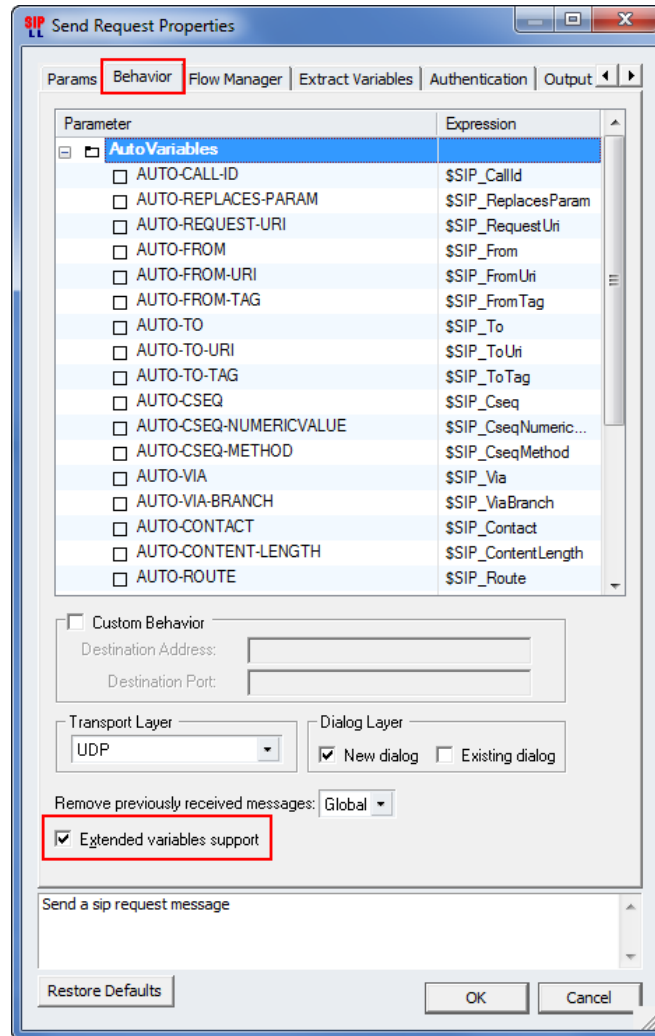
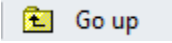


Figure 229. Enabling the Extended Variables Support for the current procedure

8. Change the Contact header information for both **Re-Send REGISTRATION** procedures and enable the checkbox for **Extended Variables Support**.
9. To change the state machine execution and create a larger diversity of test cases, re-link the output points of the existing procedures to others, and create different execution flows for the Registration process. Depending on the DUT configuration, the Registration may succeed or fail, and no malfunctions or security breaches should be encountered. As an example, after a few unsuccessful registration attempts, the DUT will should no longer

Test Case: Telephony Denial of Service

process any incoming request from that originating source IP to protect the stability of the system, and minimize any break-in attempts or identity probing.

10. After all changes have been made, return to the **Scenario Editor** in the **Scenario Channel #0** by pressing the  button from the controls ribbon.
11. Repeat Steps from 4 to 10 to add a new procedure to the call flow and modify the **Expires** header value to a number larger than $2^{32} - 1$. The DUT should report this large expire attempts as a security breach attempt or parser error. If no validation is in place, this case can be considered as failed and proper actions should be performed to minimize the risk during live operations. Additionally, other alphanumeric characters can be configured to probe the SIP parser engine stability for invalid values.

