



## Provisioning Guide

# Dialogic<sup>®</sup> BorderNet<sup>™</sup> Session Border Controller (SBC)

Release 3.8.1

June 2019

# Table of Contents

- 1. Introduction
  - 1.1 Purpose of this Document
  - 1.2 Glossary
  - 1.3 Contact Us
- 2. Overview
  - 2.1 Status Bar and Main Menu
  - 2.2 Menu Options
  - 2.3 GUI Functionality
- 3. System
  - 3.1 IP Configuration
    - 3.1.1 IP Addresses
    - 3.1.2 Ethernet Link
    - 3.1.3 VLAN Interface
    - 3.1.4 IP Routes
  - 3.2 EMS (Element Management System)
  - 3.3 User Management
    - 3.3.1 Users
    - 3.3.2 Assigning Roles
    - 3.3.3 Creating a User Account
    - 3.3.4 Editing a User Account
    - 3.3.5 Deleting a User Account
  - 3.4 Login Policy
    - 3.4.1 User Policies
    - 3.4.2 Password Policies
    - 3.4.3 Login Policies
  - 3.5 Changing Passwords
  - 3.6 Administration
    - 3.6.1 Deployment
    - 3.6.2 Network Time Protocol
    - 3.6.3 Domain Name Server (DNS)
    - 3.6.4 System Services
    - 3.6.5 System Information
    - 3.6.6 Licenses
    - 3.6.7 SNMP Trap Managers
    - 3.6.8 Email Configuration
    - 3.6.9 Audit Logs
    - 3.6.10 Configuring SDR
- 4. SIP Configuration
  - 4.1 Interface
  - 4.2 Peer
  - 4.3 Interface-Peer
  - 4.4 Parameter Profile
  - 4.5 Media Profile
  - 4.6 App Params
  - 4.7 SRTP Profile
  - 4.8 Registration
  - 4.9 Trunk Authentication
  - 4.10 WebRTC Support

- 5. H.323 Configuration
  - 5.1 Interface
  - 5.2 Peer
  - 5.3 Interface-Peer
  - 5.4 Parameter Profile
  - 5.5 Media Profile
  - 5.6 Global Parameters
- 6. Security Configuration
  - 6.1 Security Profile
  - 6.2 Access Control List
  - 6.3 Digital Certificates
  - 6.4 TLS Profiles
  - 6.5 IPsec Overview
  - 6.6 IPsec Manual Key Profiles
  - 6.7 IPsec IKE Profiles
  - 6.8 IPsec Policies
- 7. Policy Configuration
  - 7.1 Overview
  - 7.2 Advanced Policy
    - 7.2.1 Routing Parameters
    - 7.2.2 Treatment
  - 7.3 Rerouting
    - 7.3.1 External Route Server (SIP Redirect Server)
    - 7.3.2 Local Number Portability (LNP)
    - 7.3.3 Matrix
    - 7.3.4 ENUM
  - 7.4 Number Translation
    - 7.4.1 Number Translation on High Availability (HA) Deployments
    - 7.4.2 Number Translation and the Session Description Record (SDR)
  - 7.5 Directory Lookup
  - 7.6 Criteria Sets
  - 7.7 Time Band
  - 7.8 Global Variables
- 8. Common Features
  - 8.1 Static Routing
  - 8.2 Local DNS
  - 8.3 DNS Servers
  - 8.4 Service Profiles
    - 8.4.1 SIP-REC
    - 8.4.2 SIP Protocol Requirements
  - 8.5 Emergency Profiles
    - 8.5.1 Overview
    - 8.5.2 Emergency Call Configuration
    - 8.5.3 Creating a New Emergency Profile
  - 8.6 Codec Profiles
  - 8.7 Transcoding Profiles
    - 8.7.1 Transcoding Overview
    - 8.7.2 Transcoding Activation
    - 8.7.3 Transcoding Configuration
    - 8.7.4 Fax Transcoding
    - 8.7.5 DTMF Transcoding
    - 8.7.6 Creating a Transcoding Profile

- 8.8 Diameter Profile
  - 8.8.1 Diameter Overview
  - 8.8.2 Offline & Online Charging
  - 8.8.3 Diameter Rx Interface
  - 8.8.4 Configuring a Diameter Profile
- 8.9 Port Allocation Table
- 8.10 SIP Profilers
  - 8.10.1 Overview
  - 8.10.2 Conventions
  - 8.10.3 Creating a SIP Profiler
  - 8.10.4 Editing SIP Profilers
  - 8.10.5 Adding Rules
  - 8.10.6 Creating a SIP Profiler with XML Files
  - 8.10.7 Deleting SIP Profilers
  - 8.10.8 Profiler Document Hierarchy
  - 8.10.9 Profiler Document Structure
- 8.11 ISUP Profilers
- 8.12 Transcoding Gateways
- 8.13 Number Translation Profile
- 8.14 Bulk Provisioning
- 8.15 Customized LRBT
- 9. Definitions
  - 9.1 Trunk Groups
  - 9.2 Network Types
  - 9.3 SIP Connect
  - 9.4 Transport Protocol
    - 9.4.1 Supported Configurations of Transport Interworking
    - 9.4.2 UDP to TCP Automatic Transition
  - 9.5 SIP-I Support
  - 9.6 Traffic Policing
  - 9.7 Surrogate Registration
- 10. SIP Profiler Variables and Elements
  - 10.1 SIP Profiler Variables
    - 10.1.1 Examples
    - 10.1.2 Local Variables
    - 10.1.3 Transaction Variables
    - 10.1.4 Session Variables
  - 10.2 SIP Profiler Elements
    - 10.2.1 Group Elements
    - 10.2.2 Operator Elements
    - 10.2.3 Constant Data Elements
    - 10.2.4 SIP Data Elements
    - 10.2.5 Other Data Elements
    - 10.2.6 Action Elements
  - 10.3 Examples
    - 10.3.1 Retrieving and Modifying SIP Header/Parameter Values
    - 10.3.2 Retrieve a Header Parameter
    - 10.3.3 Adding New SIP Headers
    - 10.3.4 Insert a New Unknown SIP Header
    - 10.3.5 Deleting SIP Headers
    - 10.3.6 Adding a New SIP Header Parameter
    - 10.3.7 Deleting SIP Header Parameter

10.3.8 Retrieving and Storing Data in From/To Variables

10.3.9 Retrieving Data from Configuration Tables

10.3.10 Retrieving Data from IP Layer Fields

11. Session Detail Records

## Copyright and Legal Notice

Copyright © 2019 Dialogic Corporation. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Dialogic Corporation at the address provided below.

All contents of this document are furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation and its affiliates or subsidiaries ("Dialogic"). Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in certain safety-affecting situations. Please see <http://www.dialogic.com/company/terms-of-use.aspx> for more details.

Due to differing national regulations and approval requirements, certain Dialogic products may be suitable for use only in specific countries, and thus may not function properly in other countries. You are responsible for ensuring that your use of such products occurs only in the countries where such use is suitable. For information on specific products, contact Dialogic Corporation at the address indicated below or on the web at [www.dialogic.com](http://www.dialogic.com).

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic does not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic and no such licenses are provided except pursuant to a signed agreement with Dialogic. More detailed information about such intellectual property is available from Dialogic's legal department at 3300 Boulevard de la Côte-Vertu, Suite 112, Montreal, Quebec, Canada H4R 1P8. **Dialogic encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.**

Dialogic, Dialogic Pro, DialogicONE, Dialogic Buzz, PowerMedia, PowerVille, PowerNova, MSaaS, BorderNet, Brooktrout, Veraz, Cantata, TruFax, SnowShore, Eicon, NMS Communications, I-Gate, and ControlSwitch, among others as well as related logos, are either registered trademarks or trademarks of Dialogic Corporation and its affiliates or subsidiaries. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 3300 Boulevard de la Côte-Vertu, Suite 112, Montreal, Quebec, Canada H4R 1P8. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

The names of actual companies and products mentioned herein are the trademarks of their respective owners.

This document discusses one or more open source products, systems and/or releases. Dialogic is not responsible for your decision to use open source in connection with Dialogic products (including without limitation those referred to herein), nor is Dialogic responsible for any present or future effects such usage might have, including without limitation effects on your products, your business, or your intellectual property rights.

## Revision History

Revision	Release Date	Notes
3.5	December 2013	Release 3.2.0
3.6	February 2016	Release 3.3.0 - formatting and styling
3.7	February 2016	Release 3.4.0
3.8	November 2016	Updated for release 3.5.0
3.9	March 2017	Updated for release 3.6.0: <i>MaxAllowedInactiveMediaCallDuration</i> , <i>DNSQueryRetriesThreshold</i> , and <i>DNSQueryTimeout</i> parameters added to App Params. Added support for Near-end NAT, Replaces header, and Surrogate Registration. Advance Policy is added to the bulk provisioning, entity type option.
4.0	May 2017	Updated the Bulk Provisioning section
5.0	September 2017	Updated for release 3.7.0 - Added Local Operator ID Field in Interface/Peer configuration windows, for the P-Charging-Vector support. - Updated the Add Transcoding Profile window. - Added the Customized LRBT tab - Added the LRBT configuration to Service Profile - Added the SDR field 17 for LRBT - Added the OMR configuration to Media Profile, and to Peer Configuration windows - Updated the SDR field 148 for OMR - Updated the inactivity timer parameters - Added the REFER message handling in SIP Parameter Profile
6.0	March 2018	Update for release 3.7.5 - SIP UDP to TCP transition update Interface and peer provisioning - Added new section for Trunk Authentication - Added support for Number translation - Added new DTMF parameter to transcoding profile for DTMF transcoding - Added support for CA integration for TLS - Added provisioning for SIP-REC - Added External system DNS
7.0	September 2018	Updates to features for release 3.7.6
8.0	December 2018	Updates to features for release 3.8.0
9.0	June 2019	Updates to features for release 3.8.1

# 1. Introduction

## 1.1 Purpose of this Document

This document provides the information needed to configure the **BorderNet Session Border Controller (SBC)** after it is installed.

For product details, refer to the *BorderNet SBC Product Description* document.

## 1.2 Glossary

For the purposes of this document the following abbreviations apply:

Abbreviation	Meaning
DoS	Denial of Service
LBO	Local Break Out
LRBT	Local Ring Back Tone
OMR	Optimal Media Routing
SBC	Session Border Controller
SDR	Session Data Record
VoIP	Voice over IP

Table 1: Glossary

## 1.3 Contact Us

For a list of Dialogic locations and offices, please visit: <https://www.dialogic.com/contact.aspx>.



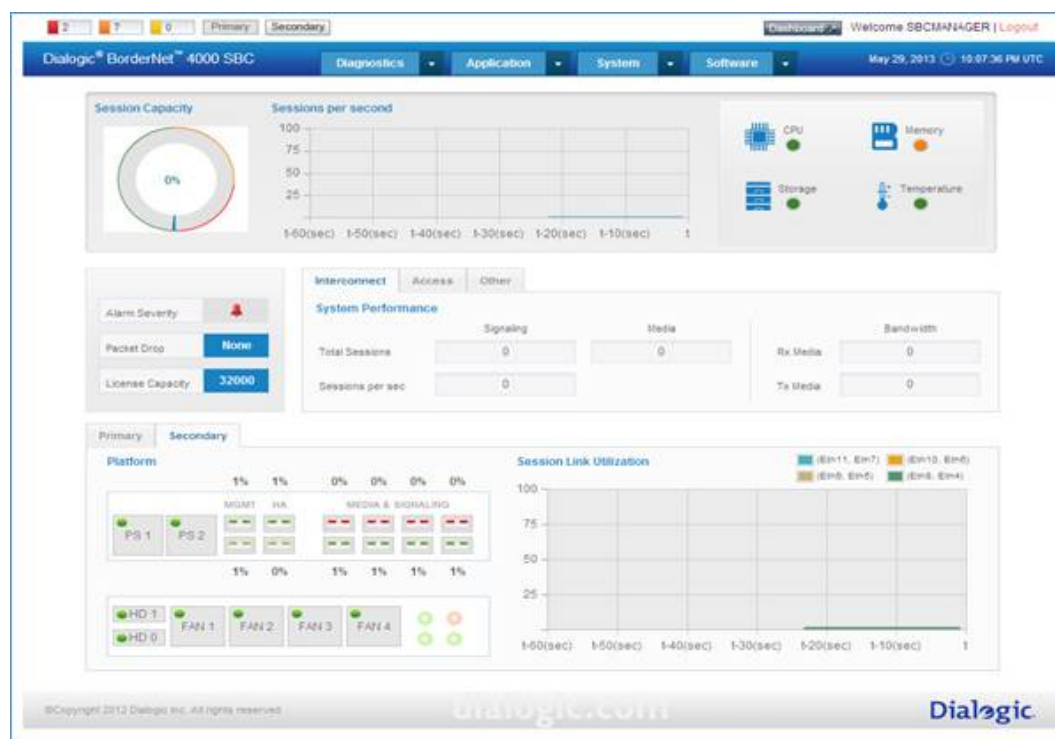
## 2. Overview

The **BorderNet SBC** provides call signaling, control and media termination in a VoIP network. It is deployed on the border of a network, managing the incoming and outgoing signaling and media traffic for service providers that require call session control and network security.

The BorderNet SBC provides an integrated Web UI that contains management for **Fault, Configuration, Accounting, Performance and Security (FCAPS)** functionality and monitoring capabilities to perform system management tasks, using the following browsers:

- Internet Explorer (v8,v9,v10) Note: IE 11 is not supported.
- Mozilla Firefox (v5 and above)
- Google Chrome

To access BorderNet SBC, enter the system management IP address assigned during the initial installation/deployment (ask the System Administrator for IP address). Upon login, the BorderNet SBC Dashboard is displayed.



The Dashboard provides an executive summary of system health and performance, including real-time information such as:

- Memory and CPU usage
- Available storage and system temperature
- Alarm severity, number of packets dropped, and license capacity
- Interconnect System Performance, including:
  - Total signaling and media sessions
  - Number of signaling sessions per second
  - Bandwidth of Rx and Tx media
- Access System Performance, including:
  - Number of successful and in-progress registrations
  - Number of registrations per second
- Primary and secondary platform information and session link utilization (applicable for 1U rack unit BorderNet SBC)

## 2.1 Status Bar and Main Menu

The **Status Bar** and **Main Menu** are displayed at the top of every screen. The **Status Bar** displays the number of outstanding **Critical**, **Major**, and **Minor** alarms in the system.

For HA deployment, the **Primary** and **Secondary** buttons indicate which system is active (the disabled button indicates the inactive system). The **Main Menu** provides drop-down lists for BorderNet SBC features and system information.



To return to the **Dashboard** view any time, click on the Dashboard icon at the top-right side of the window.


## 2.2 Menu Options

The BorderNet SBC provides four main menu options:








- **Diagnostics.** Provides access to tools that analyze system's health, such as alarms detection and tracing. For detailed information, see the *BorderNet SBC Maintenance Guide*.
- **Application.** Provides access to SIP Configuration, H.323 Configuration, Security Configuration, and Common.
- **System.** Provides access to IP configuration, User Management, Login Policy and Administration functions.
- **Software.** Provides information on the BorderNet SBC's installed software, upgrade, roll-back, backup and restore functionalities. For detailed information, see the *BorderNet SBC Maintenance Guide*.

## 2.3 GUI Functionality

The BorderNet SBC provides common navigation icons throughout the Web UI.

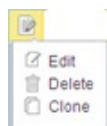
All the summary screens contain an **Action List**  icon next to each line, as shown in the **SIP Interface** screen below.

SIP Interface + Add New SIP Interface

Status	Name	Domain	Network Type	SIPconnect	Trust Level	Assoc Peers Only	Signaling IP	VLAN Name	Signal Port	Signal Protoc	Signal TOS	TGRP Context	Parameter Profile	Media Profile	Service Profile	Security Profile
	OV Spectra_Accel	SpectraLocal.c	Access-Loc	No	High	<input type="checkbox"/>	10.20.50.35	spectra	5080	UDP	0		Access-local	Access-local	Access-local	Access-local
	OV Spectra_Accel	spectraaccess	Access-Pul	No	High	<input type="checkbox"/>	10.20.50.35	spectra	5070	UDP-TC	0		default_acces	default_acces	default_acces	default_acces
	OV Spectra		Interconnec	No	High	<input checked="" type="checkbox"/>	10.20.50.35	spectra	5060	UDP-TC	255		Default	Default	Default	Default
	OV ovip-702-V4-A	local702	Access-Loc	No	High	<input type="checkbox"/>	10.70.10.3	OvIP-702	3040	UDP	0		Access-local	Access-local	Access-local	Access-local
	OV ovip-702-v4-A	public702	Access-Pul	No	High	<input type="checkbox"/>	10.70.10.2	OvIP-702	3030	UDP	0		default_acces	default_acces	default_acces	default_acces
	OV ovip-701-V4-A	local701	Access-Loc	No	High	<input type="checkbox"/>	10.70.10.3	OvIP-701	3040	UDP	0		Access-local	Access-local	Access-local	Access-local
	OV ovip-701-v4-A	public701	Access-Pul	No	High	<input type="checkbox"/>	10.70.10.2	OvIP-701	3030	UDP	0		default_acces	default_acces	default_acces	default_acces

There are two ways to modify summary screen line items:


1. Click the **Action List** icon to view the available options for that specific line item.



The above example has the option to **Edit** the properties of the line item, **Delete** the line item from the system, or **Clone** the item.

The **Clone** option provides the ability to make a copy of the existing record. This allows an Administrator to quickly copy an existing configuration and create a new one by only changing required and unique values.

1. Double-click the **line item** to open the **Edit** screen.

All detailed view screens contain an optional function in the form of a **Post-It** note  icon, as shown in the **Edit SIP Interface** screen above. Double-click the Post-It note to add specific information or comments about the data set entry.

→ To configure the BorderNet SBC:

1. Enter the IP address configured during installation.
2. Log in as either the System Administrator (for system configuration) or the Application Administrator (for application configuration).

See the BorderNet SBC *Quick Start Guide* for initial login instructions.

## 3. System

The **System** tab enables the management and configuration of the following:

- [IP Configuration](#)
- [User Management](#)
- [Administration](#)

### 3.1 IP Configuration

#### 3.1.1 IP Addresses

The BorderNet SBC supports up to 2048 IP addresses. Each IP address must be unique. Configured IP addresses are used as access IP addresses for signaling and media peers to send traffic toward the BorderNet SBC.

For HA systems, these IP addresses are floating and will be configured dynamically on the active platform.

##### 3.1.1.1 Configuring IP Addresses on VLANs

IP addresses can be configured on VLAN interfaces or directly on the session link. To configure an IP address via the VLAN interface, see the procedure for [Configuring VLANs](#).

---

**Note:**

A VLAN interface must have its own unique subnet. The VLAN subnet consists of the primary IP address and the subnet mask assigned to the VLAN. Overlapping address ranges and subnets are not allowed.

---

##### 3.1.1.2 Configuring IP Addresses on Session Links

VLANs are not mandatory for configuring IP addresses. Networks that do not use VLANs can configure the IP addresses on the session link directly. When the VLAN ID is zero, the BorderNet SBC allows multiple VLANs with the VLAN ID = 0. Multiple IP address subnets (IPv4 or IPv6) can be configured on the session links.

In this scenario, the VLAN interface configuration is still used, but the VLAN ID must be set to "0". This bypasses VLAN functionality and creates a non-VLAN object. IP addresses created on this object are configured directly on the session links. Multiple IP addresses can be configured on session links. Egress traffic from the session link is not tagged with the VLAN ID.

→ To configure an IP address directly on the session link:

1. From the **System** drop-down menu, select **VLAN Interface**.  
The **VLAN Interface** screen appears.
2. Click **Add New VLAN** to open the **Add VLAN** screen.
3. Set the **Status**.
  - To enable the session links upon creation, select **ON**. This is the default value.
  - To create the session links without enabling it, select **OFF**.
4. Enter a name into **VLAN Name** field.
5. Select the **Ethernet Link** from the drop-down menu.
6. Enter a VLAN ID.

If set to 0 the VLAN functionality is disabled and the IP addresses are allowed to be configured directly on the session link.

7. Select the **IP Address Type**:

- For **IPv4**, use the valid range (1-32) for IP and Subnet Mask addresses for the VLAN.
- For **IPv6**, use the valid range (1-128) for IP and Subnet Mask addresses for the VLAN.

8. Enter the **Primary IP** address.

9. The **Gateway IP** address is optional.

- To configure a Gateway, enter an IP address. This IP address will be used as the default Gateway for all egress traffic from this VLAN.
- To create a VLAN interface without a Gateway, leave the field empty.

10. Enter the **Subnet Mask**.

11. Enter **Secondary IP** addresses. Secondary IP addresses must be from the same subnet.

12. Click the green plus  icon to enter secondary IP addresses into the field below.

13. Click **Save** to create the session links.

### 3.1.2 Ethernet Link

The **Ethernet Link** window provides a graphical representation of the paired ports on the BorderNet SBC. Using this window, the operator can control the traffic from the link-level.

Ethernet Link						
	Status	Primary Eth Port	Sec. Eth Port	Port Type	Name	Auto Negotiation
		Eth5	Eth9	Session	Public_40	Yes
		Eth4	Eth8	Session	home_30	Yes
		Eth1	Eth2	HA	HA	Yes
		Eth0	Eth3	Management	Management	Yes

Each port pair shown above provides dedicated access to one of the following networks:

- Management network
- High Availability (HA) network
- Session network via session links

The port pairs are pre-configured in **Active-Standby** mode and cannot be modified. The primary port is configured as active, and the secondary port is configured as standby.

This configuration provides redundancy. In the event of an active link failure, the standby link takes over. Port pairs are managed as a single logical link and share the same properties to provide seamless switchovers when required.