Test Case: Telephony Denial of Service

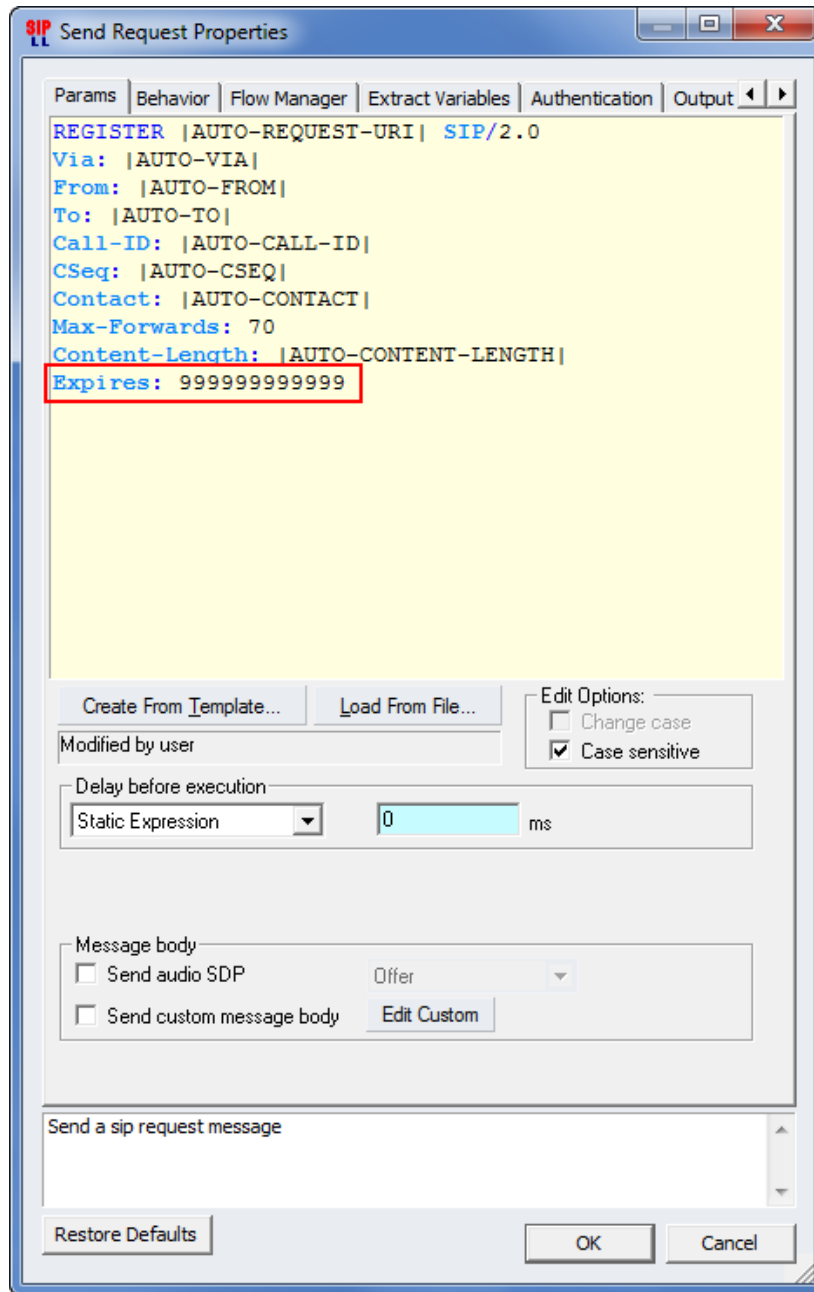


Figure 230. Configuring expiration value over the permitted range

Test Case: Telephony Denial of Service

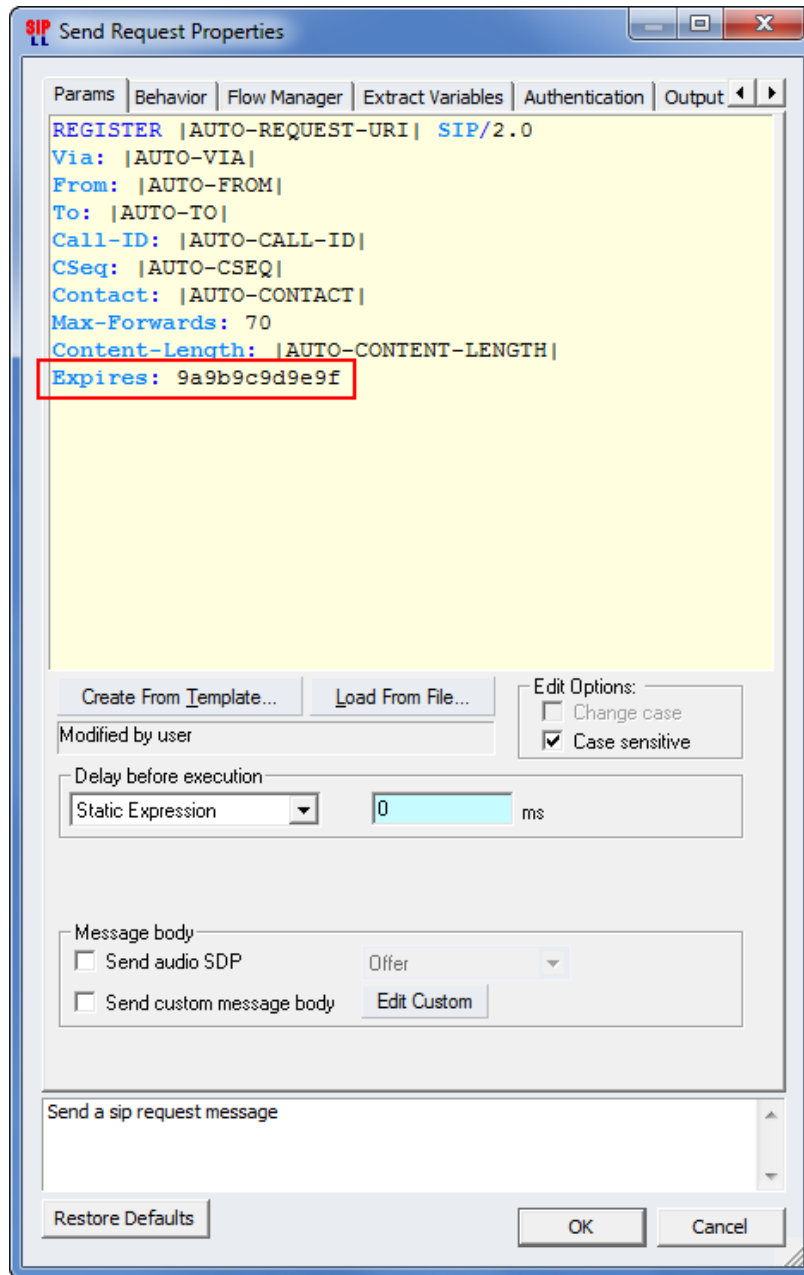


Figure 231. Configuring expiration header with alphanumeric characters

Test Case: Telephony Denial of Service

- Repeat steps from 4 to 10 to add a new procedure to the call flow and multiply the **To**, **From**, and **Contact** headers in the originating requests. Additionally, you may configure the header value to contain values from other variables. Consult the example below.