### 3.1.2.1 Ethernet Link Types

The Ethernet link types are pre-configured within the system and cannot be modified. The following illustration shows the interface numbering:

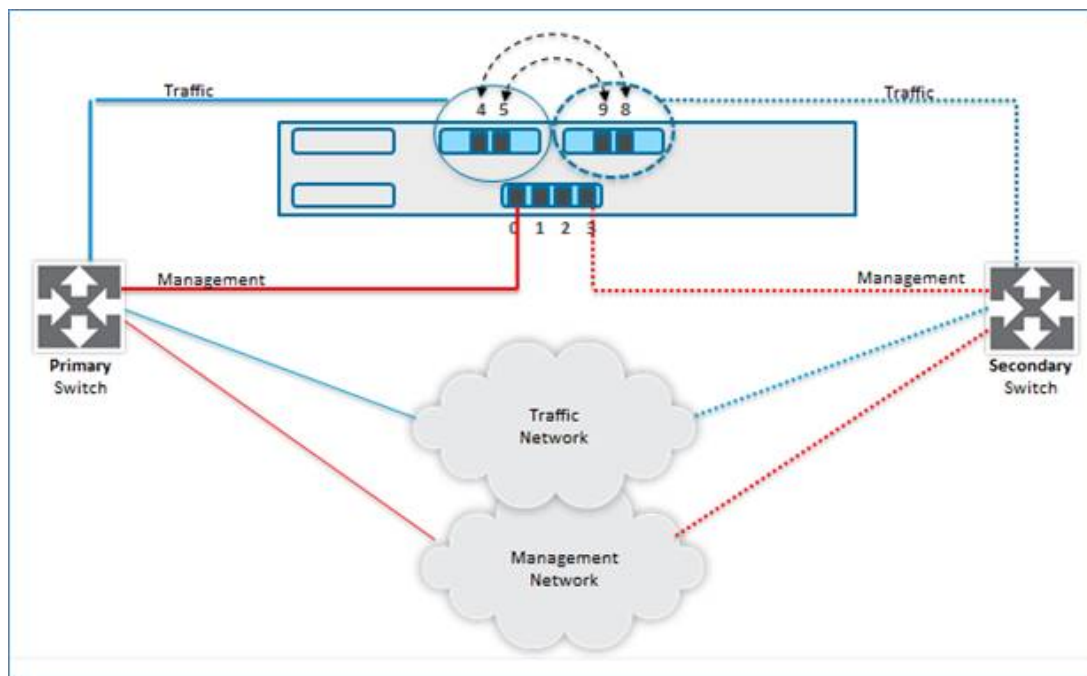


Figure 1: Ethernet Link Types

The BorderNet SBC provides three Ethernet link types:

Link	Description	Port Pairs
Management	Provides management traffic access, such as HTTP, FTP, NTP, SNMP, and Telnet/SSH.	Primary: Eth0 Secondary: Eth3
High Availability	Provides HA communication between two paired BorderNet SBC paired platforms and includes internal messaging communication and data synchronization.	Primary: Eth1 Secondary: Eth2

Link	Description	Port Pairs
Session	Used for session traffic; up to four session links can be configured at one time. Session links can also be configured for card-level redundancy.	Primary: Eth4 Secondary: Eth8 Primary: Eth5 Secondary: Eth9

### 3.1.2.2 Ethernet Link Configurations

The Application Administrator can activate or deactivate a link status for maintenance windows or to investigate suspicious traffic.

→ To deactivate a link from the Ethernet Link screen:

1. Slide the **On/Off** bar in the **Status** column of the desired link until the **Yes** tool tip appears (shown below).
2. Click to deactivate the link.

---

#### WARNING:

Deactivating a link is traffic-affecting.



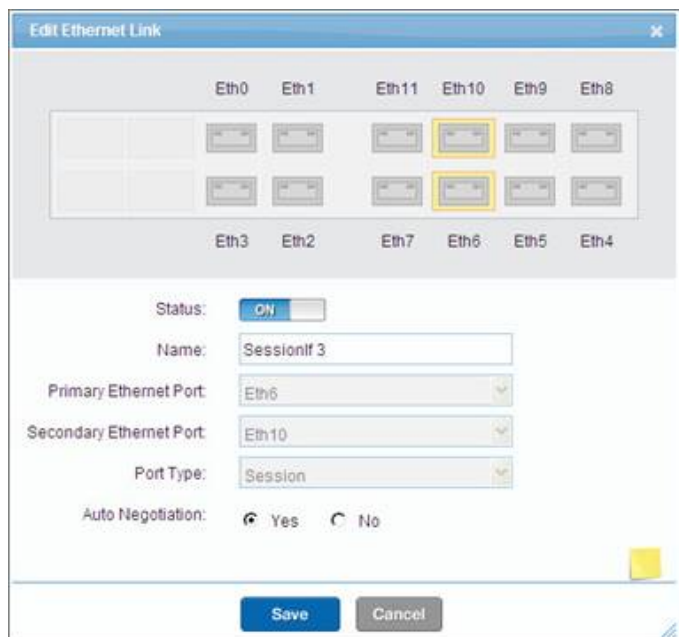
A confirmation window warns of the traffic impact.

---

3. Click **Confirm** to deactivate the link or **Cancel** to keep the link active.



**Auto-Negotiation** automatically locates the highest link speed available. By default **Auto-Negotiation** is set to **Yes**, as shown below.



In the event that a switch only accepts a specific speed, the Application Administrator can turn off **Auto-Negotiation** and manually set the speed and mode of the link:



Speed can be set to 10, 100, or 1000 Mbps. Duplex mode controls transmission and reception.

Full duplex allows transmission and receipt to occur at the same time. Half duplex allows transmission and receipt to occur one at a time.

## 3.1.3 VLAN Interface

### 3.1.3.1 Overview

The BorderNet SBC supports Overlapping IP addresses. In networks where peer IPs are private IPs (mainly in the enterprise domain) it is possible for two peers to have same IP (and port) but belong to different **Virtual Local Area Networks (VLANs)**.

BorderNet SBC uses the VLAN information already available in the system to provision these peers with overlapped IP and ports.

VLANs divide the physical LAN into separate broadcast domains. VLANs can span multiple switches, and the separate broadcast domains can also span multiple switches. VLANs help to save bandwidth by reducing broadcast traffic and increasing security because traffic can only be seen by intended hosts on the same VLAN.

The BorderNet SBC employs an IEEE standards-based **802.1Q** VLAN that tags each outgoing frame. Each session link configured with the VLAN acts as a trunk port and can carry traffic from multiple VLANs. Traffic on the VLAN is tagged with a configured VLAN ID, shown in the summary screen below.



VLAN Interface							
Status	Name	Ethernet Link	VLAN ID	IP Address	Primary IP	Subnet Mask	Secondary IPs
ON	Vlan_40_26	Public_40	998	IPv4	10.10.40.26	24	
ON	Vlan_30_26	home_30	999	IPv4	10.10.30.26	24	

For more information, see the BorderNet Session Border Controller Overlapping IP Feature Guide.

### 3.1.3.2 Network Architecture

The following diagram shows a set of peers that have overlapped IP addresses and ports, when the BorderNet SBC is configured with two interfaces with the same IP and port on different VLAN devices.

**Note:**

A VLAN device consists of the VLAN ID and the Session.

In the Network Architecture diagram below, traffic from Peer1 and Peer2 can be distinguished based on the VLAN device used (for example, S2.100 and S4.200).

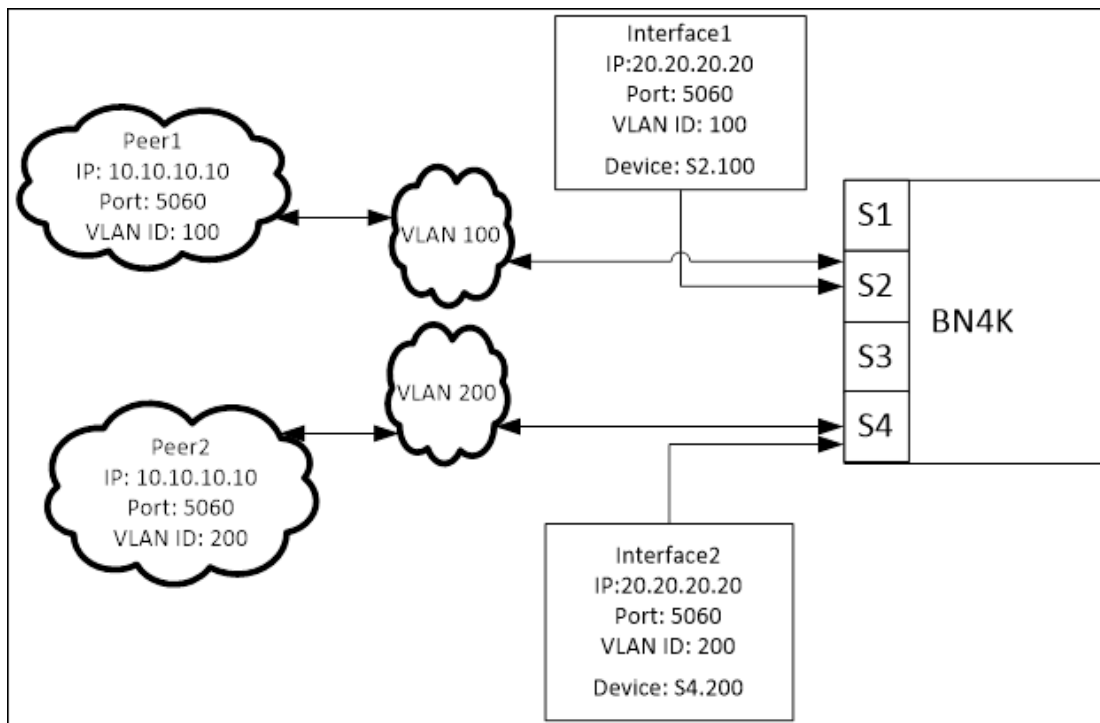


Figure 2: Network Architecture

There are three distinct parts to IP overlapping:

- IP Address
- VLAN ID
- Physical Session

To allow an overlapping IP address, either the VLAN ID or the Physical Session must be different.

For example:

IP Address	VLAN	Session
20.20.20.20	1	1
20.20.20.20	2	1
20.20.20.20	1	2
20.20.20.20	2	2

### 3.1.3.3 Functional Architecture

Any IP in the system should be viewed as a combination of the IP string and the VLAN device upon which it is attached, and the combination of those two should be unique in the system.

In this context:

- **Overlapping IPs** are those IP addresses whose string part is the same but the VLAN device part is different.
- An **Overlapping Subnet** consists of two subnets wherein one is a subset of the other (for example, 10.5.20.0/24 & 10.5.0.0/16, where the former is a subset of the latter).
- An **Overlapping IP Interface** consists of two interfaces (signaling or media) with same overlapping IP, same port and same transport protocol.
- An **Overlapping IP Peer** consists of two peers with same IP and port.

The following figure shows Overlapping IPs and their impact on the different BorderNet SBC entities, namely VLAN, Interface, Peer, Interface-Peer association and PAT.

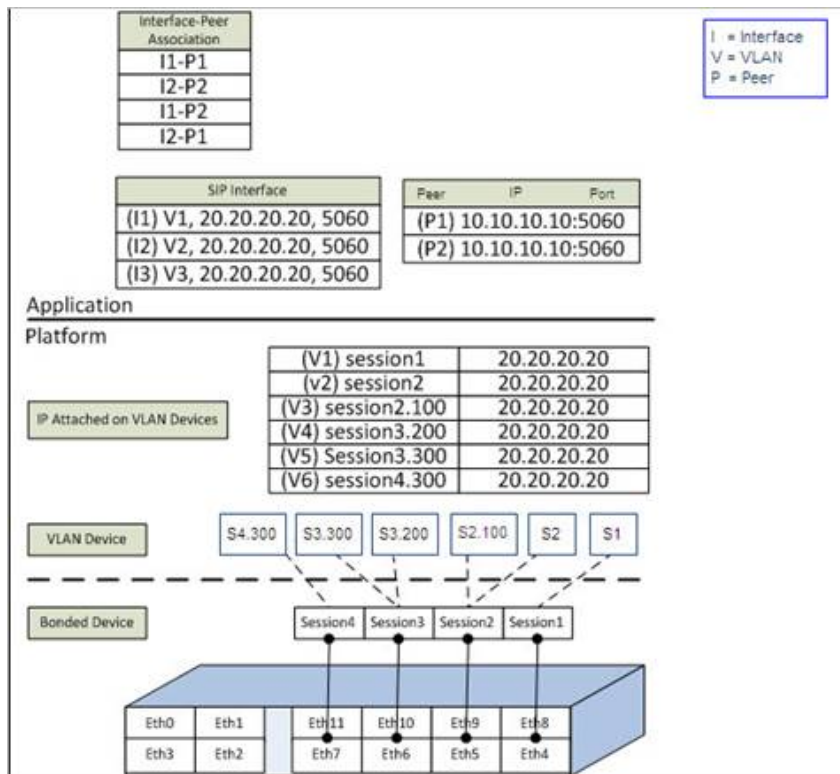


Figure 3: Overlapping IPs

On each bonded device, multiple VLAN devices have been created (V1, V2....V6). The same IP address-20.20.20.20-has been attached to all such VLAN devices.

PATs representing media interfaces can also be configured with overlapping IPs. VLAN device information will help distinguish between two PATs.

### 3.1.3.4 802.1Q VLANs

The BorderNet SBC uses VLANs to split the physical session network into different logical networks, which segregates and routes traffic to different Peering entities. VLANs are supported on session links, and multiple VLAN interfaces can be created on a link.

Similar to the Ethernet interfaces, VLAN interfaces are configured on port pairs, which creates a redundant VLAN interface to take over traffic in the event of an interface switch-over. The physical Ethernet link acts as a trunk and carries traffic tagged with the VLAN ID for multiple VLANs on a link.

### 3.1.3.5 Creating VLANs

The VLAN to IP Subnet association is 1-to-1. A VLAN can be associated to an IPv4 subnet or an IPv6 subnet.

→ To configure a VLAN interface on a session link:

1. Select **System** → **VLAN Interface**.

The **VLAN Interface** window opens.

2. Click **Add New VLAN** to add a new VLAN.

3. Set the **Status** of the VLAN.
  - To enable the VLAN upon creation, select **ON**. This is the default value.
  - To create the VLAN without enabling it, select **OFF**.
4. Enter the **VLAN Name**.
5. Select the **Ethernet Link** from the drop-down menu.
6. Enter a valid **VLAN ID**. Values range from **1 - 4095**. Traffic belonging to this VLAN interface will be tagged with this ID number.

---

**Note:**

The value "0" is only used to assign IP addresses without a VLAN. VLAN functionality is not present with value "0".

---


7. Select the **IP Address Type** (either IPv4 or IPv6).

---

**Note:**

All IP address fields (Primary, Secondary and Gateway IP Address) must be the same type.

---

8. Enter the **private (mandatory), and public (optional) Primary IP** addresses.
9. The **Gateway IP** address is optional.
  - To configure a Gateway, enter an IP address. This IP address will be used as the default Gateway for all egress traffic from this VLAN.
  - To create a VLAN interface without a Gateway, leave the field empty.
10. Enter the **Subnet Mask**. The subnet mask and primary IP address defines the subnet for the VLAN.
  - For **IPv4**, the valid range is 1-32 for the Subnet Mask.
  - For **IPv6**, the valid range is 1-128 for the Subnet Mask.
11. Enter **private (private) and public (optional) Secondary IP** addresses (must be from the same subnet).
12. Click the green plus  icon to enter secondary IP addresses into the field below.
13. Click **Save** to create the VLAN interface.

The VLAN interface is now added to the **Summary** screen.

---

**Note:**

The IPv6 address can be entered in any valid format. The BorderNet SBC converts and saves the IPv6 address in the most compressed re-presentation per RFC 5952. For example, 2001:DB8:0:0:0:0800:200C:4171 would be converted to 2001:DB8::800:200C:4171; these are the same IP address in different representations.

---

### 3.1.3.6 Editing a VLAN

The VLAN status can be activated (**ON**) or deactivated (**OFF**) at any time.

---

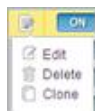
**WARNING:**

Deactivating a VLAN may be traffic-affecting.

---

→ To edit an existing VLAN:

1. Select **System** → **VLAN Interface**.  
The **VLAN Interface** window opens.
2. Select **Edit** from the note icon drop-down menu in the first column of the VLAN to be modified.



3. Make the desired changes. Note that some fields are not modifiable.
4. Click **Save** to modify the VLAN.

### 3.1.3.7 Deleting a VLAN

VLANs can be deleted via the WebUI. If the VLAN interface or the IP addresses configured on the VLAN interface are used in another configuration (for example, a SIP Interface), the VLAN cannot be deleted until the dependent configuration has been deleted.

→ To delete an existing VLAN:

1. Select **System** → **VLAN Interface**.  
The **VLAN Interface** window opens.
2. In the **Status** column of the VLAN to be deleted, slide the **Status Bar** to **OFF**.



A confirmation box appears.

3. Click **Confirm** to deactivate the VLAN.
4. Select **Delete** from the **Action List** icon drop-down menu in the first column of the VLAN to be deleted.
5. Confirm the deletion.  
The VLAN is removed from the **VLAN Interface** summary screen.

### 3.1.3.8 Cloning a VLAN

→ To clone an existing VLAN:

1. Select **System** → **VLAN Interface**.  
The **VLAN Interface** window opens.
2. Select **Clone** from the **Action List** icon drop-down menu in the first column of the VLAN to be cloned.
3. Make any desired changes.
4. Click **Save** to clone the VLAN.

### 3.1.3.9 Show Associated Entities

→ To view the associated entities to VLAN

1. Select a **VLAN** from the summary window and click the **Show Associated Entities** button.  
The **Select Associated Entry** window appears:



2. Select either **ACL** or **SIP Interface-Peer** and click **OK**.  
One of the following screens showing entries associated with the selected VLAN will appear.

Security Access Control List Summary									
Status	Name	Application	Action	IP Address Type	Remote IP	Remote Netmask	Remote Port	Local Entity	
ON	ovip-702-access-publi	SIP	Accept	IPv4	0.0.0.0	0	0	ovip-702-v4-access-pu	
OFF	ovip-Ing-2	SIP	Accept	IPv4	10.70.10.10	32	0	ovip_v4_client_702	

Status	Peer	Interface	Connectivity Feature
ON	ovip_v4_server_peer_21	ovip_v4_server_702	<input type="checkbox"/>
ON	ovip-702-Access-Local-Core	ovip-702-V4-Access-Local	<input type="checkbox"/>
ON	ovip_v4_server_peer_2_top	ovip_v4_server_702	<input type="checkbox"/>
ON	ovip_v4_client_peer_2_top	ovip_v4_client_702	<input type="checkbox"/>
ON	ovip_v4_server_peer_2	ovip_v4_server_702	<input type="checkbox"/>
ON	ovip_v4_client_peer_2	ovip_v4_client_702	<input type="checkbox"/>

The **Port Allocation** window lists the **VLAN Name** field as well.

## 3.1.4 IP Routes

IP Routes on the BorderNet SBC can be configured as destination routes to either an IPv4 destination address or an IPv6 destination address. This is done via a Gateway IP address that is from the same route type (IPv4 or IPv6) as the destination address of the route being configured.

IP Routes have the following properties:

- Destination IP Address - can be a host, network or subnet address.
- Subnet Mask
- Gateway IP Address
- Type of Service (TOS)
  - Values range from 0 - 255.
  - The TOS byte is included in the IP header and is used while matching the Route entry.
  - This field must contain a valid **TOS (IPv4)** or **Traffic Class (IPv6)** value.
- Metric
  - Values range from 1-255.
  - The Metric is used to prioritize routes.

An **IP Route** status can be enabled (**ON**) or disabled (**OFF**).

---

### Note:

IPv6 routing requires a license. Contact Dialogic Technical Support for licensing information.

---

### 3.1.4.1 Adding an IPv4 Route

The following procedure shows how to add an IPv4 Route.

1. From the **System** drop-down menu, select **IPv4 Route**.  
The **IPv4 Route Configuration** screen appears.

Status	Destination IP Address	Gateway IP Address	Subnet Mask	TOS	Metric
ON	10.3.175.0	10.20.0.1	24	0	1
ON	10.20.20.84	10.20.0.1	32	0	1

2. Click the **Add New IPv4 Route** button in the upper right portion of the screen.  
The **Add IPv4 Route** screen appears.

3. Select the IPv4 Route **Status**.
  - To enable the IPv4 Route upon creation, select **ON**.
  - To create the IPv4 Route without enabling it, select **OFF**. This is the default value.
4. Enter the following information:
  - **Destination IP Address** Destination IP Address
  - **Subnet Mask** (values are between 0 - 32) Subnet Mask (values are between 0 - 32)
  - **Gateway IP Address** Gateway IP Address
5. Enter the **TOS** value.  
By default, this value is 0. When configured, the TOS byte in the IP header is matched with the TOS value configured in the route entry for route selection.
6. To modify the TOS value, double-click in the TOS field.  
The **Edit TOS Bit Values** screen appears.

7. Select the **Differentiated Service Point Code (DSCP)** from the **DSCP Mode** drop-down menu.

---

**Note:**

The Class/DSCP Value is automatically populated with corresponding selections based on the DSCP Mode selected. The TOS Value is automatically populated with the appropriate value based on the Class/DSCP Value selected.

---

Possible mode selections and values include:

DSCP Mode	Corresponding Class/DSCP Values	TOS Value Range
Best Effort	Default PHB	0
Transparent	0xFF	255
AF: Assured Forwarding	AF11, AF12, AF13, AF21, AF22, AF23, AF31, AF32, AF33, AF41, AF42, AF43	40 - 152
EF: Expedited Forwarding	EF	184
CS: Class Selector	CS1, CS2, CS3, CS4, CS5, CS6, CS7	32 - 224
NS: Non Standard	1 - 63	4 - 252

1. Click **Save** to add the selected **TOS Value** to the **IP Route**.
2. Enter the priority **Metric** (values are between 0 - 255).
3. Click **Save** to add the **IPv4 Route** to the system.

### 3.1.4.2 Adding an IPv6 Route

The following procedure shows how to add an **IPv6 Route**.

1. From the **System** drop-down menu, select **IPv6 Route**.

The **IPv6 Route Configuration** screen appears.

Status	Destination IPv6 Address	Gateway IPv6 Address	Subnet Mask	Traffic Class	Metric
ON	fc01:6161:6161:5150::	fc01:8181:8181:6c42::1	64	0	1

2. Click the **Add New IPv6 Route** button in the upper right portion of the screen.

The **Add IPv6 Route** screen appears.

3. Select the IPv6 Route **Status**.
  - o To enable the IPv6 Route upon creation, select **ON**.
  - o To create the IPv6 Route without enabling it, select **OFF**. This is the default value.

4. Enter the following information:

- o **Destination IP Address** Destination IP Address
- o **Subnet Mask** (values are between 1 - 128)
- o **Gateway IP Address** Gateway IP Address

5. Enter the **Traffic Class** value. By default, this value is 0.

When configured, the **Traffic Class** byte in the IPv6 header is matched with the **Traffic Class** value configured in the route entry for route selection.

6. To modify the **Traffic Class** value, double-click in the **Traffic Class** field.

The **Edit TOS Bit Values** screen appears.

7. Select the Differentiated Service Point Code (DSCP) from the **DSCP Mode** drop-down menu.

---

**Note:**

The Class/DSCP Value is automatically populated with corresponding selections based on the DSCP Mode selected. The Traffic Class Value is automatically populated with the appropriate value based on the Class/DSCP Value selected.

---



Possible mode selections and values include:

DSCP Mode	Corresponding Class/DSCP Values	TOS Value Range
Best Effort	Default PHB	0
Transparent	0xFF	255
AF: Assured Forwarding	AF11, AF12, AF13, AF21, AF22, AF23, AF31, AF32, AF33, AF41, AF42, AF43	40 - 152
EF: Expedited Forwarding	EF	184
CS: Class Selector	CS1, CS2, CS3, CS4, CS5, CS6, CS7	32 - 224
NS: Non Standard	1 - 63	4 - 252

1. Click **Save** to add the selected **TOS Value** to the **IP Route**.
2. Enter the priority **Metric** (values are between 0 - 255).
3. Click **Save** to add the **IPv6 Route** to the system.

## 3.2 EMS (Element Management System)

The **EMS** is an external element which enables the management of multiple BorderNet SBCs from a single application.

The EMS can run up to 100 BorderNet SBCs which are of V3.7.0 or higher, using a Restful API and SNMP external interfaces.

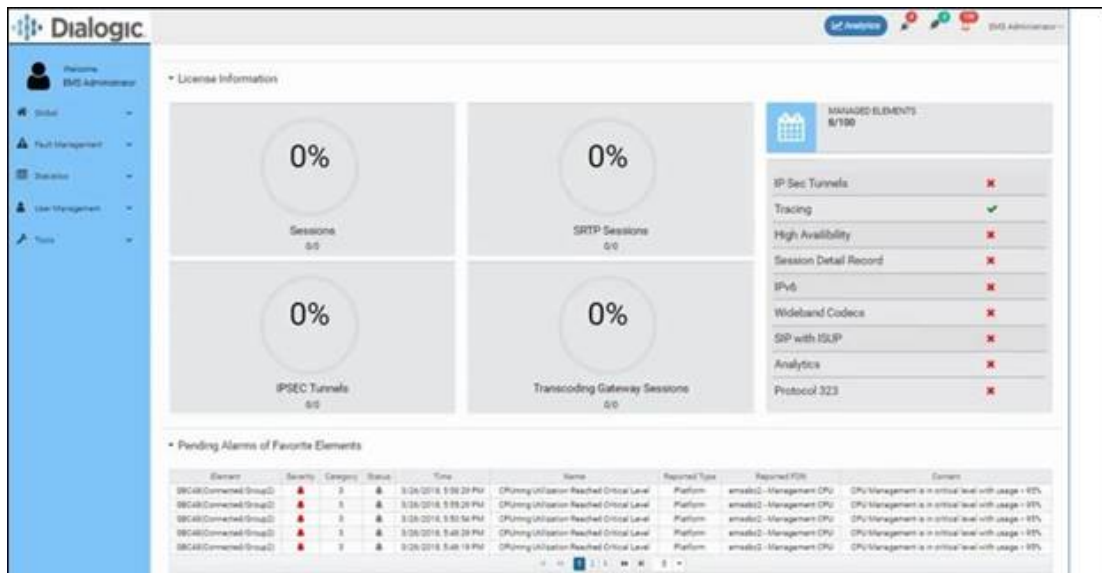
The EMS enables the following capabilities:

- Dashboard and Topology view
- Fault Management (alarms both current and history)
- Statistics
- User management
- Centralized license management
- Analytics
- Debugging tools and call tracing
- Provisioning and software management (future releases only)

The EMS includes the following default users:

User Name	Password	Role Name
emsadmin	emsadm	Administrator
emsuser	emsusr	Normal operator
emsquery	emsqry	Viewer only user

When you login to the EMS you will see the main GUI as follows:



From the left-hand panel of the GUI you can access numerous configuration possibilities including the following:

- Global - dashboard format including license information, pending alarms, managed elements.
- Topology - map view of site locations.
- Application Parameters - numerous fields including analytics, communication, LDAP, logs etc.
- Fault Management - table of pending alarms and their properties including event history.
- Statistics - charts showing CPU usage, traffic, system stats, interfaces, peers.
- User Management - table showing local users, logged in users, audit logs, roles.
- Tools - server information, license codes etc.
- Provisioning - numerous fields pertaining to the provisioning status of the EMS including SIP configuration, policy configuration and others.

## 3.3 User Management

### 3.3.1 Users

The System Administrator controls access to the BorderNet SBC. To add a user to the system, the System Administrator [creates a new user account](#). The account status is either **enabled** to give the user access to the system or **disabled** to create the account but deny the user access to the system.

The System Administrator can disable a user account at any time. If a user is logged in, the session will continue until the user logs out, and then the account is disabled.

### 3.3.2 Assigning Roles

The BorderNet SBC provides six pre-defined user roles with associated privileges, described in the following table.

Role	Privileges
------	------------

Role	Privileges
System Administrator	<ul style="list-style-type: none"> <li>Manages system configuration</li> <li>Monitors deployment, system services and system information</li> <li>Configures NTP and licenses</li> <li>Manages user access</li> </ul>
Application Administrator	<ul style="list-style-type: none"> <li>Manages applications (IP, SIP, H.323 and security)</li> <li>Configures routes, application profiles, SNMP traps and Email</li> <li>Configures VLANs and IP addresses</li> </ul>
Provisioning User	<ul style="list-style-type: none"> <li>Creates and associates interfaces and peers</li> <li>Configures static routing and associates application profiles</li> </ul> <p><b>Note:</b> This role cannot configure VLANs and IP addresses for home interfaces.</p>
Trace Manager	Provides access to tracing through the Wireshark tracing tool.
Query User	Provides read-only access to general information
Security Auditor	Provides system-wide access and the ability to review every action performed in the system on a user-by-user basis <b>Note:</b> This role provides read-only access to the system information.

Table 2: BorderNet SBC User Roles

A System Administrator can assign one or more roles to a single user and deactivate specific roles if desired. Currently, user roles cannot be modified or deleted. Customizable roles will be available in a later release.

---

**Note:**

A maximum of four roles can be assigned to an individual user.

---

The **Provisioning User** role is a subset of the **Application Administrator** privileges and does not need to be expressly assigned to a user with Application Administrator privileges. All roles have read access, and the **Query User** role does not need to be assigned in conjunction with other roles. This eliminates the need for more than four roles to be assigned to any given user.

### 3.3.3 Creating a User Account

→ To create an account for a new user named John Smith:

- From the **System** drop-down menu, select **Users**.  
The **Users Summary** screen appears.
- Click the **Add New User** button in the top right corner of the screen.
  - The **Add User** screen appears.

The screenshot shows the 'Add User' dialog box. It includes the following fields and controls:

- Username:** Text input field.
- First Name:** Text input field.
- Last Name:** Text input field.
- Is Deletable:** Radio buttons for Yes and No.
- New Password Needed:** Radio buttons for Yes and No.
- Status:** Drop-down menu currently showing 'ENABLED'.
- Roles:** A section with two lists:
  - Available Role:** A list box containing APPLICATION\_ADMIN, SYSTEM\_ADMIN, SECURITY\_AUDITOR, TRACE\_MANAGER, and LI\_ADMIN.
  - Selected Roles:** A list box containing QUERY\_USER and PROVISION\_USER.
  - Buttons: 'Add >>' and '<< Remove' between the two lists.
- Password:** Text input field.
- Re-enter Password:** Text input field.
- Buttons:** 'Save' and 'Cancel' at the bottom.

2. Enter the following user information:
  - User Name: **jsmith**
  - First Name of the user: **John**
  - Last Name of the new user: **Smith**
  - Note that the **Is Deletable** field is disabled. Currently, all users can be deleted.
  - Select the appropriate radio button for **New Password Needed**: **Yes** assigns the password to the user for the initial login only. The user will be required to change the password after the initial login. **No** assigns the password to the user for the duration of the expiry period.
3. Select the user's **Status** from the drop-down list:
  - **Enabled** allows the user to access the system.
  - **Disabled** creates a new user account but does not allow the user to access the system.
4. From the **Available Role** list, select the desired role(s) and use the **Add>>** button to move each role to the **Selected Roles** list.
5. Set the password for the user's initial login:
  - Enter the password in the **Password** box. After the initial login, this password can be changed by either the user or the System Administrator.
  - Re-enter the user's initial login password in the **Re-enter Password** box.
6. Click **Save**.

John Smith now appears in the **Users Summary** screen, highlighted below.

Users Summary							
	User Name	First Name	Last Name	New Password Needed	Status	Created By	Creation Time
	<input type="text"/>	<input type="text"/>	<input type="text"/>				
	SBCQUERY	SBC	Query	No	ENABLED	INTERNAL	2010-12-31T13:00:00
	SBCUSER	SBC	User	No	ENABLED	INTERNAL	2010-12-31T13:00:00
	SBCMANAGER	SBC	Manager	No	ENABLED	INTERNAL	2010-12-31T13:00:00
	SYSMANAGER	System	Manager	No	ENABLED	INTERNAL	2010-12-31T13:00:00
	SECMANAGER	Security	Auditor	No	ENABLED	INTERNAL	2010-12-31T13:00:00
	SBCTRACE	Trace	Manager	No	ENABLED	INTERNAL	2010-12-31T13:00:00
	JSMITH	John	Smith	No	ENABLED	SYSMANAGER	2012-01-04T09:34:41

### 3.3.4 Editing a User Account

→ To edit a user account:

1. From the **System** drop-down menu, select **Users**.
2. From the **Users Summary** screen:
  - Either double-click the **User Name**.
  - Or click the **Note** icon to the left of the **User Name** and select **Edit** from the drop-down menu.
3. Enter the changes into the **Edit User** screen.

**Edit User**

Username:

First Name:

Last Name:

Is Deletable:  Yes  No

New Password Needed:  Yes  No

Status:

Roles:

Available Role: APPLICATION\_ADMIN, SYSTEM\_ADMIN, SECURITY\_AUDITOR, TRACE\_MANAGER, LI\_ADMIN

Selected Roles: QUERY\_USER, PROVISION\_USER


Password:

Re-enter Password:

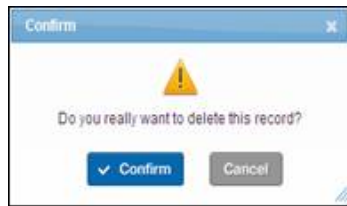
4. Click **Save** to save the changes.

### 3.3.5 Deleting a User Account

→ To delete a user account:

1. From the **System** drop-down menu, select **Users**.
2. From the **Users Summary** screen, click the **Note**  icon to the left of the **User Name** and select **Delete** from the drop-down menu.

A confirmation request message appears.



3. Click **Confirm** to delete the record.  
The user is deleted from the **Users Summary** screen.

## 3.4 Login Policy

The System Administrator has the option of setting login policies based on security needs. Policies can be set for creating a user, setting a password, and managing runtime.

To access the **Login Policy Configuration** screen, log in as a System Administrator and select **Login Policy** from the **System** drop-down menu.

Login Policy Configuration					Save
Policy List					
Policy Name	Policy Value	Default Value	Enabled?		
USERNAME_MINIMUM_LENGTH	Default	4	4	<input type="checkbox"/>	
USER_INITIAL_LOGIN_EXPIRE	Default	9999	9999	<input type="checkbox"/>	
PASSWORD_MINIMUM_LENGTH	Low	1	6	<input type="checkbox"/>	
PASSWORD_MINIMUM_LENGTH_ADMIN	Low	1	6	<input type="checkbox"/>	
PASSWORD_CONTENT_USERNAME	No	100	100	<input type="checkbox"/>	
PASSWORD_SIMILAR_CHECK	Default	1	1	<input type="checkbox"/>	
PASSWORD_REPEATING_CHARACTERS	Default	20	20	<input type="checkbox"/>	
PASSWORD_MINIMUM_LOWCASE	Default	0	0	<input type="checkbox"/>	
PASSWORD_MINIMUM_UPPERCASE	Default	0	0	<input type="checkbox"/>	
PASSWORD_MINIMUM_NUMBER	Default	0	0	<input type="checkbox"/>	
PASSWORD_MINIMUM_PUNCTUATION	Default	0	0	<input type="checkbox"/>	
PASSWORD_EXPIRING_PERIOD	Default	9999	9999	<input type="checkbox"/>	
PASSWORD_GRACE_PERIOD	Low	30	30	<input type="checkbox"/>	
PASSWORD_GRACE_LOGINS	Low	3	6	<input type="checkbox"/>	
FAILED_LOGIN_ATTEMPTS	High	3	9999	<input type="checkbox"/>	
DELAY_LOGIN_INTERVAL	Low	0	1	<input type="checkbox"/>	

Each policy has four pre-set values:

- Default value
- Low security
- Medium security
- High security

For example, to set the minimum length of a password to the highest level of security, select **High** from the **Policy Value** drop-down menu. The value **8** is automatically populated in the corresponding text field.

The pre-set values cannot be modified.

→ To customize a value:

1. Select **Other** from the **Policy Value** drop-down menu.
2. Enter the desired value in the corresponding text field.
3. Click **Save** to keep the changes to the **Login Policy Configuration** table.

---

**Note:**

The **Enabled** box must be checked to apply the policy values to the BorderNet SBC.

---

### 3.4.1 User Policies

Policies can be set to control user creation.

User Policy	Description	Default
<b>USERNAME_MINIMUM_LENGTH</b>	Specifies the minimum number of characters in a user name.	4
<b>USER_INITIAL_LOGIN_EXPIRE</b>	Indicates the number of days the user has to perform the initial login after the account is created. The account will be disabled if the initial login does not occur before the expiration date.	9999

### 3.4.2 Password Policies

Policies can be set to control password creation, maintenance, and expiry.

Password Policy	Description	Default
<b>PASSWORD_MINIMUM_LENGTH</b>	Specifies the minimum number of characters that must be in the password.	6
<b>PASSWORD_MINIMUM_LENGTH_ADMIN</b>	Specifies the minimum number of characters that must be in the password for System Administrator accounts.	6
<b>PASSWORD_CONTENT_USERNAME</b>	Verifies that the user name is not contained in the password.	No
<b>PASSWORD_SIMILAR_CHECK</b>	Specifies a minimum number of characters that must be different from the previous password.	1
<b>PASSWORD_REPEATING_CHARACTERS</b>	Indicates the maximum number of times the same character can be repeated in a password.	20
<b>PASSWORD_MINIMUM_LOWERCASE</b>	Indicates the minimum number of lowercase letters that must be in a password.	0
<b>PASSWORD_MINIMUM_UPPERCASE</b>	Indicates the minimum number of uppercase letters that must be in a password.	0
<b>PASSWORD_MINIMUM_NUMBER</b>	Indicates the minimum number of numeric characters that must be in a password.	0
<b>PASSWORD_MINIMUM_PUNCTUATION</b>	Indicates the minimum number of special characters that must be in a password.	0
<b>PASSWORD_EXPIRING_PERIOD</b>	Indicates the number of days in which a password expires.	9999
<b>PASSWORD_GRACE_PERIOD</b>	Indicates the number of days a user has to login in after the password has expired.	60

Password Policy	Description	Default
PASSWORD_GRACE_LOGINS	Specifies the number of logins that can occur within the password grace period.	6

### 3.4.3 Login Policies

Policies can be set to control login attempts.

Login Policy	Description	Default
FAILED_LOGIN_ATTEMPTS	Indicates the number of consecutive failed login attempts allowed before the account is locked or disabled.	9999
DELAY_LOGIN_INTERVAL	Indicates how many minutes an account will be locked after exceeding the failed login attempt policy.	1

For example, a System Administrator wants to ensure that user names have the highest level of security, passwords contain a minimum of three numeric characters, and all users are locked out following three unsuccessful login attempts. The following procedure shows how the System Administrator would set these values.

- From the **System** menu, select **Login Policy**.  
The **Login Policy Configuration** screen appears.
- Locate the **USERNAME\_MINIMUM\_LENGTH** policy row.
  - In the **Policy Value** column, select **High** from the drop-down menu.
  - The number **6** automatically populates the corresponding field.
  - In the **Enabled** column, click the check box.
- Locate the **PASSWORD\_MINIMUM\_NUMBER** policy row.
  - In the **Policy Value** column, select **Other** from the drop-down menu.
  - Enter the number **3** in the corresponding field.
  - In the **Enabled** column, click the check box.
- Locate the **FAILED\_LOGIN\_ATTEMPTS** policy row.
  - In the **Policy Value** column, select **High** from the drop-down menu.
  - The number **3** automatically populates the corresponding field.
  - In the **Enabled** column, click the check box.
- Click the **Save** button in the upper right corner of the screen to apply the updated login policy configuration.

The screenshot shows the 'Login Policy Configuration' screen with a 'Save' button in the top right. Below the title is a 'Policy List' table with the following columns: Policy Name, Policy Value, Default Value, and Enabled?. The table contains 17 rows of policies. Several rows are highlighted with red boxes: USERNAME\_MINIMUM\_LENGTH (Policy Value: High, Default Value: 4, Enabled: checked), PASSWORD\_MINIMUM\_NUMBER (Policy Value: Other, Default Value: 0, Enabled: checked), FAILED\_LOGIN\_ATTEMPTS (Policy Value: High, Default Value: 9999, Enabled: checked), and DELAY\_LOGIN\_INTERVAL (Policy Value: Low, Default Value: 1, Enabled: checked). Other policies include USER\_INITIAL\_LOGIN\_EXPIRE, PASSWORD\_MINIMUM\_LENGTH, PASSWORD\_MINIMUM\_LENGTH\_ADMIN, PASSWORD\_CONTENT\_USERNAME, PASSWORD\_SIMILAR\_CHECK, PASSWORD\_REPEATING\_CHARACTERS, PASSWORD\_MINIMUM\_LOWERCASE, PASSWORD\_MINIMUM\_UPPERCASE, PASSWORD\_MINIMUM\_PUNCTUATION, PASSWORD\_EXPIRING\_PERIOD, and PASSWORD\_GRACE\_PERIOD.

Policy Name	Policy Value	Default Value	Enabled?
USERNAME_MINIMUM_LENGTH	High	4	<input checked="" type="checkbox"/>
USER_INITIAL_LOGIN_EXPIRE	Default	9999	<input type="checkbox"/>
PASSWORD_MINIMUM_LENGTH	Low	0	<input type="checkbox"/>
PASSWORD_MINIMUM_LENGTH_ADMIN	Low	0	<input type="checkbox"/>
PASSWORD_CONTENT_USERNAME	No	No	<input type="checkbox"/>
PASSWORD_SIMILAR_CHECK	Default	1	<input type="checkbox"/>
PASSWORD_REPEATING_CHARACTERS	Default	20	<input type="checkbox"/>
PASSWORD_MINIMUM_LOWERCASE	Default	0	<input type="checkbox"/>
PASSWORD_MINIMUM_UPPERCASE	Default	0	<input type="checkbox"/>
PASSWORD_MINIMUM_NUMBER	Other	0	<input checked="" type="checkbox"/>
PASSWORD_MINIMUM_PUNCTUATION	Default	0	<input type="checkbox"/>
PASSWORD_EXPIRING_PERIOD	Default	9999	<input type="checkbox"/>
PASSWORD_GRACE_PERIOD	Low	60	<input type="checkbox"/>
PASSWORD_GRACE_LOGINS	Low	6	<input type="checkbox"/>
FAILED_LOGIN_ATTEMPTS	High	9999	<input checked="" type="checkbox"/>
DELAY_LOGIN_INTERVAL	Low	1	<input type="checkbox"/>



BorderNet SBC now requires that user names have the highest level of security, passwords contain a minimum of three numeric characters, and that all users are locked out following three unsuccessful login attempts.

---

**Note:**

This change will not apply to existing passwords. Users will be prompted to update passwords with the new policy requirements when they attempt to change the password or when the password expires.

---

## 3.5 Changing Passwords

When a user account is created, the System Administrator assigns an initial password.

Users can select **System > Change Password** to change their own passwords. The System Administrator can also change user passwords by [editing the user's account](#).

By default, passwords must be a minimum of 6 characters. The System Administrator can use login policies to apply additional requirements to passwords, such as setting a minimum number of special characters a password must contain.

## 3.6 Administration

The System Administrator is responsible for monitoring and configuring the SBC platform.

### 3.6.1 Deployment

The **Deployment** window allows the user to view and switch between standalone and high availability deployment modes.

Geo-Redundancy enables the deployment of the BorderNet SBC in High Availability mode where each platform/instance (primary and secondary) is located on two different networks or sites.



Figure 4: BorderNet in Geo-Redundancy Mode

There is no restriction with regards to the locations of the BorderNet SBCs. This enables more complex deployments where each BorderNet entity has its own set of IP addresses that can be on a totally different network. Therefore Geo-Redundancy allows each BorderNet SBC on a High Availability deployment to be located in cities or countries thousands of miles apart from each other.

---

**WARNING:**

Redeployment is traffic-affecting, and should be performed during a scheduled maintenance window.

---

→ To modify a deployment mode:

1. Select **System → Deployment**.

The **System Deployment Information** window opens. In the example below the system has been deployed as a standalone system:

2. Click on the **Redeploy** button to modify the deployment configuration.

1. In this case the deployment will turn to **High Availability (HA)** mode.

**System Deployment Information**

Platform Serial Number : V666287338  
License Request ID : AF9FBF237D562D09E4A0D348D0C9F4A0 529EBD69D47082D45EB3D49643013A85

Deployment Type : HA  
Designated Role : Primary

**Active Platform Details**

Hostname : susbc1  
Utility IP : 192.168.201.19  
Netmask : 24  
Gateway IP : 192.168.201.253  
System Management IP : 192.168.201.21  
Inter-Task/HA-Link IP : 192.168.200.220  
Inter-Task/HA-Link Netmask : 24

**Standby Platform Details**

Hostname : susbc2  
Utility IP : 192.168.201.20  
Inter-Task/HA-Link IP : 192.168.200.221

**Redeploy**

2. In the following window enter the new deployments parameters for both primary and secondary platforms.

3. Click **Start**.

**System Deployment Information**

Platform Serial Number : V666287338  
License Request ID : AF9FBF237D562D09E4A0D348D0C9F4A0 529EBD69D47082D45EB3D49643013A85

Deployment Type : HA  
Designated Role : Primary  
License File : Choose File No f...sen

**Primary Platform Details**

Hostname : susbc1  
Utility IP : 192.168.201.19  
Netmask : 24  
Gateway IP : 192.168.201.253  
System Management IP : 192.168.201.21  
Inter-Task/HA-Link IP : 192.168.200.220  
Inter-Task/HA-Link Netmask : 24

**Secondary Platform Details**

Hostname : susbc2  
Utility IP : 192.168.201.20  
Inter-Task/HA-Link IP : 192.168.200.221

**Start** **Cancel**

**Note:**

Refer to *BorderNet SBC Installation and Deployment Guide* for deployment procedures

## 3.6.2 Network Time Protocol

Date and time can be configured locally or synchronized with the NTP server via the **Network Time Protocol (NTP)**. To synchronize the system, up to two NTP server IP addresses can be configured.

The date and time is set during the initial turn-up of the system. BorderNet SBC auto-adjusts the clock according to the DST of the set time zone. The default time zone is UTC (GMT).