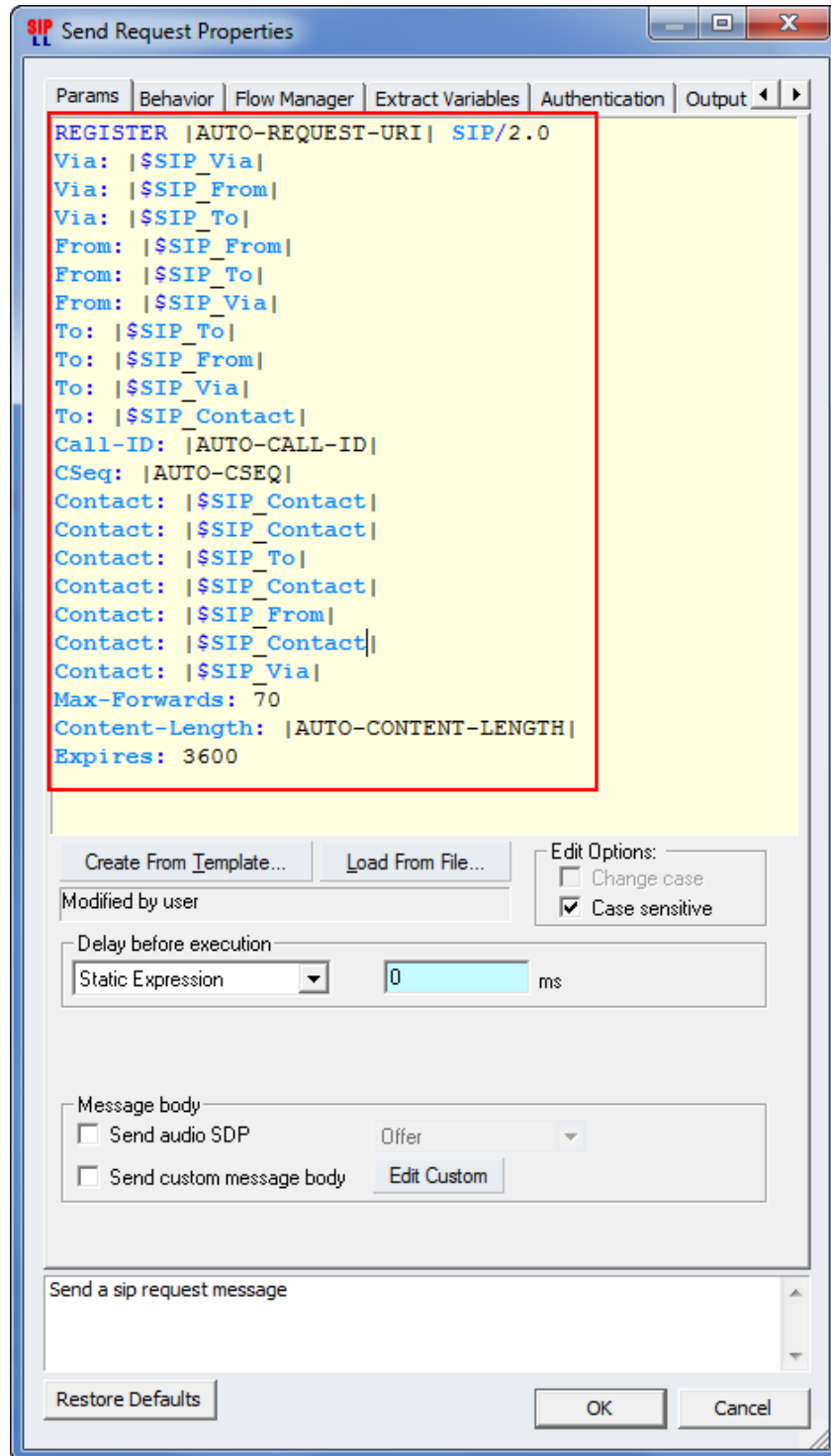


Figure 232. Configuring multiple headers

Test Case: Telephony Denial of Service

- To enable the automatic processing of the configured variables, in the **Properties** window select the **Extract Variables** tab and enable the checkboxes for the automatic variables used. Consult the example below as reference:

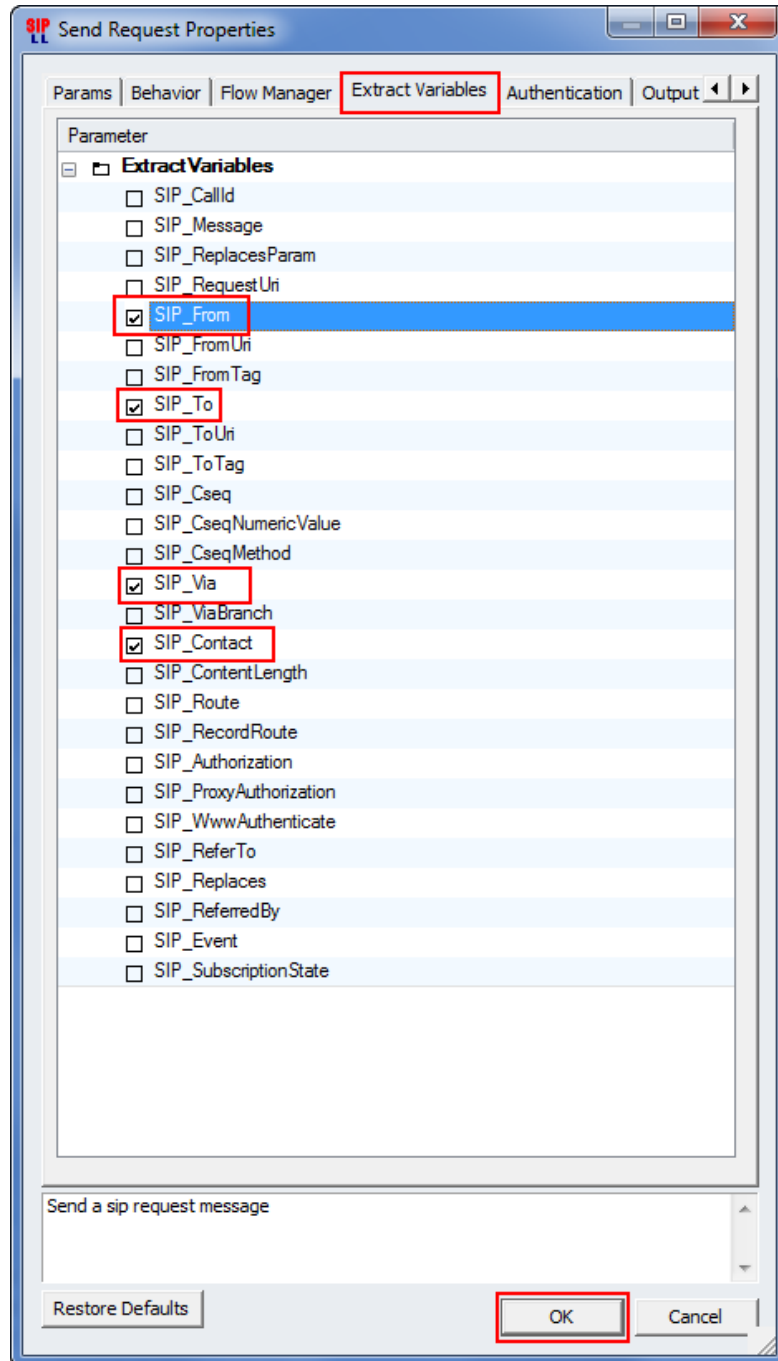


Figure 233. Enabling automatic variables processing

Test Case: Telephony Denial of Service

- Repeat steps from 4 to 10 to add a new procedure to the call flow and enable the offer of SDP information in the originating requests. In the **Request Properties** configuration tab, enable the checkbox of the **Send audio SDP** parameter. Set the option to *Offer* from the available drop down options. The DUT should ignore these types of requests with SDP information, and no stability issues should be noticed.

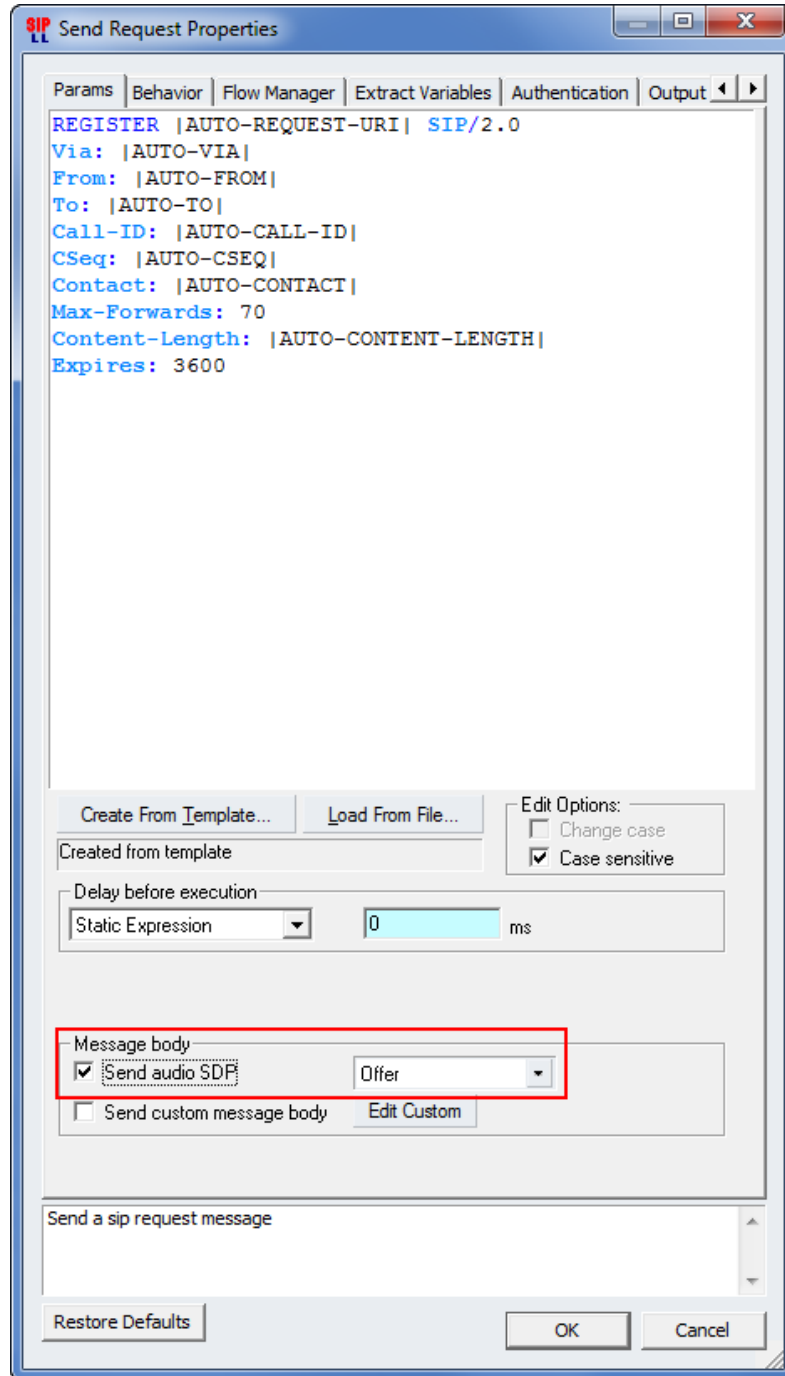


Figure 234. Configuring SDP offer in the originating request

Test Case: Telephony Denial of Service

- Repeat steps from 4 to 10 to add a new procedure to the call flow and add a large number of custom-defined headers. The reason is to validate the DUT capacity to properly parse the received messages and not compromise the security or stability. When the message size is large, the expected performance should slightly decrease and no critical errors should occur. If the custom headers defined are recognized by the DUT, these should be properly processed regardless of the place where these are found in the message.