---

**WARNING:**

Changing the date, time, or time zone may affect traffic and may require a system restart. It is recommended to change these settings during the initial turn-up of the system or during a scheduled maintenance window.

---

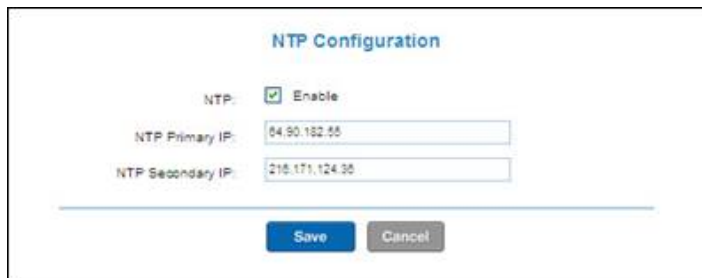
### Configuring the Time Zone

1. From the **System** drop-down menu, select **Administration > NTP** to view the **NTP Configuration** screen, shown below.

2. Click **Time Zone**.  
The **Select Time Zone** screen appears.
3. Select a time zone from the drop-down menu.

4. Select a continent and region from the drop-down menus.

5. Click **Save** to set the time zone.  
To synchronize the time with the NTP, an NTP should be configured.
6. Check the **NTP Enable** box.  
The **NTP Configuration** screen appears.



7. Enter the primary and secondary IP addresses from Public NTP Time Servers (for example, 64.90.182.55 for New York City, NY and 216.171.124.36 for San Jose, CA).
8. Click **Save** to configure the system.

---

**Note:**

Once configured, the NTP Server configuration is applied system-wide, and both HA BorderNet SBC platforms are synchronized to the configured NTP servers.

---

9. To set a new date and time, select a date from the calendar.
10. Set the time in HH:MM:SS and select AM or PM from the drop-down menu.
11. Click **Save**.

## 3.6.3 Domain Name Server (DNS)

BorderNet SBC uses a DNS to translate domain names to IP addresses. There is also a DNS cache feature which lowers queries to external domain name servers.

### 3.6.3.1 System External DNS

The external DNS configuration option configures a name server in the system `/etc/resolv.conf` file. It is then used by system services requiring an external DNS server for resolving.

For each name server entry added by the BorderNet SBC, a remark in the format of `;" added by Dialogic"` is appended, in order to indicate this is an entry which was added by the SBC. When the BorderNet SBC service is restarted or booted, then all of the current entries with the `;" added by Dialogic"` remark are deleted, and the `/etc/resolv.conf` file is then updated with the entries as present in the system external DNS configuration screen.

---

**Note:**

System external DNS is not replacing the 'DNS servers' & 'Local DNS' configuration as provisioned in

**Application→Common→DNS**. While system external DNS is used by general non-SIP system services, the 'DNS servers' & 'Local DNS' configuration are used by the BorderNet application to resolve SIP URIs of SIP messages.

---

### System External DNS Configuration

The screenshot shows a configuration window for 'System External DNS Configuration'. It features the following fields and controls:

- Status:** A toggle switch currently set to 'ON'.
- Name:** A text input field containing 'ExternalDNS'.
- IP Address Type:** A dropdown menu currently set to 'IPv4'.
- Primary DNS Server:** A text input field containing '1.2.3.4'.
- Alternate DNS Server(s):** A list box containing the entry '5.6.7.8,53'. To the right of the list box are a red minus sign, a green plus sign, and up/down arrow icons.
- Buttons:** 'Save' and 'Cancel' buttons are located at the bottom of the window.

- Status. Turn the system external DNS service On/Off. When the 'Off' state is chosen then all entries with a "; added by Dialogic" remark are deleted from '/etc/resolv.conf' file. When the 'On' state is selected all the entries listed are added to the '/etc/resolv.conf' file, appended with the "; added by Dialogic" remark.
- Name. General name for the DNS configuration.
- IP Address type. Select 'IPv4' to enter an IP version 4 type address. Select 'IPv6' to enter an IP version 6 type address.
- Primary DNS Server. Enter an IP address of an external DSN server to be added to the '/etc/resolv.conf' file.
- Alternate DNS Server(s). Extra servers to be added to the '/etc/resolv.conf' file, located after the Primary DNS server.

### 3.6.3.2 DNS Cache Service

An "in memory" cache, called the **DNS cache**, is designed to reduce queries to external DNS servers, thereby increasing the efficiency of the call setup process and reducing the load of queries on the operator's network. It does this by caching DNS request records for a period of time after each response.

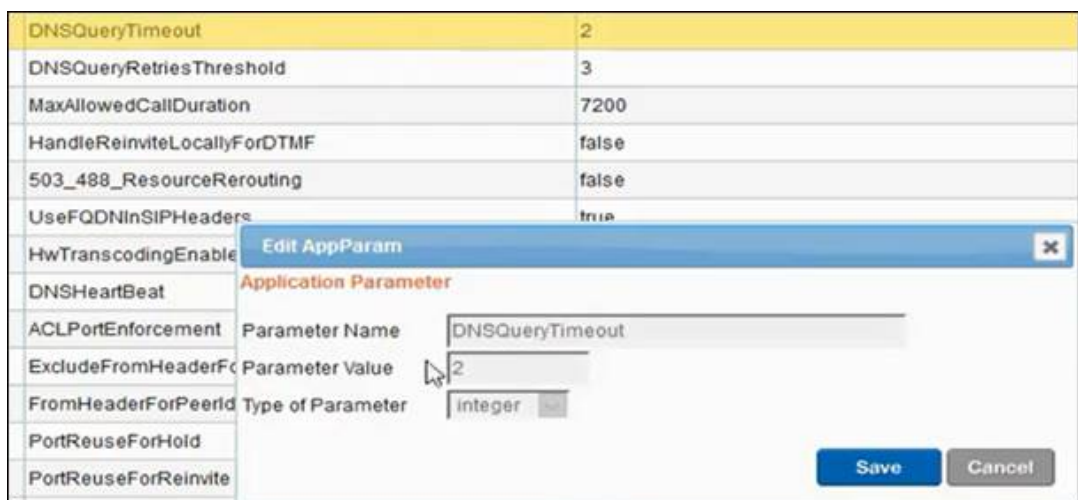
BorderNet therefore resolves the issue of domain name to IP address translation by first querying the DNS cache and using this result to initiate the call. If there is a negative response from the DNS cache the DNS query is then forwarded to the provisioned DNS server instead. If it fails to respond an SNMP trap is issued.

An alarm will be raised after the query default value is exceeded and/or the number of non-responded queries has passed a configured threshold. At first a minor severity alarm will be raised but this is also customizable.

This is accessed from the **Application** tab in the GUI. Select **App Params** under the **SIP Configuration** menu as shown below.



These values can be edited, for example the default value for the **DNS Query Timeout** is 2 seconds and the **DNS Query Retries Threshold** value is 3 retries.



BorderNet first sends the DNS requests to the first server in the list, and then to subsequent servers if there is no response. The returned result is inserted into the cache with the **TTL (Time to Live)** and then used to initiate the call. The TTL value is set by the administrator and is usually less than 24 hours.

The DNS cache is an internal entity which maintains resource records with an associated TTL and it deletes them after the TTL expires. The DNS cache does not sync to a standby platform as this would cause an unnecessary processing load.

The DNS cache screen displays a tabular view which can be exported to Excel. A list of abbreviations used in this display is shown here.

<b>A</b>	Address Record – IPv4
<b>AAA</b>	IPv6 address record
<b>APL</b>	Address Prefix List
<b>CAA</b>	Certification Authority Authorization
<b>CIDR</b>	<b>Classless Inter-Domain Routing.</b> CIDR notation is a compact representation of an IP address and its associated routing prefix. The notation is constructed from an IP address, a slash (/) character, and a decimal number. The number is the count of leading 1 bits in the routing mask, traditionally called the network mask. The IP address is expressed according to the standards of IPv4 or IPv6. Example: 192.168.100.14/24 represents the IPv4 address 192.168.100.14 and its associated routing prefix 192.168.100.0, or equivalently, its subnet mask 255.255.255.0, which has 24 leading 1-bits. the IPv6 block 2001:db8::/48 represents the block of IPv6 addresses from 2001:db8:0:0:0:0:0:0 to 2001:db8:0:ffff:ffff:ffff:ffff:ffff.
<b>NAPTR</b>	Name Authority Pointer, type of RR in the DNS Widely used for SIP service. In combination with SRV allows chaining of Domains/URIs.
<b>RR</b>	Resource Record
<b>SRV</b>	Service Record, type of RR in the DNS

Following is a list of definitions used in the DNS cache.

<b>BIND</b>	Berkeley Internet Name Domain Server
<b>DNS</b>	<b>Domain Name Server:</b> Naming system for computers, services, or any resource connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities.
<b>Domain Name</b>	Identification string that defines a realm of administrative autonomy, authority or control within the Internet. Domain names are formed by the rules and procedures of the Domain Name System (DNS). Any name registered in the DNS is a domain name.
<b>Subdomain</b>	Domain names are organized in subordinate levels (subdomains) of the DNS root domain, which is nameless. The first-level set of domain names are the top-level domains (TLDs), including the generic top-level domains (gTLDs), such as the prominent domains com, info, net, edu, and org, and the country code top-level domains (ccTLDs).
<b>FQDN</b>	<b>Fully Qualified Domain name:</b> Domain name that is completely specified with all labels in the hierarchy of the DNS, having no parts omitted. Labels in the Domain Name System are case-insensitive.
<b>URL</b>	<b>Uniform Resource Locators:</b> Commonly informally termed a web address (a term which is not defined identically) is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it. A URL is a specific type of Uniform Resource Identifier (URI), although many people use the two terms interchangeably. A URL implies the means to access an indicated resource, which is not true of every URI. URLs occur most commonly to reference web pages (http), but are also used for file transfer (ftp), email (mailto), database access (JDBC), and many other applications.
<b>Hostname</b>	Individual Internet host computers use domain names as host identifiers, also called host names. The term host name is also used for the leaf labels in the domain name system, usually without further subordinate domain name space. Host names appear as a component in URLs for Internet resources such as web sites.
<b>Most common types of records</b>	<b>SOA</b> (Start Of Authority), IP address (A or AAAA), <b>MX</b> - SMTP mail exchangers, <b>NS</b> - name servers, <b>PTR</b> - pointer for reverse DNS lookups, <b>CNAME</b> - domain Name aliases.
<b>zone file</b>	The DNS database is traditionally stored in structured zone file. Zone file is a text file that describes a DNS zone. A DNS zone is a subset, often a single domain, of the hierarchical domain name structure of the DNS. The zone file contains mappings between domain names and IP addresses and other resources, organized in the form of text representations of resource records (RR).
<b>DNS zone</b>	Any distinct, contiguous portion of the domain name space in the DNS for which administrative responsibility has been delegated to a single manager.

→ To view the DNS cache report

1. Open the **Diagnostics** tab in the main screen and scroll down to **System Status**.
2. Select **DNS Cache** in the list as shown here.



1. The **DNS Cache Records** report will be displayed.

TTL	Type	Name	Address	Flags	Parameters # NAPTR				Parameters # SRV				Parameters # CNAME				
					Order	Preference	RegExp	Replacement	Service	Port	Priority	Weight	Target	Negative	Change		
52	A	green.sbc.dial	10.38.125.233														
4	A	white.sbc.dial	10.38.125.233														
4	A	white.sbc.dial	10.38.125.233														
4	A	white.sbc.dial	10.38.125.233														
52	NAPTR	green.sbc.dial		0	1	1			_sip._udp.gre.SIP-DQU								
52	NAPTR	green.sbc.dial		0	1	1			_sip._udp.gre.SIP-DQU								
4	NAPTR	blue.sbc.dial		10	10				_sip._udp.sbc.SIP-DQU								
4	NAPTR	white.sbc.dial		0	10	10			_sip._udp.wht.SIP-DQU								

### 3.6.3.3 BorderNet Scale Up/Scale Down Options

The BorderNet SBC can be scaled up and down (vertical scaling) according to system requirements. Both actions require that the server be shut down and restarted.

- Scale Up refers to the addition of resources to the existing system such as CPU, memory, storage.
- Scale Down refers to a downgrade to a less powerful machine type.



Figure 5: Scale Up

**Scale Up** and **Scale Down** work only on AWS and require a failover procedure initiated by the active platform. This will terminate calls which have not yet been answered and also active calls using TCP or TCP-based signaling protocol, including TCP, TLS and WebSocket.

Once the new instance with the new machine type has become the Active platform, the new instance process is conducted again on the current Standby system, in order for both the Active and Standby platforms to have the same instance type.

The following process is necessary:

- Shut down the Standby system (the system with the previous machine type).
- Change the instance type to the new machine type chosen.
- Load the new instance with the BorderNet configuration.

After synchronization the process is complete.

#### NOTES:

- If the measured value of CPU utilization is **equal or above** the value set in the **CPU Utilization Threshold (%)** parameter, AND if it is kept at that level for a duration equal or more than the time set in the **Threshold Contiguous Time (sec)** parameter, then a switch machine type procedure shall be triggered, with a machine type value taken from the **Scale-Up** configuration.
- If the measured value of CPU utilization is **equal or below** the value set in the **CPU Utilization Threshold (%)** parameter, AND if it is kept at that level for a duration equal or more than the time set in the **Threshold Contiguous Time (sec)** parameter, then a switch machine type procedure shall be triggered, with a machine type value taken from the Scale-Down configuration.
- After having one machine type already migrated to the new instance type, the process of migrating the second platform should only be conducted once.
- After both platforms have migrated to the new instance type there is no extra failover required.



The **Scale Up** and **Scale Down** actions are directly controlled from the GUI through the **Scalability Profile** window accessible from the menu under the **System** tab.

**Scale UP/DOWN configuration**

Enable:

**Scale Up Parameters**

Cpu Utilization Threshold (%)

Concurrent Session Threshold

Threshold Configuration Time (Sec)

Machine Type

**Scale Down Parameters**

Cpu Utilization Threshold (%)

Concurrent Session Threshold

Threshold Configuration Time (Sec)

Machine Type

Save Cancel

### 3.6.3.4 BorderNet Scale Out/Scale In Options

The BorderNet SBC can be scaled out and in (horizontal scaling) according to system requirements. Both actions require that the server be shut down and restarted.

- Scale Out refers to the addition of servers to the existing server or multiple servers. It requires support of a distributed architecture, where the workload is balanced between the different servers. Scalability can be architected into the system, so it is not automatic and is generally more challenging than Scaling Up.
- Scale In refers to the process in which a set of servers are removed (brought down), leaving a lower number of servers (or even a single one) in an operational state



Figure 6: Scale-Out

**Scale Out** and **Scale In** work only on AWS and require a failover procedure initiated by the active platform. This will terminate calls which have not yet been answered and also active calls using TCP or TCP-based signaling protocol, including TCP, TLS and WebSocket.

The following limitations refer to the scope for Scale Out/Scale In on the BorderNet SBC:

- Only Amazon (AWS) is supported.
- Only a concurrent sessions indicator is used as a threshold parameter for scaling decisions.
- Abnormal scenarios, such as a new instance which is not able to become active or is not responsive, are not handled in the current phase.
- Changing configuration at runtime is not part of the current phase. This will be implemented after the full integration of EMS.
- New instances are not yet configured. A change of configuration will be done only in a full scale-in state where only the redirect BorderNet is up.
- In the current phase, only the first redirect can be deployed in a High Availability configuration. All new instances will be deployed as standalones.

The **Scale Out** and **Scale In** actions are directly controlled from the GUI through the **Edit Scalability Profile** window.

### Scale IN / OUT configuration

Enable:

---

Scale AMI name:

Machine Type:

---

Scale IN Concurrent Session Threshold:

Scale IN Threshold Configuration Time (Sec):

Scale OUT Concurrent Session Threshold:


Scale OUT Threshold Configuration Time (Sec):




---

### 3.6.4 System Services

The System Administrator can start, stop, and restart configured services on the BorderNet SBC.

→ To view configured services:

1. From the **System** drop-down menu, select **System Services**, shown below.
2. Select the  icon to start, stop, or restart a service.

Service	Interfaces	Control Ports	Status
 SNMP	Management	161	Stopped
 Transcoding			Stopped
 RemoteTracing	Management	2010	Started
 Ping			Started
 Ftp	Utility	21	Stopped
 Telnet	Utility	23	Stopped
 Ibcf			Started
 SecureShell	Management and Utility	22	Started

The following table describes the System Services available on the BorderNet SBC.

Service	Associated Properties
SNMP	Allows connection establishment on a SNMP well-known port, supporting only GET commands.
Transcoding	Software transcoding is applicable in this version.
Remote Tracing	When started, tracing allows the external entities to send trace requests to the BorderNet SBC using Wireshark clients. The service runs on control port 2010. By default, the tracing service is allowed only on the System Management IP Address. This can be modified to allow tracing on the Utility IP address or both the System Management IP address and Utility IP address. The ACLs are dynamically created based on the tracing session requests and are not required to be explicitly added. Note: See the BorderNet SBC Maintenance and Troubleshooting Guide for instructions on how to use the Wireshark service.
Ping	The ping service can be started and stopped on a system-wide basis. No ACLs are required.

Service	Associated Properties
FTP	The FTP service controls port 21. By default, the value is OFF, and the interface type is Utility IP address. ACLs are required for remote IPs to access the service. See the <a href="#">Access Control Lists</a> section for additional information.
Telnet	The Telnet service controls port 23. By default, the Telnet service is turned OFF. When started, the Telnet service allows external remote entities to send Telnet requests to the system. The interface type is Utility IP address, but this can be modified to the System Management IP address or both the System Management IP address and Utility IP address. ACLs are required for remote IPs to access the service. See the <a href="#">Access Control Lists</a> section for additional information.
IBCF	The IBCF service affects both SIP and H.323 sessions. <b>WARNING:</b> Stopping and starting IBCF is service-affecting.
Secure Shell	The Secure Shell service controls port 22. By default, the value is OFF, and the interface type is Utility IP address. ACLs are required for remote IPs to access the service. See the <a href="#">Access Control Lists</a> section for additional information.

Table 3: System Services Available on BorderNet SBC

**Note:**

It is recommended to have either the Secure Shell or Telnet running at all times.

## 3.6.5 System Information

The System Administrator reviews the BorderNet SBC's properties and host information.

Select **System** → **System Information** to access property and host information, as shown below.

**System Information**

**System Properties**

- System Name: BorderNet4000
- Location: System Location
- License Reference: [View License](#)
- Platform Redundancy: Yes
- Management IP: 192.168.201.70
- Control Switch: [View Configuration](#)

**Primary Host**


- Status: Active
- BN4000 Host Name : sbc10g69
- BN4000 Id : 1001
- HA Link IP: 192.168.200.100
- HA Link Peer IP :
- Utility IP Address: 192.168.201.69
- Subnet Mask: 24

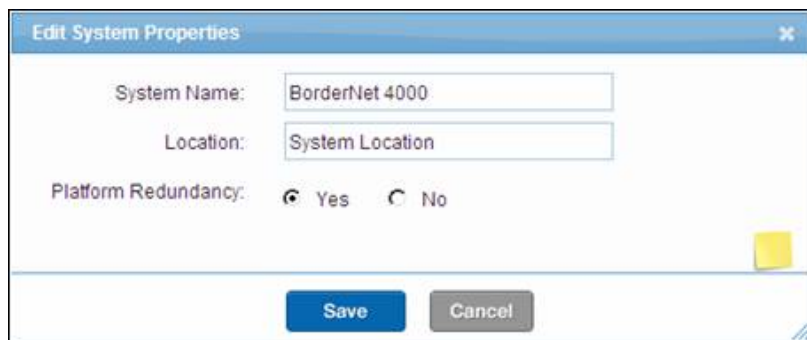
**Secondary Host**

- Status:
- BN4000 Host Name :
- BN4000 Id :
- HA Link IP:
- HA Link Peer IP :
- Utility IP Address:
- Subnet Mask:

### 3.6.5.1 Editing System Properties

→ To edit System Properties:

1. Select the  icon.  
The System Administrator can change the System **Name** and **Location**.
2. Click **Save** to apply the changes.



### 3.6.5.2 Managing Host Platforms



The Primary and Secondary host platforms are managed from the Web UI.

The **Status** field shows the active and standby platforms, shown below.




In the event of a switch-over, the **Status** is automatically updated.

The host platforms use the following icons:

- Reset 
- Platform Switchover  (active host platform only)

### 3.6.5.3 Resetting the Host Platform

Click the **Reset** icon to reset the host platform. This can also be used as a way to initiate a switch-over when the reset is done on the active platform.

---

**WARNING:**

The system will reboot when the **Reset** icon is selected. This is traffic-affecting on Standalone systems because there is no secondary BorderNet SBC system to take over traffic during the reboot.

---

### 3.6.5.4 Performing a Manual Switchover on the Host Platforms

Select the **Platform Switchover** icon from the active platform to initiate a manual switchover of the platform roles (the standby platform assumes the "active" role and active platform assumes the "standby" role). The active and standby status is automatically updated in the Web UI.

---

**WARNING:**

This action does not result in a platform reboot, but connectivity will be lost during the platform switchover. The operator will be required to login to a new session.

---

### 3.6.5.5 LDAP Configuration

**LDAP (Lightweight Directory Access Protocol)** is a software protocol for enabling anyone to locate organizations, individuals, and other resources such as files and devices in a network, whether on the public Internet or on a corporate intranet.

On TCP/IP networks, the **Domain Name System (DNS)** is the directory system used to relate the domain name to a specific network address (a unique location on the network). LDAP allows you to search for an individual on the network without knowing where they're located.

An LDAP directory can be distributed among many servers. Each server can have a replicated version of the total directory that is synchronized periodically. An LDAP server is called a **Directory System Agent (DSA)**. An LDAP server that receives a request from a user takes responsibility for the request, passing it to other DSAs as necessary, but ensuring a single coordinated response for the user.

LDAP uses a relatively simple, string-based query to extract information from MS Active Directory. A regular end user will never have to manually perform an LDAP query, because Outlook is LDAP-enabled and knows how to perform all the necessary queries on its own.

BorderNet supports a TLS/LDAPS secure connection and the default port for the secure LDAP is 636. Certificates received from the LDAP server are automatically accepted by BorderNet and EMS. No customized role attribute is required as role definition is performed based on role groups and the user's association to a role group.

In BorderNet the group names match the pre-defined roles available on the BorderNet. In order to support different privileges options for different BorderNets, the customer can define groups with the BorderNet pre-defined roles prefixed with a string. For example: IL\_SYSTEM\_ADMIN and US\_SYSTEM\_ADMIN. The BorderNet has an optional prefix parameter (example: prefix=IL, prefix=US)

In the EMS new roles are created with new names, so the customer can either create a new group or use an existing one. There is no need for a group prefix. The EMS roles can be customized and there is a single EMS on a network.