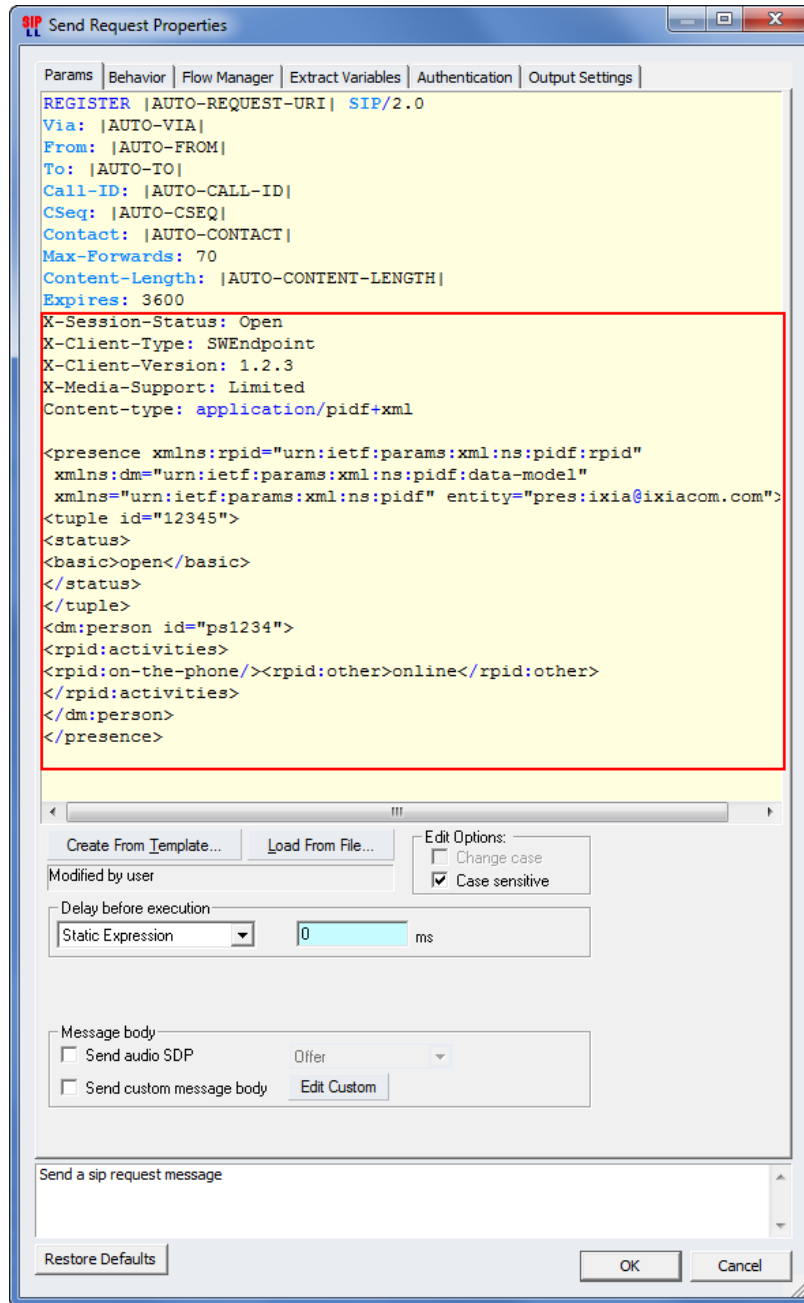



Figure 235. Configuring custom headers

Test Case: Telephony Denial of Service

16. After the above steps have been completed, link the added activities in the Scenario Editor to enable their execution during the run time. Any unlinked procedures will be ignored during traffic emulation by the IxLoad VoIPSIP peer activity. Using the  cursor, link the procedure's **OK** and **Error** output to the entry point of the next procedure. The last procedure's outputs should be linked to **STOP** similar to the screenshot below.

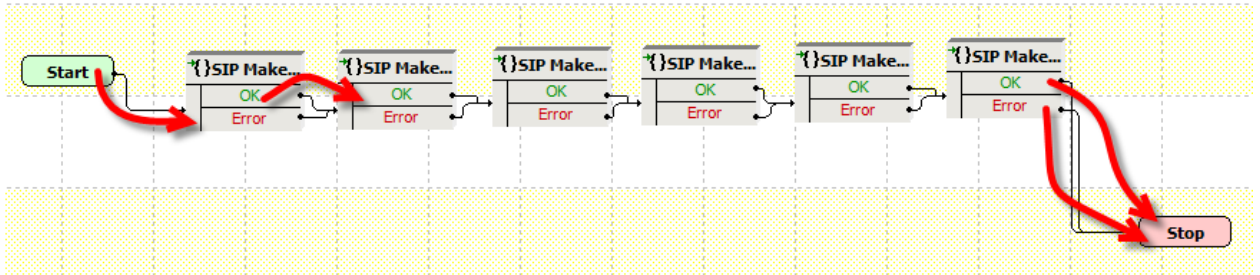


Figure 236. Linking all the configured procedures

Setting the VoIPSIP Peer Activity Parameters

17. Click the **Dial Plan** tab and set the **Source Phone Numbers** `505[0000-]` in this example; enable the checkbox **Override phone numbers from destination activity** and configure the desired destination phone numbers as targets for the TDoS. In this example configure `555(0000-]`.
18. Click the **SIP** tab and ensure that the **Enable signaling on this activity** check box is selected. Configure the originating **SIP port** as needed. This example will use the `[5060-]` default value. Increase the value for **Maximum message size on UDP** to 2500 bytes. This parameter controls the SIP buffer size for the transmitted packets. Configure the desired user name and password for the authentication of the endpoint. This example will use `ixia` for the **User name** and **Password**. Enable the checkbox **Use external server** and configure the DUT address or name. This example uses the following parameters:
- 19.

Table 87.

Parameter	Value
Server address	<code>20.20.20.200</code>
Server port	<code>5060</code>
Domain name or local IP	<code>20.20.20.200</code>
Outbound Proxy	<i>Checkbox enabled</i>
Registrar server	<i>Checkbox enabled</i>

Enable the checkbox **Override default destination domain name or host: port** and configure the destination IP or name of the DUT. This value is used in the construction of the SIP To header for the transmitted packets. Ignoring this value and leaving the default should construct

Test Case: Telephony Denial of Service

an improper message that the DUT should ignore. Omitting this value or configuring other values represents a new test case. This example will use the value *20.20.20.200*.

The screenshot shows the SIP configuration interface with the following settings:

- SIP:** Enable signaling on this activity (SIP Port: 5060-)
- Transport settings:**
 - Maximum message size on UDP: 2500
 - Override transport specified in scenario: UDP Only
 - TCP send immediate
 - Enable FQDN resolution
 - DNS expiration timeout: 60 seconds
 - Cache FQDN resolution
- Authentication UAC:**
 - User name: ixia
 - Password: ixia
 - AKA authentication settings: Select configuration: <None>
 - Edit configurations...
- Use external server:**
 - Use external server
 - Server address: 20.20.20.200
 - Server port: 5060
 - Domain name or local IP: 20.20.20.200
 - Outbound proxy
 - Registrar server
 - Auto register simulated user agents
 - Override registrar: IP:PORT
- Construction of SIP messages:**
 - Override default contact settings (Edit Contact ...)
 - Override default destination domain name or host:port (Domain name or Host:Port: 20.20.20.200)
 - Use Tel URI scheme for Source
 - Use Tel URI scheme for Destination
 - Use automatic headers (Auto headers ...)

Figure 237. SIP tab configuration example

20. Check whether the dispatching rule in **Override default dispatching rules** has the proper sequence of phone numbers in the **Formula for dispatching** field – *5551[0000-]* in this example.

Setting the Timeline and Objective

In this example, the DUT will receive all the malformed traffic configured in the test. Depending on the verbosity level of the logs, a large amount of information might be logged. It is recommended to start with lower traffic levels and increase after the initial debug has been performed and the stress test is ready for execution. The test objective will be configured to *100 Loops per Second*.

21. Click **Timeline & Objective** from the test configuration panel.
22. For the **VoIPSIPPeer1**, set the test **Objective Type** to **Loops Initiated per Second**.
23. Set the test **Objective Value** to *100*.

Test Case: Telephony Denial of Service

24. Set the **Sustain Time** to 1 hour.

Execute the Test

Map the port according to the test environment and run the test by clicking the  button.

Test Variables

Table 88. Test configuration

Parameter Name	Current Value	Additional Options
IP Type	IPv4	IPv6
Type of traffic	SIP signaling	Audio, Video, T38, Audio/Video RTCP traffic, Other SIP methods
Phone number in use	User defined	
Number of loops per second	User defined	
Call duration	User defined	
Mix with data protocols (for example, FTP, HTTP, Telnet)	Not included	Any combination of data protocols supported by IxLoad
Mix of call flows		successfully calls, canceled calls, unanswered calls, busy calls
Mix of call features		call forward, call transferred, call hold/retrieve

Results Analysis

The DUT shall be monitored for the following:

- Memory size, memory allocation/de-allocation issues while:
 - Phone numbers are added/deleted to/from the user table
 - Invalid SIP Translations are continuously detected and stability faults are not detected
- CPU usage
- Size of LOG files

Test Case: Telephony Denial of Service

The following questions provide guidelines on how to recognize specific problems during or at the end of the test execution:

- Has the test objective been achieved? Check the **Call Rates** view.