The Authentication process works as follows:

- Search the user.
- If a 'member of' attribute is available, then this attribute lists all the groups this user belongs to. (No need to search for groups, they are already listed).

- If a 'member of' attribute does not exist, then search all the groups to find the ones containing this user.
- Use the list of groups as a list of roles.
- The group list can contain also other group names in the tree, so it will ignore any unknown role name.
- Order of authentication - local users, LDAP, RADIUS.
- The required parameters are shown in the table below.

Parameter Name	Description	Mandatory	Optional Values	Default Value
<b>Enable</b>	Enable/disable LDAP configuration. Type: checkbox.		Checked/unchecked	Disable (not checked)
Connection				
<b>LDAP Server IP</b>	IP address of the LDAP server	Yes	IPv4 address	None
<b>LDAP Server Port</b>	TCP port number of the LDAP server	Yes	0-65535	389
<b>Use TLS</b>	Enable secure connection using LDAP over TLS (usually over port 636) Type: checkbox.		Checked/unchecked	Disable (not checked)
<b>Admin DN</b>	A user with privilege to access the LDAP server directory. A full path is required.	No	String. Example: CN=Administrator,CN=Users DC=dialogic,DC=com	None
<b>Admin password</b>	Password of Admin user. Should be hidden (the user should see '*' signs and not the real password)	No	String	None
Users				
<b>Users base DN</b>	Search scope to look for users (search starting with this point/ under this branch)	Yes	String CN=Users,DC=dialogic,DC=com	None
<b>User identification attribute</b>	Attribute type to uniquely identify a user. This is the attribute that will be used as the login identifier. Usually 'uid'. For AD it will be sAMAccountName	Yes	String	uid
Groups				
<b>Group membership attribute</b>	Attribute of a user entry listing all the groups this user is associated with.	No	String	memberOf
<b>Groups base DN</b>	Search scope to look for groups containing the user (search starting with this point/ under this branch)	Yes	String Example: CN=Guests,DC=dialogic,DC=com	None
<b>Group identification attribute</b>	Attribute type to uniquely identify a group (a group search filter). This is the attribute that will be used as the group name which is mapped to an access level role.	Yes	String Example: CN	None
<b>Only on BorderNet: Group name prefix</b>	String placed before the 'group identification' and removed by the BorderNet. Used for flexible provisioning of several groups with several prefixes on the LDAP server.	No	String	None

Table 4: Parameters for LDAP Configuration

### 3.6.5.6 RADIUS Configuration

**Remote Authentication Dial-In User Service (RADIUS)**, was originally designed to deliver AAA services for dial-up internet. As such, most of its parameters are network access oriented and are aimed to supply different networking properties for the user accessing the network services. Typical parameters include service type, protocol type, IP address to assign the user (static or dynamic), access list to apply, or a static route to install in the NAS routing table.

A **Network Access Server (NAS)** operates as a client of RADIUS. The client is responsible for passing user information to designated RADIUS servers, and then acting on the response which is returned.

The RADIUS server response includes a list of attribute-value pairs that describe the parameters to be used for a session.

As part of its authentication capabilities, the RADIUS protocol is widely used for user authentication which is not necessarily related to network access. On top of the regular PAP/CHAP password authentication, it can also support a variety of other user authentication protocols like EAP-TTLS, EAP-TLS and PEAP.

Transactions between the client and RADIUS server are authenticated through the use of a shared secret, which is never sent over the network. In addition, any user passwords are sent encrypted between the client and the RADIUS server, to eliminate the possibility that someone snooping on an unsecure network could determine a user's password.

RADIUS uses UDP as the transport layer, and therefore it implements reliability options on the application (RADIUS) level. If no response is returned within a predetermined length of time, the request is re-sent a number of times. The client can also forward requests to an alternate server or servers in the event that the primary server is down or unreachable.

RADIUS message types include the following:

- **Access-Request** - This is the first message sent from the client to the server, asking permission to access the network. It contains user and network information for authentication and authorization. An Access-Request can include multiple attributes, each containing some information regarding the requested service.
- **Access-Accept** - Sent from the server to the client, granting permission to access the network. An Access-Accept message can provide specific configuration information for the client, such as IP address, QoS profile, user authorization or any other attribute needed.
- **Access-Reject** - Sent from the server to the client, denying permission to access the network. Can include reject cause and a message to the user.
- **Access-Challenge** - Sent by the server to issue a challenge to which the user must respond. The client then re-submits its original Access-Request with the extra information required by the Access-Challenge.

→ To perform **RADIUS Configuration**:

1. From the **System** drop-down menu, select **RADIUS Configuration**.  
The **RADIUS Manager Configuration** screen opens.

**RADIUS manager configuration**

Enable:

Primary Server IP:

Secondary Server IP:

Server Port:

Retry Attempts:

Retry Interval:

Shared Secret:

Authentication Method:

Attribute type to contain Role string:

Save Cancel

2. Edit the parameters according to the options detailed below.
  - **Enable.** Enable/Disable RADIUS functionality. Uses IPv4 only.
  - **Primary Server IP.** Main IP address of RADIUS server.
  - **Secondary Server IP** Secondary RADIUS server, if Primary Server not responding.
  - **Server Port.** Destination UDP port of requests sent to the RADIUS server. Default value 1812.
  - **Retry Attempts.** No of attempts before switching to Secondary Server. Possible values 1-10. Default value = 3.
  - **Retry Interval.** Time in seconds between each retry attempt. Values 1-90. Default value = 3.
  - **Shared Secret.** Password shared between the BorderNet and the RADIUS server. String.
  - **Authentication Method.** Type of authentication protocol used to deliver username and password. PAP/CHAP. Default = PAP.
  - **Attribute Type to Contain Role String.** Attribute type in the **Access-Accept** message, to contain user role. The type parameter in the RADIUS specification is one octet, so it can have values of 1-255. Default is the **Class** attribute (type=25).
3. Click **Save**.

## 3.6.6 Licenses

The BorderNet SBC provides reliable licensing management.

There are three modes of licensing:

- Regular standalone licensing using a local license file on the BN
- Licensing of a single BN through a Nalpeiron server which is used for license retrieval and then the BN builds a local file. There is no on-going license enforcement through Nalpeiron. License refresh is triggered manually.
- EMS-based network licensing. The initial license is retrieved from the Nalpeiron server using a DLGC interface and then the EMS builds a local file. There is no on-going license enforcement through Nalpeiron and only periodic usage updates are sent for statistical purposes. License refresh is triggered manually.

### 3.6.6.1 BorderNet SBC Licenses

The System Administrator manages the licenses on the BorderNet SBC.

→ To access the **License Configuration** screen:

1. From the **System** drop-down menu, select **License**.
2. Select the **Apply New License** button in the upper right corner of the screen to upload additional license files.



License Configuration		Apply New License
License Features		
Feature Name	Trial Capacity	
SESSIONS	64000	
PROTOCOL 1	SIP	
PROTOCOL 2	H323	
TRACING	Enabled	
HIGH AVAILABILITY	Disabled	
SESSION DETAIL RECORD	Enabled	
IPV6	Enabled	
IPSEC TUNNELS	25000	
TRANSCODING GATEWAY SESSIONS	1000	
WIDEBAND AUDIO CODECS	Enabled	
SIP WITH ISUP	Enabled	
HOST 1	ANY	
EXPIRATION DATE	2016-09-30	

The **License Configuration** screen shows the features and capacity licensed on the BorderNet SBC system. Refer to the *BorderNet Maintenance Guide* for alarms related to system licenses (such as license expired).

### 3.6.6.2 Network Wide Licenses

Network -wide licensing provides a centralized location for dynamically sharing the pool of licenses among a network of BorderNet SBCs. This licensing solution is agnostic to both the deployment mode (hardware, virtualized, cloud) and the operating system (any Linux flavor supported by the BorderNet SBC).

The solution is based on two logical components:

- Licensing server - entity controlling the licenses for the network of BorderNet SBC elements.
- Licensing client - a Daemon process installed on every BorderNet SBC in the network.

Dialogic supports both regular and cloud-based operations so a network licensing solution helps to accommodate licensing flexibility for public, private and hybrid cloud-based environments. The licensing mode is based on SaaS for a highly available cloud-based environment with 99.9% uptime.

Network-wide licensing allows also the management of application features by dynamically sharing a bank of capabilities among the clients. For example, a customer may have 10 BorderNet SBCs and 10,000 SIP sessions. The 10,000 transcoding sessions will then be dynamically shared among the 10 BorderNet SBCs on an as-needed basis.

In the case of EMS-based NWL, the EMS creates a non-reproducible license and uses it locally. It then sends periodic usage reports as accumulated values only and not per BorderNet. This is illustrated in the figure below.

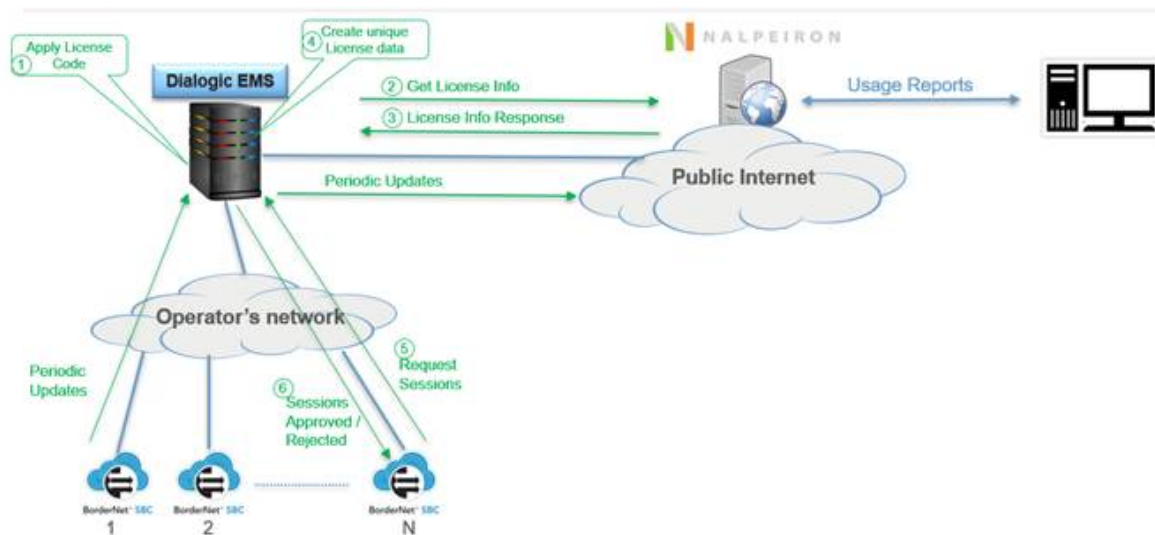


Figure 7: EMS-Based Network Wide Licensing

EMS to BorderNet communication is based on the current **RESTfull API** mechanism and all licensing messages are encrypted and authorized.

Message types include the following:

- EMSGetLicense - BorderNet to EMS: request for a new initial license, or a request for updating the existing license, if the license has been changed. If this request fails it is reattempted every 60 seconds.
- EMSGetLicenseResponse - EMS to BorderNet: list of full licenses and features.
- EMSUpdate - BorderNet to EMS: periodic updates on the amount of sessions and features used and requested. Used to both request and inform on current sessions usage.
- EMSUpdateResponse - EMS to BorderNet: The amount of sessions approved per each feature.

Network Wide Licensing configuration can be implemented directly from the BorderNet screen:

### Network Wide Licensing Configuration

**Enable:**  Nalpeiron server  EMS server

**IP Type:**

**DICLA IP Address:**

**DICLA Port:**

**DICLA Client Code:**

In all circumstances the BorderNet SBCs will attempt to receive approval from the license server for their operations and the full license capacity, or the latest approved capacity will be allowed, which is also based on timeout considerations. See the table below as an example.

	BorderNet SBC	NWL	BorderNet SBC Action
Starting to work	Sends request for 120 sessions	Approved	Will use up to the full license, 1,000 sessions 120 sessions will be kept in memory
5 minutes later	Sends request for 140 sessions	Approved	Will use up to the full license, 1,000 sessions 140 sessions will be kept in memory, replacing the 120 sessions
5 minutes later	Sends request for 150 sessions	Denied	Will use up to the last approved request – 140 sessions

As shown in the diagram below, the licensing clients, on first access to the licensing server, acquire a code-based framework of features available. Subsequent access will guarantee the appropriate permissions for the quantitative features in the code-based framework.

The license code is basically a numerical string configured onto the BorderNet SBC.

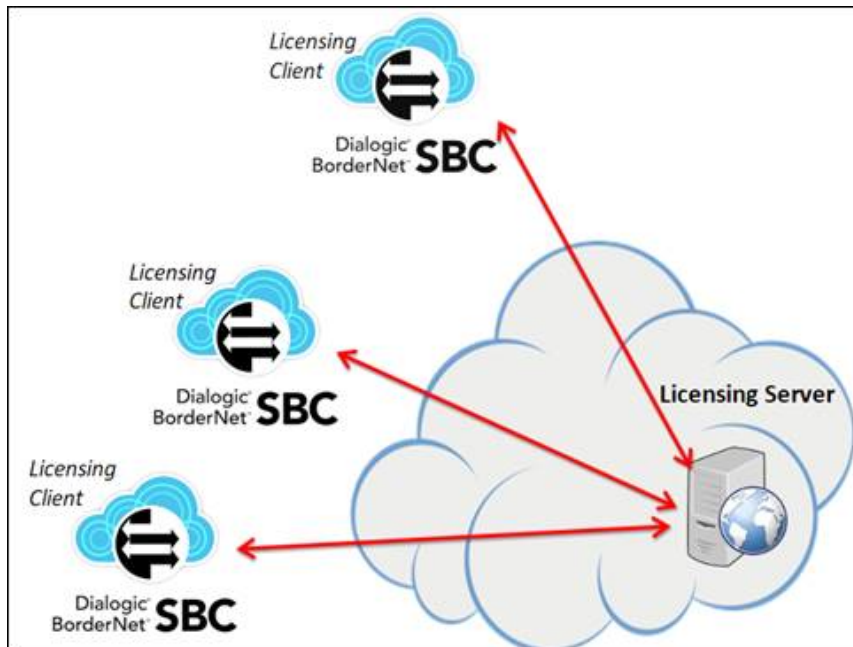


Figure 8: BorderNet SBC's Network Wide Licensing Architecture

### 3.6.7 SNMP Trap Managers

The BorderNet SBC uses **Simple Network Management Protocol (SNMP)** for sending alarm traps to external SNMP managers, and also for remote SNMP managers to retrieve limited information from the BorderNet via GET requests.

It supports SNMPv3, which enables each SNMP packet to be both authenticated and encrypted in a secure way.

SNMPv3 requires an application to know the identifier (snmpEngineID) of the remote SNMP protocol engine in order to retrieve or manipulate objects maintained on the remote SNMP entity. The EngineID is also one of the inputs used for key derivation of the authentication and privacy keys.

In order to learn the snmpEngineID of a remote SNMP protocol engine, a discovery mechanism is used.

For SNMPv3 traps there is no discovery process. Traps are also not acknowledged.

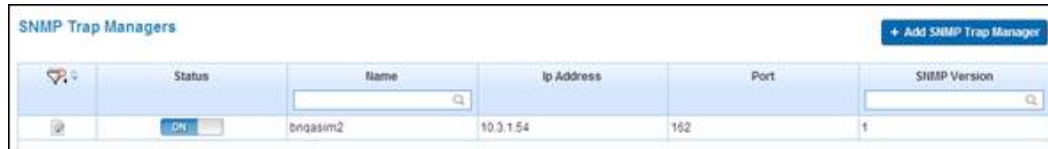
The authoritative SNMP engine for a trap packet is the sending SNMP agent. Since the generator of the message and the authoritative engine are one and the same, there is no need for the SNMPv3 discovery process. All the information is already inside the single trap message.

As mentioned, SNMPv3 traps use the engineID of the local application sending the trap rather than the engineID of the remote application (like in a GET request). This means that you have to create users in your remote user database (the SNMP trap server) for every engineID you wish to send traps from. Some servers allow all EngineIDs and identify the traps by their user-name.

SNMP Trap Managers are configured to manage sending alarms in real time to North-bound trap managers.

→ To configure an SNMP Trap Manager:

1. From the **System** menu, select **SNMP Trap Managers** to access the **SNMP Trap Manager** screen.



2. Select the +Add SNMP Trap Manager button in the upper right corner of the screen to configure a new SNMP Trap Manager. Explicit IP routes must be added for the **SNMP Trap Manager**.

3. Click **Save**.

SNMP is also used on BorderNet to provide session information as a response to GET requests, supporting the below two OIDs. The **SNMP Access List** must be configured.

The traps and GET requests do not have common configuration. The traps will be sent to the servers configured as Trap Managers, and the GET request is allowed for every network element which is enabled in the ACL. Both the GET and traps use a fixed non-configurable 'public' community.

#### SNMPv3 Configuration Parameters

- **SNMPv3 Mode.** Type of security to be deployed. There is no privacy without authentication.
- Possible values:
  - oNo authentication, no privacy
  - oAuthentication, no privacy
  - oAuthentication, privacy
- **User Name.** Mandatory string 1-32 characters.
- **Authentication Protocol.** Authentication algorithm.
- Possible values:
  - oNone
  - o HMAC-MD5
  - oMHAC-SHA
- **Privacy Protocol.** Encryption algorithm. **Privacy Protocol.** Encryption algorithm.

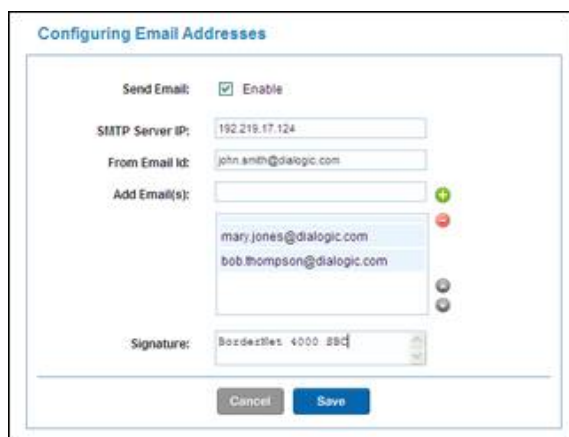
- Possible values:
  - oNone
  - o DES
  - o AES-128
  - o AES-192
  - oAES-256
- **Authentication Key.** A phrase used as the secret for the authentication algorithm. Mandatory if 'Authentication Protocol' parameter is not set to **None**.
- **Privacy Key.** A phrase used as the secret for the encryption algorithm. Mandatory if 'Privacy protocol' parameter is not set to **None**.

## 3.6.8 Email Configuration

Email is enabled by defining the SMTP server for emails.

→ To perform email configuration:

1. From the **System** drop-down menu, select **Email Configuration** to configure the email address, as shown below.



To configure email, the server must be in the user's network, and explicit IP routes must be added for the SMTP server. If the server is not in the user's network, a gateway must also be established.

2. Ensure that there is an IP route added to the BorderNet SBC, and that a path is established to allow SMTP traffic.

---

**Note:**

Management messages are not sent over session interfaces. An IP route must be configured in order for the management message to go through the system.

---

## 3.6.9 Audit Logs

For details on Audit Logs see the *BorderNet SBC Maintenance Guide* document.

## 3.6.10 Configuring SDR

This window allows the System Administrator to manage the **SDR (Session Detailed Records)** in the BorderNet SBC. When SDR is enabled, the BorderNet SBC automatically creates an ACL to allow SDR traffic. For details on the BorderNet SBC SDR, see [Session Detailed Records](#) section.

→ To configure the SDR functionality:

1. Select **System** → **Administration** → **SDR Configuration**.

The **SDR Configuration** window opens.

2. Edit the appropriate fields.

### 3.6.10.1 Record Configuration

The top portion of the **SDR Configuration** screen enables the System Administrator to establish how the records will be sent to the SDR destination:

- **Enable.** Indicates whether the BorderNet SBC generates Session Detail Records:
  - A check mark means SDRs will be recorded.
  - No check mark means SDRs will not be recorded
- **Transport Method.** Specifies how SDR files are sent to the SDR destination:
  - FTP (default)
  - SCP
- **Destination IP addresses.** Specifies where to send the SCR files.
  - The primary address indicates the external SDR destination to receive the files.
  - The secondary address is used if the primary address cannot be reached.
- **Destination Directory.** Specifies in what directory to place the SCR files.
  - The primary destination directory indicates the external SDR destination directory path.
  - The secondary destination directory is used if the primary directory cannot be reached.
- **User Name and Password.** Indicates the user name and password required to enable the BorderNet SBC to access the SDR destination. BorderNet SBC uses this information to transport the files.

### 3.6.10.2 SDR Billing Parameters

SDR billing parameters allow the System Administrator to manage the way SDR records are received:

- **File Creation Interval (seconds).** Determines how often SDR records are written to a file. The default value is 10 seconds. Short intervals enable the billing system to see the records faster.

- File maximum size (MB). Maximum SDR file size in Mega Bytes.
- **Compression.** Indicates whether the records are compressed (YES) or uncompressed (NO).
  - Compressed files save disk space and are sent more quickly through the transport system.
  - Uncompressed files save time reading the files once they are received at the SDR destination.
  - The default value is **NO**.

---

**Note:**

Compressed files are sent to the SDR destination as a .gz file. Uncompressed files are sent to the SDR destination as a .csv file.

---

- **Number of hours to keep files in disk.** Indicates how long BorderNet SBC will retain the SDR files after sending them to the SDR destination. The default value is 48 hours.
- **Field delimiter.** Sets the special character used to separate each field in the SDR file. The field delimiter can be a comma (,) or a semi-colon (;). The default value is a comma.
- **Create SDR format.** Selects the format of the SDR. Possible values: ControlSwitch iCDR and BN formats.
- **Write header in each SDR file.** If set to Yes, writes the SDR header in each file.
- **Time Zone.** Select the time zone, using the drop-down menu.
- **Number of decimal digits.**

### 3.6.10.3 Customizing SDR

SDRs can be customized by selecting or deselecting specific parameters.

→ To customize an SDR:

1. Select **System** → **SDR Customization** to display five categories in the SDR parameter list, as shown below.



2. Click the triangle icon next to the parameter category to view the specific parameters, as shown below.



Parameters that are included in the SDR have a blue checkmark.

3. To deselect a parameter, click the checkmark (a blue box replaces the checkmark to show that the parameter is deselected).
4. Click **Save**.

# 4. SIP Configuration

This section describes how to configure SIP in BorderNet SBC.

**Note:**

Only an Application Administrator can configure the SIP application.

## 4.1 Interface

A SIP interface is the local BorderNet SBC IP address, through which the BorderNet SBC sends and receives messages. The first step in configuring SIP is to create a SIP interface.