Table 89. Rate statistics

Statistic Name	Value	Questions
Loops Initiated per Second		<ol style="list-style-type: none"> Have the loops been attempted continuously at a constant call rate during the Sustain Time? How do the Loops initiated per second rate and the Responses Received rate compare to each other?
SIP Responses Received		

- Have any scenario loop failures been reported? Check the **Loops** statistics view.

Table 90. Statistics highlighting the pass/fail result based on flow execution

Statistic Name	Questions
Total Loops	<ol style="list-style-type: none"> Are the Successful Loops and Total Loops values equal? Have any Failed Loops, Aborted Loops, or Warning Loops been reported?
Successful Loops	
Failed Loops	
Aborted Loops	
Warning Loops	<p>Note: failed/aborted and warning loops highlight failures at the scenario level.</p>

Consult the information found in the **Event Viewer** that will indicate the reason of failure and possible actions to remediate the fault.

Time	Chassis;Card;Port	Event Title
05/08/2014 05:38:38.0...	10.205.17.114;1;12	Loop 290694: SIP Parser Error in: Unrecognized Header Line. Text: [</dm:person>
05/08/2014 05:38:38.0...	10.205.17.114;1;12	Loop 290694: SIP Parser Error in: Unrecognized Header Line. Text: [</presence>
05/08/2014 05:38:38.0...	10.205.17.114;1;12	Loop 290696: SIP Parser Error in: Expires Header. Text: [Expires: 999999999999
05/08/2014 05:38:38.0...	10.205.17.114;1;12	Loop 290697: SIP Parser Error in: Expires Header. Text: [Expires: 9a9b9c9d9e9f

Total Events : 1521899 Events In View : 20000

Figure 238. Event Viewer logged messages sample

- Has the test reported any errors? Once the DUT detects the TDoS attempt or the system is at the maximum processing limit, various errors might occur. Check the information displayed in **Errors** view.

Test Case: Telephony Denial of Service

Table 91. Statistics used to determine the SIP Registration rate

Statistic Name	Questions
Timeout errors	5. Are there any timeout errors or transport errors reported?
Transport errors	6. Are there any Call Flow errors reported? Could you identify the cause?
SIP Call Flow errors	
SIP Parser errors	7. Are there any increasing values for the Variable Extraction statistics reported? Verify if the proper variables are used to construct the test procedures.
SIP SDP errors	
SIP Extract Variables errors	
SIP Internal errors	

Conclusions

This test methodology provided details of how to emulate with IxLoad a Voice SIP endpoint initiating forged SIP Register messages towards the DUT in the attempt of a Telephony DoS with malformed messages. The scenario described in this chapter represents a configuration baseline and additional alterations can be performed to stress in multiple ways the stability and security of the system using other methods like INVITE, OPTIONS, SUBSCRIBE, ACK, BYE, or NOTIFY. The most challenging part of the configuration is determining the settings that have the most impact on the DUT, as a system failure during live operation can bring major penalties to the security of the voice infrastructure and business operations.

Contact Ixia

Corporate Headquarters
Ixia Worldwide Headquarters
26601 W. Agoura Rd.
Calabasas, CA 91302
USA
+1 877 FOR IXIA (877 367 4942)
+1 818 871 1800 (International)
(FAX) +1 818 871 1805
sales@ixiacom.com

Web site: www.ixiacom.com
General: info@ixiacom.com
Investor Relations: ir@ixiacom.com
Training: training@ixiacom.com
Support: support@ixiacom.com
+1 877 367 4942
+1 818 871 1800 Option 1 (outside USA)
online support form:
<http://www.ixiacom.com/support/inquiry/>

EMEA
Ixia Technologies Europe Limited
Clarion House, Norreys Drive
Maiden Head SL6 4FL
United Kingdom
+44 1628 408750
FAX +44 1628 639916
VAT No. GB502006125
salesemea@ixiacom.com

Renewals: renewals-emea@ixiacom.com
Support: support-emea@ixiacom.com
+44 1628 408750
online support form:
<http://www.ixiacom.com/support/inquiry/?location=emea>

Ixia Asia Pacific Headquarters
21 Serangoon North Avenue 5
#04-01
Singapore 5584864
+65.6332.0125
FAX +65.6332.0127
Support-Field-Asia-Pacific@ixiacom.com

Support: Support-Field-Asia-Pacific@ixiacom.com
+1 818 871 1800 (Option 1)
online support form:
<http://www.ixiacom.com/support/inquiry/>