The **SIP Interface** window (below) lists the SIP interfaces defined on BorderNet SBC.

→ To create a SIP Interface:

1. Select **Application** → **SIP Configuration** → **Interface**.

The **SIP Interface** window opens.

Status	Name	Domain	Network Type	SIPconnect	Trust Level	Allow Assoc Peers Only	Signaling IP	VLAN Name	Signal Port	Signaling Protocol	Signaling TOS	TGRP Context	Parameter Profile	Media Profile	Service Profile	Security Profile
ON	Spectra_Ac	SpectraLocal	Access-Loc	No	High	<input type="checkbox"/>	10.20.50.35	spectra	5080	UDP	0		Access-local	Access-local	Access-local	Access-local
ON	Spectra_Ac	spectraaccess	Access-Pul	No	High	<input type="checkbox"/>	10.20.50.35	spectra	5070	UDP-TCP	0		default_access	default_access	default_access	default_access
ON	Spectra		Interconnect	No	High	<input checked="" type="checkbox"/>	10.20.50.35	spectra	5060	UDP-TCP	255		Default	Default	Default	Default
ON	ovip-702-V4-A	local702	Access-Loc	No	High	<input type="checkbox"/>	10.70.10.3	OvIP-702	3040	UDP	0		Access-local	Access-local	Access-local	Access-local
ON	ovip-702-v4-A	public702	Access-Pul	No	High	<input type="checkbox"/>	10.70.10.2	OvIP-702	3030	UDP	0		default_access	default_access	default_access	default_access
ON	ovip-701-V4-A	local701	Access-Loc	No	High	<input type="checkbox"/>	10.70.10.3	OvIP-701	3040	UDP	0		Access-local	Access-local	Access-local	Access-local
Off	ovip-701-v4-A	public701	Access-Pul	No	High	<input type="checkbox"/>	10.70.10.2	OvIP-701	3030	UDP	0		default_access	default_access	default_access	default_access

2. Click on the **+Add New SIP Interface** button, and enter the below parameters:

**Add New SIP Interface**

Status:  ON  OFF  
 Name:   
 Domain:   
 Network Type:  Select  
 Diameter Charging Type:  None  
 Credit Control Failure Handling:  Continue  
 Network Property:  IMS  
 Trust Level:  High  
 Associated Peers:   
 Transport IP Address Type:  IPv4  IPv6  
 Signaling IP:  Select  
 Signaling Port:  5060  
 Transport Protocol:  UDP-TCP  
 Max allowed UDP MTU:  8  
 Time Zone:   
 Signaling TOS:   
 TGRP Context:   
 Enforce IPsec:  No  Yes  
 Parameter Profile:  Select  
 Media Profile:  Select  
 Service Profile:  Select  
 Security Profile:  Select



## Main Tab

- **Status.** Enable/disable the SIP interface, by selecting **ON** or **OFF**.
- **Name.** The unique **Name** of the SIP interface, identifying the BorderNet SBC.
- **Domain.** The BorderNet SBC domain name.
- **Network Type.** The **network type** selected using the drop-down menu.
- Possible values:
  - **Interconnect** indicates a public network.
  - **Local** indicates a private network.
  - **Access-Public** indicates a public network towards UEs.
  - **Access-Local** indicates home access network.
  - **Access-Interconnect** indicates visiting access network
  - **SipRec** indicates SIP recording. This interface will be used to connect to a SIP-REC SRS.
  - For Access-Local and Access-Interconnect network types, see [Registration](#).
  - For Interconnect and Access-Public, see also [SIP Connect](#).
- **Network Property.** Selecting the **IMS** check-box enables the sending and the receiving of SIP IMS headers. If this check-box is selected, the following field is displayed: **Network Property**. Selecting the **IMS** check-box enables the sending and the receiving of SIP IMS headers. If this check-box is selected, the following field is displayed:
- **Local Operator ID.** The local operator identifier - a string that identifies the originator of the P-Charging-Vector header. From this value the **BoderNet SBC** derives the **orig-voi** parameter (appears in the P-Charging-Vector header). **Local Operator ID.** The local operator identifier - a string that identifies the originator of the P-Charging-Vector header. From this value the **BoderNet SBC** derives the **orig-voi** parameter (appears in the P-Charging-Vector header).
- **Trust Level.** Select the **Trust Level** from the drop-down list. Possible values: **High**, **Medium**, or **Low**.
- **Associated Peers.** If checked, indicates that the SIP Interface allows only **Associated Peers**, all traffic from all peers is allowed.
- **Transport IP Address Type.** Select between **IPv4** and **IPv6**. **Transport IP Address Type.** Select between **IPv4** and **IPv6**.

**Note:**

IP addresses are filtered based on the selected Type.

- **Signaling IP.** Select the signaling IP address, and their associated VLAN information.
- **Signaling Port.** Enter the **signaling port**. **Signaling Port.** Enter the **signaling port**.
- **Transport Protocol.** Select the transport protocol (see [Transport Protocol](#))
- **Max Allowed UDP MTU.** For UDP connections, enter the maximum allowed MTU which will be used by a UDP connection. Any message beyond this limit will invoke a new TCP connection to be used as a replacement for that call (automatic UDP to TCP transition). A value of zero disables this feature (the default behavior). **Max Allowed UDP MTU.** For UDP connections, enter the maximum allowed MTU which will be used by a UDP connection. Any message beyond this limit will invoke a new TCP connection to be used as a replacement for that call (automatic UDP to TCP transition). A value of zero disables this feature (the default behavior).
- **Signaling TOS.** Enter the **TOS** value. By default, this value is 0. To modify this field enter a value into the field, or double-click in the TOS field to open the Edit TOS Bit Values screen.



- Select the **Differentiated Service Point Code (DSCP)** from the **DSCP Mode** drop-down menu.

1. Possible mode selections and values include:

---

**Note:**

The Class/DSCP Value is automatically populated with corresponding list of selections based on the DSCP Mode selected.

The TOS Value is automatically populated with the appropriate value based on selected Class/DSCP.

---

DSCP Mode	Corresponding Class/DSCP Values	TOS Value Range
Best Effort	Default PHB	0
Transparent	0xFF	255
AF: Assured Forwarding	AF11, AF12, AF13, AF21, AF22, AF23, AF31, AF32, AF33, AF41, AF42, AF43	40 - 152
EF: Expedited Forwarding	EF	184
CS: Class Selector	CS1, CS2, CS3, CS4, CS5, CS6, CS7	23 - 224
NS: Non Standard	1 - 63	4 - 252

1. Click **Save** to add the selected **TOS Value** to the **SIP Interface**.
  - o **TRGP Context.** The **Trunk Group** ID domain for RFC 4904 mapping. TRGP Context is not required for otg/dtg Trunk Group mapping.
  - o **Enforce IPsec.** Check the box to enforce IPsec on the interface.
  - o Profiles are mandatory for SIP Interfaces. If no custom profile is assigned to the interface, a default profile will be attached to it. Select profiles using the drop-down menus:
  - o **Parameter Profile.** Captures the SIP-specific parameters that need to be configured and influences session behaviors. This profile is for general purpose and core protocol-specific parameters.
  - o **Media Profile.** Captures all media-related parameters and configurations, including port allocations and codec configurations.
  - o **Service Profile.** Captures session-impacting services (these services are usually not part of core session behavior), such as routing, redirection, and transparency settings.
  - o **Security Profile.** Captures the relevant security properties to be exercised on the sessions. Security properties include control mechanisms such as rate-limiting on sessions and packets, maximum concurrent sessions, blacklisting, and so forth.
  - o **TLS and SRTP Profile** appear only when TLS transport protocol is selected (see [TLS Profile](#) and [SRTP Profile](#) sections).

#### Service Tab

- **Routing.** Advanced Policy and Advanced Policy for Re-Route. Select these from the drop-down list of available policies.
- **Sip-Rec.**
  - o **Recording Enable:** Yes/No
  - o **Recording Preference:** Support/Ignore
  - o **SRS Peer:** Select from the available drop-down list.
  - o **Release Call on SRS Failure:** Yes/No

The screenshot shows the 'Add Sip Interface' dialog box with the 'Service' tab selected. The 'Port Allocation' section is active, showing the following settings:

- Routing:**
  - Advanced Policy: Select
  - Advanced Policy for ReRoute: Select
- SipRec:**
  - Recording Enable:  Yes  No
  - Recording Preference:  Support  Ignore
  - SRS Peer: Select
  - Release call (CS) on SRS failure:  Yes  No

Buttons for 'Save' and 'Cancel' are located at the bottom of the dialog.

#### Port Allocation Tab

1. Select the port(s) to be allocated from the **Media Port Allocation** list
2. Use the **Add >** button to move them to the **Selected** list.
3. Click **Save** to add the **SIP Interface** to the system.

The screenshot shows the 'Add Sip Interface' dialog box with the 'Service' tab selected. The 'Port Allocation' section is active, showing the 'Media Port Allocation' view:

- Media Port Allocation:**
  - Available:** A list containing 'LOAD' and 'two\_ports'.
  - Selected:** An empty list.
  - Buttons: 'Add >' and '<< Remove'.

Buttons for 'Save' and 'Cancel' are located at the bottom of the dialog.

## 4.2 Peer

A peer model is a remote entity with which the BorderNet SBC exchanges SIP traffic. Peers are created and associated with SIP Interfaces to facilitate call routing.

→ To create a SIP peer:

1. Select **Application** → **SIP Configuration** → **Peer**.
2. The **SIP Peer** window opens.

Status	Name	Class ID	Network Type	Source List	Trust Level	Destination Address Type	Destination FQDN/IP	Destination Port	Protocol	TGRP ID	Parameter Profile	Media Profile	Service Profile	Security Profile
ON	Peer_30_20	EP	Interconnec	IPv4,10.10.30	High	IPv4	10.10.30.20	5060	UDP		Default	Media_Prof_3	Default	Default
ON	Spect2-40-28	spec2	Interconnec	IPv4,10.10.40	High	IPv4	10.10.40.28	5060	UDP		Default	Media_40_26	Default	Load_Prof
ON	Spect2-30-28	spec2	Interconnec	IPv4,10.10.30	High	IPv4	10.10.30.28	5060	UDP		Default	Media_Prof_3	Default	Load_Prof

3. Click the **+Add New SIP Peer** button to add a new SIP peer:

General Tab

- **Status.** Enable/disable the SIP peer, by selecting **ON** or **OFF**.
- **Name.** The unique **Name** of the SIP peer.
- **Class ID.** The Classification Identifier, a string that enables the logical grouping of peers.
- **Network Type.** The network type selected using the drop-down menu.
- Possible values:
  - **Interconnect** indicates a public network.
  - **Local** indicates a private network.

- **oAccess-Public** indicates a public network towards UEs.
  - **oAccess-Local** indicates home access network.
  - **oAccess-Interconnect** indicates visiting access network.
  - **oSipRec** indicates SIP recording. The connected peer will be used as a SIP-REC SRS.
  - **oFor Access-Local and Access-Interconnect network types**, see [Registration](#).
  - **oFor Interconnect and Access-Public**, see also [SIP Connect](#), and [Surrogate Registration](#).
- **Network Property.** Selecting the IMS check-box enables the sending and the receiving of SIP IMS headers. If this check-box is selected, the following field is displayed:
- **Local Operator ID.** The local operator identifier - a string that identifies the originator of the P-Charging-Vector header. From this value the BoderNet SBC derives the orig-ioi parameter (appears in the P-Charging-Vector header).
- **Selecting the OMR check-box** (applicable only when *Network type=Interconnect*), enables the user to provision: **Selecting the OMR check-box** (applicable only when *Network type=Interconnect*), enables the user to provision:
- **OMR IP Realm.** Used as the BorderNet SBC's realm, when handling the OMR, and populating the relevant OMR SDP attributes, and feature capability header. **OMR IP Realm.** Used as the BorderNet SBC's realm, when handling the OMR, and populating the relevant OMR SDP attributes, and feature capability header.
- **Trunk Authentication.** Mark this option to activate registration and authentication for SIP trunks. Only available for SIP interconnect 'Netwok Type'. See 'Trunk Authentication' section below for more configuration options.
- **Source List.** Select between **IPv4 and IPv6**, and enter a set of IP addresses from which SIP messages are received, into the Source List field. A peer can be:
  - IP address and port (for example, 10.13.4.108 and 5060, port is optional).
  - IP address, subnet mask and Port (for example, 10.13.4.108/24).
  - **Trust Level.** Select the **Trust Level** from the drop-down list. Possible values: **High, Medium, or Low.**
  - **Destination FQDN/IP.** Select IPv4, IPv6 or FQDN of the peer (the destination peer to which traffic will be sent).
  - Possible values:
    - oIP:Port
    - oFQDN.destination
- **Protocol.** Select the protocol, using drop-down menu: Possible values: UDP, TCP and TLS. Multiple protocols can be selected (see [Transport Protocol](#)).
- **Max Allowed UDP MTU.** For UDP connections, enter the maximum allowed MTU which will be used by a UDP connection. Any message beyond this limit will invoke a new TCP connection to be used as a replacement for that call (automatic UDP to TCP transition). A value of zero disables this feature (the default behavior). **Max Allowed UDP MTU.** For UDP connections, enter the maximum allowed MTU which will be used by a UDP connection. Any message beyond this limit will invoke a new TCP connection to be used as a replacement for that call (automatic UDP to TCP transition). A value of zero disables this feature (the default behavior).
- **Time Zone.** Double-click to select the desired time zone.
- **TGRP ID.** [Trunk Group](#) id (a string that contains up to 100 characters).
- **Enforce IPsec.** Check the box to enforce IPsec on the peer.
- Profiles are mandatory for SIP peers. If no custom profile is assigned to the peer, a default profile will be attached to it. Select profiles using the drop-down menus:
  - **Parameter Profile.** Captures the SIP-specific parameters that need to be configured and influences session behaviors. This profile is for general purpose and core protocol-specific parameters.
  - **Media Profile.** Captures all media-related parameters and configurations, including port allocations and codec configurations.
  - **Service Profile.** Captures session-impacting services (these services are usually not part of core session behavior), such as routing, redirection, and transparency settings.
  - **Security Profile.** Captures the relevant security properties to be exercised on the sessions. Security properties include control mechanisms such as rate-limiting on sessions and packets, maximum concurrent sessions, blacklisting, and so forth. **Security Profile.** Captures the relevant security properties to be exercised on the sessions. Security properties include control mechanisms such as rate-limiting on sessions and packets, maximum concurrent sessions, blacklisting, and so forth.

#### Service Tab

- **Routing.** Advanced Policy and Advanced Policy for Re-Route. Select these from the drop-down list of available policies.
- **Sip-Rec.**

- o **Recording Enable:** Yes/No
- o **Recording Preference:** Support/Ignore
- o **SRS Peer:** Select from the available drop-down list.
- o **Release Call on SRS Failure:** Yes/No

#### Port Allocation Tab

1. Select the port(s) to be allocated from the **Media Port Allocation** list
2. Use the **Add >** button to move them to the **Selected** list.

1. Click **Save** to add the **SIP Peer** to the system.

## 4.3 Interface-Peer

Associations should be created between SIP Interfaces and Peers to manage the traffic.

→ To create a SIP Interface-Peer association:

1. Select **Application** → **SIP Configuration** → **Interface-Peer**.

The SIP Interface-Peer window opens.

Status	Peer	Interface	Connectivity Timer	KAInterface	KATryCount
<input checked="" type="checkbox"/>	Peer_30_20	Home_Int_30_26	0	Home_Int_30_26	1
<input checked="" type="checkbox"/>	Spect2-40-28	Public_Int_40_26	0	Public_Int_40_26	1
<input checked="" type="checkbox"/>	Spect2-30-28	Home_Int_30_26	0	Home_Int_30_26	1

2. Click on the +Add New SIP Interface-Peer button to add a new association:

- **Status.** Enable/disable the SIP Interface-Peer association, by selecting **ON** or **OFF**.
  - **Peer.** Select the pre-defined peer, using the drop-down menu.
  - **Interface.** Select the pre-defined interface, using the drop-down menu.
  - **Connectivity Timer.** The interval between the SIP options in seconds, which enables connectivity check for the association, range [40 - 900], 0 disables the connectivity check (default = 0).
  - **KA Interface.** Select the interface which the connectivity check is applied on.
  - **KATryCount.** Number of retry attempts [1-10].
  - **KA Max-Forwards.**
  - **KA Successful Response Codes.**
3. Click **Save** to make the association.

## 4.4 Parameter Profile

The following table shows the default parameter profile's properties:

Property	Description	Default
Timer 1	Round-Trip Time (RTT) estimate.	500 msec
Timer 2	Maximum retransmit interval for non-INVITE requests and INVITE responses.	2000 msec
Timer C	Proxy INVITE transaction time-out	240 sec

Property	Description	Default
Max Retransmissions	Maximum number of retransmissions to attempt during the period between Timer 1 and Timer 2.	4
Supported Methods	INVITE-requests session set-up CANCEL-terminates an INVITE transaction ACK-acknowledges a completed INVITE transaction BYE-terminates a session OPTIONS-queries a remote end's status INFO-exchanges mid-call signaling information PRACK-sends a reliable provisional response UPDATE-sends an SDP offer to request a session update REFER-requests a call transfer request NOTIFY-provides event notification	All methods
Replaces header handling	Indicates how to handle the Replaces header. Possible values: Forward, Replace, and Reject.	Forward
Min SE	Minimum time that must elapse prior to Session Expiry	90 sec
Max SE	Maximum time that can elapse prior to Session Expiry	7200 sec
Session Timer Value	Periodically refreshes SIP sessions.	1800 sec
Require via INVITE Reliable Responses	The INVITE request contains a Require header with a 100rel tag to initiate a reliable response.	No
Initiate Reliable Responses on incoming INVITE	Returns a valid status code from the callback function to initiate an automatic response.	No
Force Fast Start	Initiates SIP upstream calls to a downstream endpoint.	No
Minimum Max-Forwards	Limits the number of times a request can be forwarded.	4

Table 5: Default SIP Parameter Profile

BorderNet SBC terminates **SIP REFER** messages (if selected), associated with unattended call transfer messages. Upon termination, the BorderNet SBC initiates a new call leg with the transfer target and later bridges the original call leg with the transferred leg to locally complete the call transfer. This feature must be initiated internally. Contact Dialogic Support for assistance.

SIP P-Header for 3GPP (RFC 3455) is supported - no operator configuration is required.

The default profile can be customized to create additional SIP parameter profiles. Once a customized SIP **Parameter Profile** is created, the new Parameter Profile is automatically added to the Parameter Profile drop-down menu.

The following example shows how to create a **SIP Parameter Profile** with the following requirements: 7 maximum retransmissions, all supported methods enabled, required reliable responses on incoming and outgoing INVITEs, and 20 maximum forwards.

→ To create a new SIP parameter profile:

1. Select **Application** → **SIP Configuration** → **Parameter Profile**.

The **SIP Parameter - Profile Configuration** window opens

Name	Network Type	T1 (msec)	T2 (msec)	Timer C (sec)	Max Retransmission	Allow Methods	Min SE	Max SE	Session Timer Value	Initiate INVITE Requiring 100rel	Initiate Reliable Provisional Responses	Minimum Max-Forwards
Reliable-Local	Local	500	4000	240	4	INVITE CANCEL ACK BYE	90	7200	1800	Yes	Yes	
Reliable-Interconnect	Interconnect	500	4000	240	4	INVITE CANCEL ACK BYE	90	7200	1800	Yes	Yes	
access-Interconnect	Access-Interconnect	500	4000	240	4	INVITE CANCEL ACK BYE	90	7200	1800	Yes	Yes	
access-Local	Access-Local	500	4000	240	4	INVITE CANCEL ACK BYE	90	7200	1800	Yes	Yes	
access-public	Access-Public	500	4000	240	4	INVITE CANCEL ACK BYE	90	7200	1800	Yes	Yes	
Default-Local	Local	500	4000	240	4	INVITE CANCEL ACK BYE	90	7200	1800	No	No	
Default	Interconnect	500	2000	240	4	INVITE CANCEL ACK BYE	90	7200	1800	No	No	



1. Click on the **+Add New SIP Parameter Profile** button, to create a new profile.

- **Name.** The new profile's name.
- **Network Type.** Select the **network type**, using the drop-down menu:
- **Interconnect** indicates a public network.
- **Local** indicates a private network.
- **Access-Public** indicates a public network towards UEs.
- **Access-Local** indicates home access network.
- **Access-Interconnect** indicates visiting access network.
- **Network Property.** Check the Subscriber Traffic box to enable connectivity with User Equipment.
- **T1 (msec).** Round-Trip Time (RTT) estimate.**T1 (msec).** Round-Trip Time (RTT) estimate.
- **T2 (msec).** Maximum retransmit interval for non-INVITE requests and INVITE responses.**T2 (msec).** Maximum retransmit interval for non-INVITE requests and INVITE responses.
- **Timer C (sec).** Proxy INVITE transaction time-out**Timer C (sec).** Proxy INVITE transaction time-out
- **Allow Methods.** Check the allowed methods.

---

**Note:**

INVITE, CANCEL, ACK, BYE, and OPTIONS methods are mandatory for all SIP Parameter Profiles and cannot be de-selected.

---

- **Refer message handling.** Indicates how to handle the Refer message.

- Possible values:
  - **oForward**. The BorderNet SBC forwards the REFER method transparently from one leg to another, without any interference in the call transfer process (default).
  - **oLocal**. The BorderNet SBC handles locally the call transfer process by:
    - Handling the Refer message. Upon receiving a REFER message, the BorderNet SBC replies: *202 Accepted*, to indicate that REFER is handled locally
    - Enabling negotiation between the transferee and the transfer target, including SDP manipulation, to ensure the call establishment between these two parties
    - Handling the embedded *REPLACE* header (if exists), by extracting it from the Refer-To header and adding it as a standalone header in the original Invite towards the transfer target.
    - Handling the Notify messages, sent to the Refer sender, as a call transfer progress indicator.
    - Handling the call transfer process fully, including routing, and the graceful termination of the process.
  - **oDynamic**. The BorderNet SBC can dynamically choose between local and forward options, based on the Refer-To URI routing analysis, as follows:
    - If a valid route exists - apply local handling.
    - If there is no route - apply forwarding (forward the REFER).
    - Dynamic mode actually starts with local handling and switches to forward mode in case the local handling did not find a route. If the route search in the advanced policy returns zero results (no treatment found) then REFER should be forwarded. Note that in local mode, a failure to find a route will cause a final **5xx/4xx** response on the REFER to be sent. However, in dynamic mode the REFER is forwarded rather than rejected.
- **Replaces header handling**. Indicates how to handle the Replaces header.
- Possible values:
  - **oForward** (default). The BorderNet identifies the dialog, modifies the REPLACES header parameters with the current dialog parameters, and forwards the Invite message to the remote leg.
  - **oReplace**. The BorderNet adds the option tag: *Replaces* to the Supported header, for all message, handles the SDP (preserving the session consistency based on rfc3264 & rfc4566), and provides a new Invite with Replaces header, forwarded as a Re-Invite. Upon receiving an Ack, the BorderNet terminates the existing matched dialog.
  - **oReject**. An INVITE message with a REPLACES header is rejected, using a **403 Forbidden** response.
  - **oStrip**. The BorderNet removes the REPLACES header from the incoming INVITE message, and treats this INVITE as a regular INVITE. In other words, the STRIP option causes the BorderNet to handle the INVITE with REPLACES as an INVITE without REPLACES.
- **Min SE**. Minimum time that must elapse prior to session expiry.
- **Max SE**. Maximum time that can elapse prior to session expiry.
- **Session Timer Value**. Periodically refreshes SIP sessions.
- **Initiate INVITE Requiring 100rel**. If set to Yes, the INVITE request contains a Require header with a 100rel tag to initiate a reliable response.
- **Initiate Reliable Provisional Responses**. If set to Yes, a valid status code from the callback function is returned to initiate an automatic response.
- **TGRP Format**. Select the TGRP format using the drop-down menu (RFC 4904 or OTG-DTG).
- **Insert TGRP Info**. Select Yes, to insert the TGRP information.
- Enter the desired number of **Minimum Max-Forwards**.
- Click **Save**.

## 4.5 Media Profile

A **SIP Media Profile** defines the properties the BorderNet SBC uses to handle media sessions. The following table shows the default media profiles's properties:

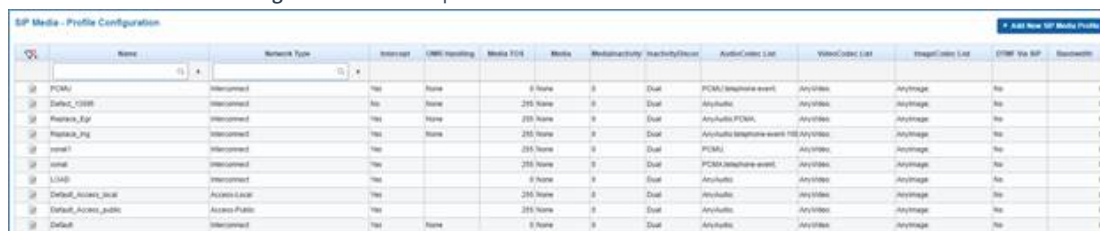
Property	Description	Default
Intercept Media	Determines whether media is intercepted for the session.	Yes
OMR	· OMR Handling. OMR ( <b>Optimal Media Routing</b> ) and <b>Local Break Out</b> (LBO) capabilities introduce mechanisms for providing an optimal media path between roaming users, in IMS networks (based on 3GPP TS 29.079 specifications). · Always Optimal Route. OMR is enabled, bypassing the BorderNet SBC, even if it is in media intercept mode (overrides the Media Intercept parameter) · Consider <b>Intercept Media</b> mode. OMR is enabled (intercepted Media overrides the OMR handling). If intercept media=yes, then local media resources are allocated (BorderNet SBC is not bypassed). · When TFR/Loopback indicator exists. If the SDP offer includes a Feature-capability header with either a TRF or a Loopback indicator, then the BorderNet SBC behaves as <b>Always Optimal Route</b> option. Otherwise the BorderNet SBC behaves as <b>None</b> option.	None
Media TOS	Indicates the SIP media Type of Service for outgoing packets. The range is 0 - 255.	0
Media Latching	Introduces the "latching" mechanism onto the source IP and Port for incoming media packets. Values include None or SDP. · If "None" is selected, there is no latching and the border gateway component can receive packets from any remote location and send the packets backward. · If "SDP" is selected, then the border gateway component will only admit media packets from the source as found in the SDP. Indicates whether media latching is present. Values include None (no media latching) or SDP (a 180 will be sent with an SDP answer).	None
Inactivity	Media Inactivity Timer. Possible values: · 0. The media inactivity mechanism is disabled (default). · [30-1200]. The value of the inactivity timer in multiples of 10 seconds. Inactivity Disconnection. Possible values: · Single peer inactivity (one way). The call is disconnected, if only one peer receives traffic · Dual peer inactivity (two ways). The call is disconnected only if no media is detected on both peers (default).	
Audio Codec Preferred List	Provides a list of available preferred Audio codecs that can be added to the SIP media profile.	AnyAudio
Video Codec Preferred List	Provides a list of available preferred Video codecs that can be added to the SIP media profile.	AnyVideo
Image Codec Preferred List	Provides a list of available preferred Image codecs that can be added to the SIP media profile.	AnyImage
DTMF Via SIP INFO	Indicates support for SIP INFO messages.	Yes

Table 6: Default Media Profile

→ To create a new media profile:

1. Select **Application → SIP Configuration → Media Profile**.

The **SIP Media - Profile Configuration** window opens.



2. Click the **+Add New SIP Media Profile** button.

The **Add SIP Media Profile** window opens.

#### Main tab

- **Name.** The new media profile's name.
- **Network Type.** Select the network type, using the drop-down menu:
  - Interconnect indicates a public network.
  - Local indicates a private network.
  - Access-Public indicates a public network towards UEs.
  - Access-Local indicates home access network.
  - Access-Interconnect indicates visiting access network.
- § **Network Property.** Check the Subscriber Traffic box to enable connectivity with the User Equipment.
- **Intercept Media.** BorderNet SBC intercepts media per-interface.
- Possible values:
  - oYes. Media is intercepted (overridden by the media interception treatment).
  - oNo. Media is not intercepted (overridden by the media interception treatment).
- Flexible. The interception depends on the media transparency selection.
- **OMR Handling.** **OMR (Optimal Media Routing)** and **Local Break Out (LBO)** capabilities introduce mechanisms for providing an optimal media path between roaming users, in IMS networks (based on 3GPP TS 29.079 specifications).
- Possible values:
  - oNone. OMR is disabled.
  - oAlways Optimal Route. OMR is enabled, bypassing the BorderNet SBC, even if it is in media intercept mode (overrides the Media Intercept parameter)
- Consider 'Intercept Media' mode". OMR is enabled (intercept Media overrides the OMR handling). If intercept media=yes, then local media resources are allocated (BorderNet SBC is not bypassed).

When **TFR/Loopback** indicator exists. If the SDP offer includes a Feature-capability header with either a TRF or a Loopback indicator, then the BorderNet SBC behaves as **Always Optimal Route** option. Otherwise the BorderNet SBC behaves as **None** option.

To complete the OMR configuration, set the **OMR** check-box, and provision the **OMR IP Realm** parameter, in the **Peer Configuration** window.

---

#### Note:

If Lawful Interception (LI) / Local Ring-Back tone (LRBT) / Transcoding is activated for a call, then the intercept mode is used and OMR is not applied.

---

- **Media TOS.** Indicates the SIP media type of service for outgoing packets, range [0 - 255].
- **Media Latching.** Introduces the "latching" mechanism onto the source IP and port for incoming media packets.

- Possible values:
  - oNone. No latching is applied. BorderNet SBC receives packets from any remote location and sends the packets backward.
  - oSDP. BorderNet SBC handles the media packets in accordance with the SDP (a 180 will be sent with an SDP answer).
  - oMedia Inactivity Timer.
- Possible values:
  - o0. The media inactivity mechanism is disabled (default).
  - o[30-1200]. The value of the inactivity timer in multiples of 10 seconds.
- **Inactivity Disconnection.**
- Possible values:
  - oSingle peer inactivity (one way). The call is disconnected, if only one peer receives traffic.
  - oDual peer inactivity (two ways). The call is disconnected only if no media is detected on both peers (default).

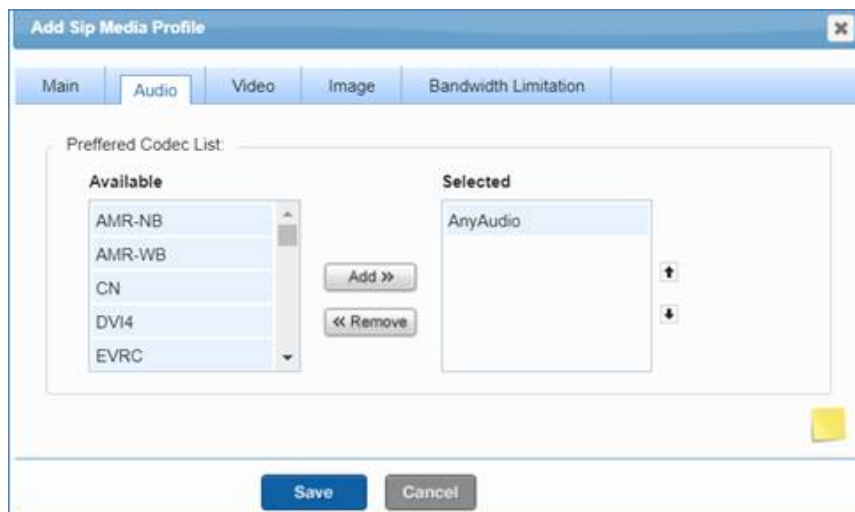
Different peers can be associated with different media profiles and therefore have different inactivity timers. If a peer is not associated with a media profile, the inactivity timer is derived from the associated SIP interface.

The inactivity media detection mechanism is triggered if a media profile of either the ingress or egress legs is configured with a media inactivity timer. If both call legs are configured with an inactivity timer, but each one is assigned with a different media profile using a different value, then the lower inactivity timer value is used.

If both call legs are configured with an inactivity detection, but each one is assigned with a different media profile using a different *Inactivity Disconnection* value, then the *Single peer inactivity (One way)* is used.

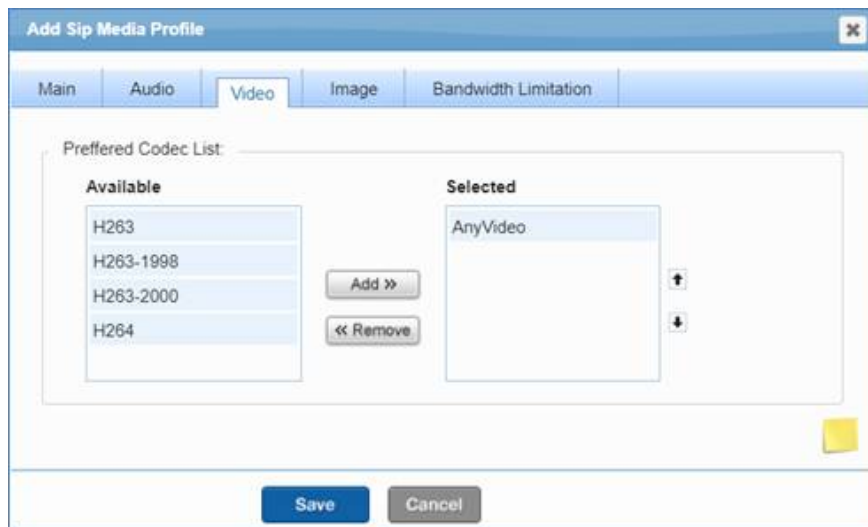
#### Audio tab

1. Select the audio codecs from the **Preferred Codec List**.
2. Use the **Add >** button to move it to the **Selected** list.



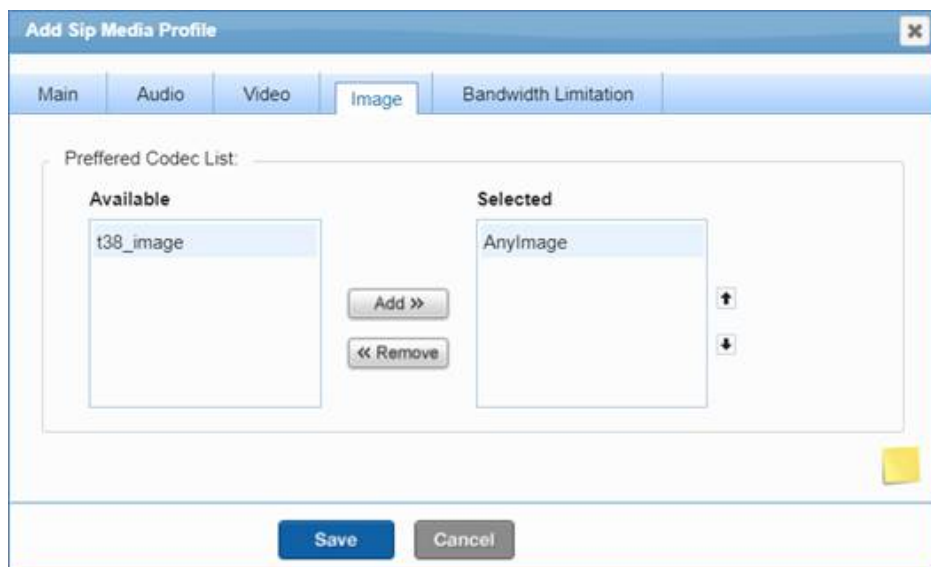
#### Video tab

1. Select the video codecs from the **Preferred Codec List**.
2. Use the **Add >** button to move it to the **Selected** list.



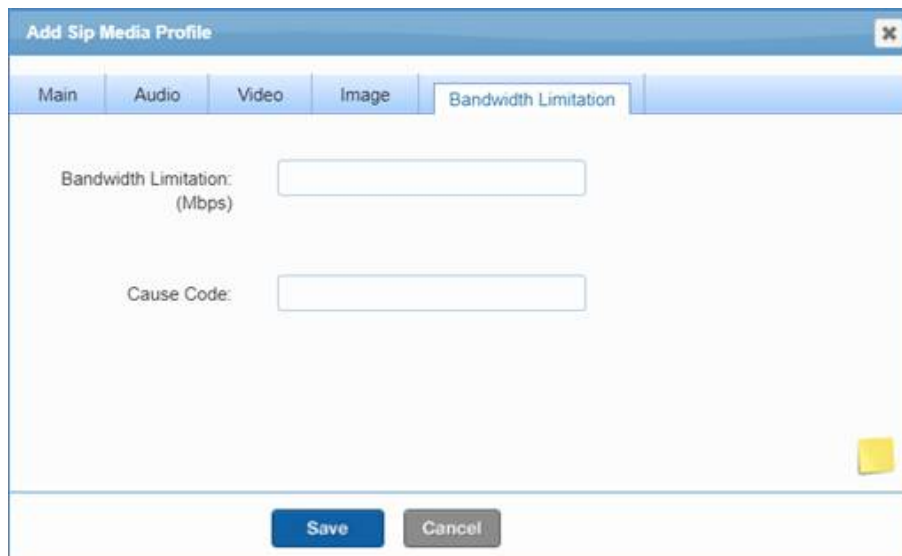
## Image tab

1. Select image from the **Preferred Codec List**.
2. Use the **Add >** button to move it to the **Selected** list.



## Bandwidth Limitation tab

- **Bandwidth Limitation (Mbps)**. The Limit of the bandwidth in mega bits per second.
- **Cause Code**. The cause code presented in the **Release** message when the session is rejected, if the on-going traffic exceeds the defined bandwidth limitation.



Click **Save** when you have finished all the configuration of all the tabs.

## 4.6 App Params

The **App Params** window allows the configuration of SIP behavior in BorderNet SBC.

→ To view the application parameters:

1. Select **Application** → **SIP Configuration** → **App Params**.

The **AppParam Summary** window opens.

ID	Name	Value	Type
1	MAX_DAYS_REPORT	4	Integer
2	DNSQueryRetriesThreshold	2	Integer
3	DNSQueryTimeout	3	Integer
4	MaxAllowedInactiveMediaCallDuration	7200	Integer
5	GenerateSDRForSessionRefresh	True	Boolean
6	WebRTCEnabled	True	Boolean
7	ISD_All_ReconnectRouting	True	Boolean
8	UserOCMHeaders	True	Boolean
9	GenerateSDRForSessionRefresh	True	Boolean
10	ACLPreEnforcement	True	Boolean
11	ExcludeFromSessionRefresh	True	Boolean
12	DisableLocalCallFlow	True	Boolean
13	DisableLocalCallFlow	True	Boolean
14	DisableLocalCallFlow	True	Boolean
15	DisableLocalCallFlow	True	Boolean

2. Select a parameter in the **AppParam Summary** window.

3. Click on the **Edit** button in the first column.

The following table lists the parameters, and description of the system's behavior when it is set to **True**:

Parameter	Description
DNSQueryRetriesThreshold	Maximum number of attempted retries towards the DNS server (integer).
DNSQueryTimeout parameters	The time interval that the BorderNet waits for the DNS to respond (sec - minimum 1 sec).
MaxAllowedInactiveMediaCallDuration	When the call duration exceeds this value, the call is detected as a hanging call, and is released. Defined in range: [60-1200], in multiples of 10 seconds, and when set to 0 (zero- default), this capability is disabled.
GenerateSDRForSessionRefresh	Create SDR for refresh messages (i.e. <i>ReInvite</i> ).

Parameter	Description
PortReuseForHold	Reuse the Invite message's port for call hold (another Invite is sent with send-only).
PATSupported	If the IP addresses in the IP packet and via are the same, and the ports are different, it is assumed that the packet has been traversed through PAT. In this case the port in the IP packet is considered and the port from via is ignored.
PortReuseForReinvite	Reuse the Invite message's port for the <i>ReInvite</i> message.
DisableLocalAck	For <i>Ack</i> messages, the BorderNet SBC stops acting as a B2B, and acts as a proxy (for example the incoming <i>200 OK</i> is forwarded as is, and the <i>Ack</i> for it is sent only when <i>Ack</i> is received as an acknowledgement for the <i>200 OK</i> ).
DisableLocalByeResp	For <i>Bye</i> messages, the BorderNet SBC stops acting as a B2B, and acts as a proxy.
DisableLocalCancelResp	For <i>Cancel</i> messages, the BorderNet SBC stops acting as a B2B, and acts as a proxy.
DisableResolvePeersOnRedirection	The call is redirected without checking versus the existing peers table.
EnableCallMonitorThread	The standby platform takes over if within a ten minutes interval no session information is mirrored.
ExcludeFromHeaderForPeerIdentification	For peer identification, only <i>via</i> and <i>contact</i> headers are checked. The <i>from</i> header is ignored.

Table 7: Application Parameters

## 4.7 SRTP Profile

Secure RTP (SRTP) is a protocol used to encrypt RTP media between two entities, enabling media confidentiality and message authentication. SRTP profiles represent the SRTP specifications.

For detailed information on SRTP, see the *BorderNet SBC SRTP User's Guide* document.

→ To create an SRTP Profile:

1. Select **Application** → **SIP Configuration** **SRTP Profile**.

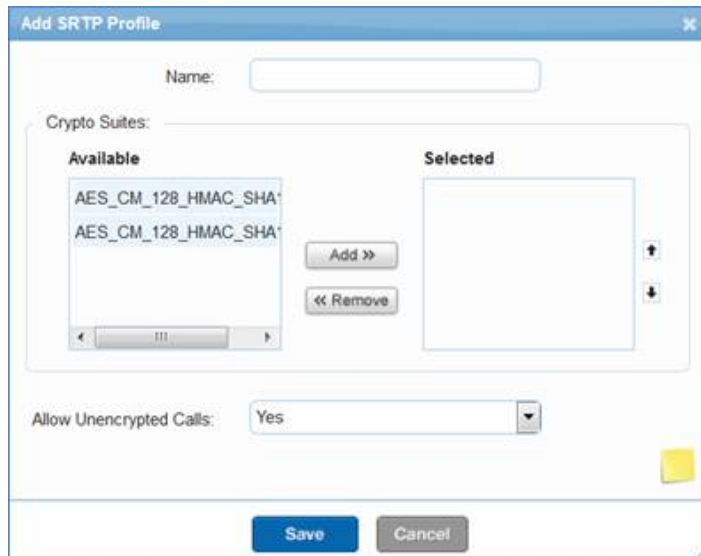
The **SRTP - Profile Configuration** opens.

Name	Crypto Suites	Allow Unencrypted Calls
demo	AES_CM_128_HMAC_SHA1_80	Yes

2. Click on **+Add New SRTP Profile**.

The **Add SRTP Profile** window opens.





3. Enter the following parameters:

- **Name.** Enter the name of the SRTP profile
- **Crypto-Suite.** Assign a crypto-suite to the profile. Each crypto-suite can be added or removed, as well as shifted up or down in the selected list. It is mandatory to configure at least one crypto-suite.
- **Allow Unencrypted Calls.** If set to **Yes**, BorderNet SBC initiates SRTP encryption for the outgoing calls, and accepts the incoming calls with regular RTP offer (without SRTP encryption). Otherwise BorderNet SBC initiates SRTP for the outgoing calls and accepts only SRTP-encrypted calls. Incoming calls without SRTP offer or SDP answers without SRTP's acceptance are rejected with *488 Bad Crypto negotiation* response.

4. Click on **Save**.

---

**Note:**

SRTP is allowed only when TLS is selected as transport protocol, for a SIP interface. Upon selecting TLS, the SRTP Profile drop-down menu allows the user to assign an SRTP Profile to the interface.

---

## 4.8 Registration

The confirmed user-equipment registration information is stored in BorderNet's cache memory. Multiple devices registrations are supported.

BorderNet SBC supports internal **Port Allocation Table (PAT)** for topology hiding.

For the **Access-Local** and **Access-Interconnect** network types, the operator is enabled to determine if the registration ports can be reused, relying on cache information or the configured port allocation table is used. The port allocation table can be used for all kinds of media or for registration only.

For **Access-Local** and **Access-Interconnect** network types, the check-box **RegPortReuse** is presented.

- If checked, the Port Allocation tab is not displayed and the registration ports can be reused. This employs the existing functionality of relying on cache-identifiers.
- If unchecked, the **Port Allocation** tab can be selected.

Ports that have been set in the **Port Allocation** table are displayed in the **Available** ports section.

→ To select desired registration ports:

1. Use the **Add** and **Remove** buttons to select the desired **Registration** ports.

2. Click **Save**.

The total IP and ports allocated on a specific **Access-Local** interface for this mapping corresponds to the total number of remote UEs that register with the core network via this **Access-Local** interface.

---

**Note:**

As long as the new SIP Interface has not been saved, the user can toggle the RegPortReuse field. Once the SIP Interface is saved, the RegPortReuse cannot be changed.

---

## 4.9 Trunk Authentication

When the **Trunk Authentication** checkbox is marked, it reveals three more parameters to be configured.

The screenshot shows a configuration window titled "Add Sip Peer". The fields are as follows:

- Status:  ON
- Name:
- Class ID:
- Network Type: Interconnect (dropdown)
- Network Property:  IMS  OMR
- Trunk Authentication:  Trunk-Authentication
- Peer AOR (user part): 97239701111
- Auth Username: dialogic\_1
- Auth Password: de9Cd08

- **Peer AOR (user part)**. Enter the user part of the 'To' header in the **Register** request, which is the trunk pilot number to register.
- **Auth Username**. The user name used to construct the **Authorization** header of the **Register** request.
- **Auth Password**. The password used to construct the **Authorization** header of the **Register** request.

## 4.10 WebRTC Support

The main goal of WebRTC is to offer real time communication natively from a web browser. It is a framework that enables peer to peer connections and allows exchange of audio, video and data between connected web browsers.

This framework includes a collection of communications protocols and APIs that enable real-time peer to peer connections within the browser.

Traditionally, these interfaces have been delivered by plugins, which had to be downloaded and installed separately from the browser.

WebRTC introduces the possibility of making those interfaces available in a standardized way within the browser.

WebRTC works only on the Access Public interface type and there is also a WebRTC Gateway between the WebRTC and SIP.

The total WebRTC effort consists of two major parts, each consisting of multiple documents:

- IETF protocol specification - describes the different network protocols to be supported when implementing WebRTC.
- World Wide Web Consortium (W3C) JavaScript API specification - describes a set of APIs, embedded in the client browser, which enable a JavaScript code using it to establish a real time connection between browsers.

WebRTC call setup has been designed to focus on controlling the media plane, leaving signaling plane behavior up to the application as much as possible. The rationale is that different applications may prefer to use different protocols, such as the

existing SIP call signaling protocol, or something custom to the particular application, perhaps for a new use case. In this approach, the key information that needs to be exchanged is the multimedia session description, which specifies the necessary transport and media configuration information necessary to establish the media plane.

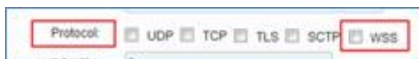
The BorderNet's deployment will obviously use SIP as the signaling protocol, sent as SIP over WebSocket.

BorderNet shall support the secured **WebSocket** protocol (**WSS**), for connecting with peers.

The **WSS** is a WebSocket protocol on top of a TLS connection. When selecting WSS from the interface configuration screen, then TLS profile will appear as well.

Protocols implemented for **WebRTC** support include the following:

- WebSocket Secured (WSS)
- ICE-Lite
- STUN connectivity checks
- DTLS-SRTP
- RTCP-Mux
- RTCP-Based Feedback (RTP/AVPF)
  - Audio+Video
  - Transparent transfer of SDP attributes and RTCP packets



Selection of WSS Protocol

## 5. H.323 Configuration

BorderNet SBC provides an **H.323 Interworking Function (IWF)** that enables an H.323 network to connect to a SIP network. **H.323-to-SIP**, **SIP-to-H.323**, and **H.323-to-H.323** (fast-start only) sessions are supported.

BorderNet SBC uses an H.323 interface (a virtual gateway) and an H.323 peer (a model for a remote endpoint).

Within the peering network, the BorderNet SBC can be configured in two modes:

- Direct. Signaling occurs directly between the BorderNet SBC and a remote gateway.
- Gatekeeper-managed. The H.323 interface registers with a gatekeeper and is managed by the gatekeeper. In this mode, call signaling can go through the gatekeeper or directly between the endpoints.

BorderNet SBC supports H.323 version 4, with the following features and codecs:

- Audio, T38 Fax and DTMF (2833, out-of-band)
- ToS field setting for signaling and media
- G711, G723, G729 codecs
- EVS, EVRC codecs
- H.323 protocol features:
  - Fast-start and slow-start call scenarios
  - Fast-start to slow-start translations
  - H.245 tunneling mode
  - Mid-call Codec changes
  - Third-party pause and re-routing scenarios
  - Early media handling
  - CLIP/CLIR interworking

The following configuration is required to setup an H.323 session:

- H.323 interface, peer, and peer-interface association
- SIP interface, peer, and peer-interface association
- Static routing with a specified peer

Configuration can be fine-tuned using different profiles. Profiles (**Parameter**, **Media**, **Security** and **Service**) are mandatory for interfaces and optional for peers.

### 5.1 Interface

The first step in configuring the H.323 application is to create an H.323 interface.

**The H.323 Interface Summary** screen shows the default H.323 interfaces on the BorderNet SBC.

→ To create an H.323 Interface:

1. Select **Application** → **H.323 Configuration** → **Interface**.

1. The **H.323 Interface** window opens.

H323 Interface + Add New H323 Interface

Status	Name	Network Type	Signaling IP	Signaling Port	TGRP Context	Parameter Profile	Media Profile	Service Profile	Security Profile	Trust Level	Allow Associated Peers Only	Interface Model
<input type="checkbox"/> ON	OUT_NO_2020_H323	Local	10.20.20.47	1720		Default	Default	NO_BH2020_BH400	H323-Security-Local	High	<input type="checkbox"/>	Direct
<input type="checkbox"/> ON	IN_NO_2020_H323	Interconnect	10.20.20.46	1720		Default	Default	Default	H323-Security-Interco	High	<input type="checkbox"/>	Direct

2. Click the **+Add New H.323 Interface** button, and enter the following parameters:

Add H323 Interface

Status:  ON  OFF

Name:

Network Type:

Signaling IP:

Signaling Port:

Time Zone:

Signaling TOS:

TGRP Context:

Parameter Profile:

Media Profile:

Service Profile:

Security Profile:

Trust Level:

Associated Peers:

Interface Model:

- o **Status.** Enable/disable the H.323 Interface, by selecting ON or OFF (default).
  - o **Name.** The unique name of the H.323 Interface.
  - o **Network Type.**
  - o Possible values:
    - oInterconnect indicates a public network.
    - oLocal indicates a private network.
  - o **Signaling IP.** Select the signaling IP address, using the drop-down menu. This menu is populated with IP addresses that were created during the [VLAN configuration](#).
  - o **Signaling Port.** Use the default value (1720), or enter a different port.
  - o **Time Zone.** Double-click the **Time Zone** field to select the time zone for this interface.
  - o **Signaling TOS.** Enter the TOS value. By default, this value is 0.
3. To modify the TOS value, double-click on the TOS field.

1. The **Edit TOS Bit Values** screen appears.

EDIT TOS bit values

DSCP Mode:

Class/DSCP Value:

TOS Value:

2. Select the **Differentiated Service Point Code (DSCP)** from the DSCP Mode drop-down menu.

**Note:**

The Class/DSCP Value is automatically populated with corresponding selections based on the DSCP Mode selected. The TOS Value is automatically populated with the appropriate value based on the Class/DSCP Value selected.

3. Click **Save** to add the selected **TOS Value** to the H.323 Interface.
  - o **TGRP Context.** Enter the TGRP Context value.
  - o Select the desired profile from the drop-down menus.
  - o **Parameter Profile.** Determines which H.323 parameters are applied to the traffic.
  - o **Media Profile.** Manages traffic aspects such as port allocations and Codec configurations.
  - o **Service Profile.** Controls the assigned routing profiles.
  - o **Security Profile.** Imposes session constraints and rate limits.
  - o **Trust Level.** Select the **Trust Level** from the drop-down list. Possible values: **High, Medium, or Low.**
  - o **Associated Peers.** Set the check-box to allow only traffic from the associated peers. Otherwise traffic from all peers is allowed.
  - o **Interface Model.**
    - o Possible values:
      - o **Direct.** Direct signalling interface with the remote endpoint - not managed by the gatekeeper.
      - o **GateKeeper Managed.** Signaling is managed and monitored by the gatekeeper. The BorderNet SBC provides auto-discovery for gatekeepers, alternate gatekeeper handling, and additive registrations. Upon this selection, the following additional options are displayed:

Interface Model:	GateKeeper Managed
RAS Port:	1719
H323 Id:	1555
H323 Zone:	45
Tech Prefix:	408, 510, 650
Number of RAS retransmissions:	2
RAS response timeout:	3
GK Registration Timeout:	10

- o The **RAS Port** communicates with the gatekeeper.
  - o The **Tech Prefix** tells the gatekeeper which prefixes to serve.
4. Click **Save** to add the H.323 interface to the system.

## 5.2 Peer

A peer models a remote entity.

Peers are created and associated with H.323 interfaces to facilitate call routing on the BorderNet SBC.

→ To create an H.323 peer:

1. Select **Application** → **H.323 Configuration** → **Peer**.  
The **H.323 Peer** window opens.

H323 Peer														+ Add New H323 Peer	
Status	Name	Class ID	Source Type	Source List	Trust Level	Host FQDN/IP	Host	Port	TGRP ID	Parameter Profile	Media Profile	Service Profile	Security Profile	Peer Type	
ON	OUT_H323_2020_NO_2020_H323		Single	IPv4.10.20.20.52/High	High	IPv4	10.20.20.52	1720						Direct	
ON	H323_IN_2020_NO_2020_H323		Single	IPv4.10.20.20.40/High	High	IPv4	10.20.20.40	1720		NoTunneling				Direct	

2. Click the **+Add New H.323 Peer** button to add a new peer.
3. Enter the following parameters:

- o **Status.** Enable/disable the H.323 peer, by selecting **ON** or **OFF**.
- o **Name.** The unique **Name** of the H.323 peer.
- o **Class ID.** The Classification Identifier, a string that enables the logical grouping of peers.
- o **Source List.** Select between **IPv4** and **IPv6**, and enter a set of IP addresses from which SIP messages are received, into the Source List field. A peer can be:
  - o IP address and port (for example, 10.13.4.108 and 5060, port is optional).
  - o IP address, subnet mask and Port (for example, 10.13.4.108/24).
- o **Trust Level.** Select the **Trust Level** from the drop-down list.
- o Possible values: **High**, **Medium**, or **Low**.
- o **Host.** Select IPv4, IPv6 or FQDN of the peer (the destination peer to which traffic will be sent. Possible values:
  - o IP
  - o FQDN.destination
- o **Port.** Select the port.
- o **Time Zone.** Double-click to select the desired time zone.
- o **TGRP ID.** **Trunk Group** id (a string that contains up to 100 characters).
- o Profiles are mandatory for H.323 peers. If no custom profile is assigned to the peer, a default profile will be attached to it. Select profiles using the drop-down menus:
  - o **Parameter Profile.** Captures the SIP-specific parameters that need to be configured and influences session behaviors. This profile is for general purpose and core protocol-specific parameters.
  - o **Media Profile.** Captures all media-related parameters and configurations, including port allocations and codec configurations.

- **Service Profile.** Captures session-impacting services (these services are usually not part of core session behavior), such as routing, redirection, and transparency settings.
  - **Security Profile.** Captures the relevant security properties to be exercised on the sessions. Security properties include control mechanisms such as rate-limiting on sessions and packets, maximum concurrent sessions, blacklisting, and so forth.
  - **Peer Type.** Possible values:
    - **Direct.** Creates a peer that is managed by the BorderNet SBC.
    - **GateKeeper Managed.** Creates a peer that is managed by a remote gatekeeper (see [Interface](#) above).
4. Click **Save** to add the **H.323 Peer** to the system.

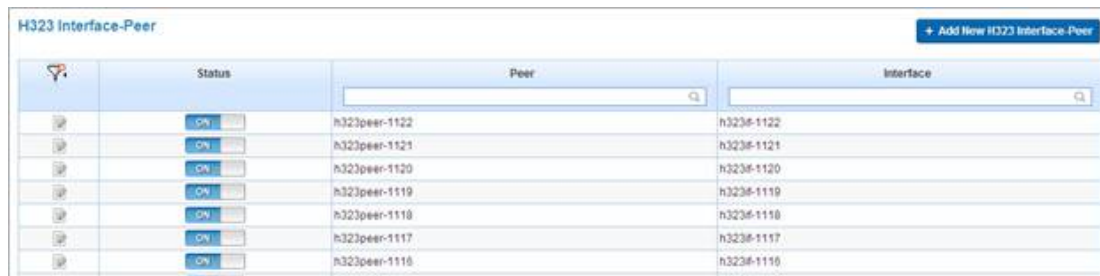
## 5.3 Interface-Peer

Associations are created between H.323 Interfaces and Peers to manage traffic.

→ To create an H.323 Interface-Peer association:

1. Select **Application** → **H.323 Configuration** → **Interface-Peer**.

The **H.323 Interface-Peer** window opens.



	Status	Peer	Interface
	<input type="checkbox"/> ON	h323peer-1122	h323f-1122
	<input type="checkbox"/> ON	h323peer-1121	h323f-1121
	<input type="checkbox"/> ON	h323peer-1120	h323f-1120
	<input type="checkbox"/> ON	h323peer-1119	h323f-1119
	<input type="checkbox"/> ON	h323peer-1118	h323f-1118
	<input type="checkbox"/> ON	h323peer-1117	h323f-1117
	<input type="checkbox"/> ON	h323peer-1116	h323f-1116

2. Click the **+Add New H.323 Interface-Peer** button.

The **Add H.323 Interface-Peer** screen appears.



Add H.323 Interface-Peer

Status:  ON

Peer: h323peer-1147

Interface: h323f-1304

**Save** **Cancel**

3. Set the following parameters:
  - **Status.** Enable/disable the association by selecting ON or OFF.
  - **Peer.** The associated peer.
  - **Interface.** The associated interface.
4. Click **Save** to create the association.

## 5.4 Parameter Profile

H.323 parameter profiles are applied to a peer (optional) or to an interface (mandatory).

Only Application Administrators can create, modify or delete profiles.

The default following table defines the default H.323 parameter profiles:



Property	Profile Description	Default
H.245 Tunneling	<ul style="list-style-type: none"> <li>If enabled, the message is tunneled with H.323.</li> <li>If disabled, the H.245 message will have its own connection.</li> </ul>	Yes (enabled)
H.245 Stage	<p>When an H.245 message needs to be exchanged in a call, this parameter indicates at what stage that will transpire. Values are:</p> <ul style="list-style-type: none"> <li>Early</li> <li>Setup</li> <li>Call Processing</li> <li>Alerting</li> <li>Connect</li> <li>Facility</li> <li>No245</li> </ul>	Connect
Egress Call Model	<p>Indicates the call mode. Values are:</p> <ul style="list-style-type: none"> <li>Transparent-the call goes through as is</li> <li>Slow-Start-the call is converted to a slow-start call</li> </ul>	Transparent
TCP Keep Alive Time	Indicates how frequently to check the connection.	1800 sec

→ To create a new parameter profile:

1. Select **Application** → **H.323 Configuration** → **Parameter Profile**.

The **H.323 Parameter - Profile Configuration** window opens.



2. Click on the **+Add New H.323 Parameter Profile** button, and enter the following parameters.



- **Name.** The name of the new profile.
- **H.245 Tunneling.** If enabled, the H.245 is tunneled with the H.323 the messages, otherwise it will run on its own port.
- **H.245 Stage.** This parameter indicates the stage that the H.245 messages are exchanged. Possible values: **Early Setup, Call Processing, Alerting, Connect, Facility, No245**
- **Egress Call Model.** Indicates the call mode.
- Possible values:
  - **Transparent.** The call goes through as is.
  - **Slow-Start.** The call is converted to a slow-start call.
- **TCP Keep Alive Time.** Indicates how frequently to check the connection.

3. Click **Save**.

## 5.5 Media Profile

An **H.323 Media Profile** defines the properties the BorderNet SBC uses to handle media sessions.

Only Application Administrators can create, modify, or delete profiles.

The following table defines the default H.323 Media Profile:

Property	Description	Default
Intercept Media	Determines whether media is intercepted from incoming traffic.	Yes
Port Allocations	Provides a list of available media ports that can be added to the SIP media profile (see <b>Port Allocation Table</b> ).	None
Media TOS	Indicates the H.323 media Type of Service for outgoing packets. Range is 0 - 255.	0
Media Latching	Indicates whether media latching is present. Values include None (no media latching) or SDP.	None
Audio	Provides a list of available preferred Audio Codecs that can be added to the H.323 media profile.	AnyAudio
Image	Provides a list of available preferred Image codecs that can be added to the H.323 media profile.	AnyImage
DTMF Via H.245UI	Indicates support for H.245UI messages.	Yes

→ To create a new media profile:

1. Select **Application** → **H.323 Configuration** → **Media Profile**.

The **H323 Media-Profile** window opens.

Name	Intercept Media	Port Allocations	Media TOS	Media Latching	AudioCodec List	ImageCodec List	DTMF Via H.245UI	H.323 Local Capability Exchange
Default	Yes		0	None	AnyAudio	AnyImage	Yes	Yes

1. Click the **+Add New H.323 Media Profile** button and enter the following parameters.

## Main tab

- **Name.** The new media profile's name.
- **Intercept Media.** BorderNet SBC intercepts media per interface.
- Possible values:
  - **Yes.** Media is intercepted (overridden by the media interception [treatment](#)).
  - **No.** Media is not intercepted (overridden by the media interception [treatment](#)).
  - **Flexible.** The interception depends on the media [transparency](#) selection.
  - **Media TOS.** Indicates the H.323 media type of service for outgoing packets, range [0 - 255].
  - **Media Latching.** Introduces the "latching" mechanism onto the source IP and port for incoming media packets.
- Possible values:
  - **None.** No latching is applied. BorderNet SBC receives packets from any remote location and sends the packets backward.
  - **SDP.** BorderNet SBC handles the media packets in accordance with the SDP (a 180 will be sent with an SDP answer).
  - **DTMF via H.245UI.** If set to **Yes**, DTMF is transferred/received via H.245UI.

## Audio tab

1. Select the audio codecs from the **Preferred Codec** list.
2. Use the **Add>>** button to move it to the **Selected** list.

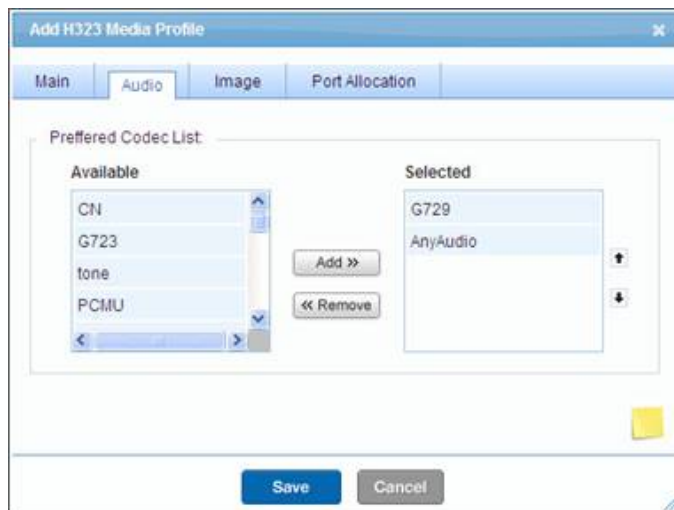
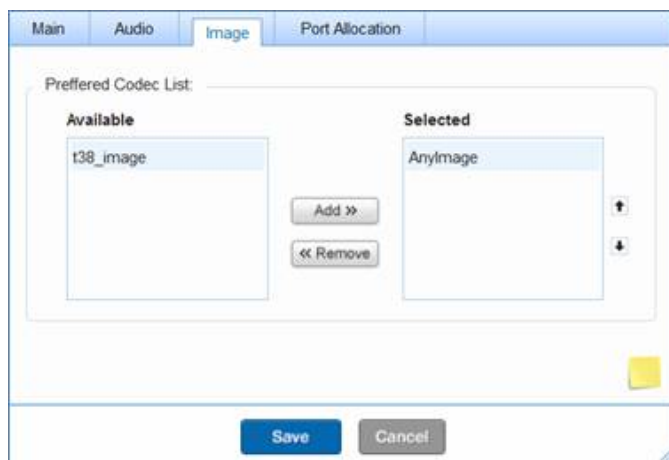


Image tab

1. Select the image from the **Preferred Codec** list.
2. Use the **Add>>** button to move it to the **Selected** list.



Port Allocation tab

1. Select the port(s) to be allocated from the **MediaPort Allocation** list.
2. Use the **Add>>** button to move them to the **Selected** list (see also [Port Allocation Table](#)).
3. Click **Save**.

## 5.6 Global Parameters

Global parameters are applicable to all H.323 interfaces.

→ To set a global parameter:

1. Select **Application** → **H.323 Configuration** → **Global Parameters**.
2. Edit the fields as appropriate.
3. Click **Save**.

The default values are shown below.

### H323 Global Parameters

Max H323 Calls:	<input style="width: 90%;" type="text" value="10000"/>
Max Channels per call:	<input style="width: 90%;" type="text" value="2"/>
No Answer Timeout:	<input style="width: 90%;" type="text" value="180"/>
Setup response Timeout:	<input style="width: 90%;" type="text" value="4"/>
Request Mode Timeout:	<input style="width: 90%;" type="text" value="5"/>
MSD Timeout:	<input style="width: 90%;" type="text" value="3"/>
TCS Timeout:	<input style="width: 90%;" type="text" value="5"/>

---

---

**Note:**

Max H.323 Calls indicates the maximum number of H.323 calls that can occur simultaneously in the system. The BorderNet SBC IBCF Service must be restarted if this parameter is changed.

---

## 6. Security Configuration

BorderNet SBC protects VoIP elements and infrastructure, ensures service availability and quality, prevents service and bandwidth theft, and provides authentication, integrity and confidentiality to a network (see also [Traffic Policing](#)).

BorderNet SBC employs security on multiple levels:

Layer 3 and Layer 4 Security	Application Security	General
<ul style="list-style-type: none"> <li>• Detects and drops malformed or malicious TCP/IP packets</li> <li>• Access Control Lists</li> <li>• Dynamic pinholes for media</li> <li>• Traffic policing</li> <li>• Topology hiding for media</li> </ul>	<ul style="list-style-type: none"> <li>• Detects and drop malformed or malicious SIP/H.323 messages (ALG)</li> <li>• Topology hiding for SIP/H.323 sessions (B2BUA)</li> <li>• Authentication, Integrity, Confidentiality measures (TLS, SSH)</li> <li>• Session constraints</li> <li>• Dynamic blacklisting</li> </ul>	<ul style="list-style-type: none"> <li>• Packet consistency checks (incorrect checksum, truncated packets etc.)</li> <li>• Fragmented IP packet checks (fragment overflow, short fragments, overlapping fragments)</li> <li>• Protocol consistency checks:</li> <li>• Large ICMP packets, prohibited ICMP messages</li> <li>• Packets with fragmented TCP/UDP headers</li> <li>• Known TCP/IP vulnerability management (TCP xmas, LAND attacks, and so forth)</li> </ul> <p>The BorderNet SBC automatically applies malicious TCP/IP packet handling. No configuration is required.</p>

### 6.1 Security Profile

BorderNet SBC provides session admission capabilities based on a set of constraints that are defined in a network security profile. The following table shows the default security profile values:

Constraint	Description	Default	Range
<b>INVITE Sessions</b>			
Session Rate Incoming	The allowed session rate (sessions per second) from a peer or at a SIP/H.323 Interface.	30	0-500
Session Rate Outgoing	The outgoing session rate (sessions per second) to a peer or from a SIP/H.323 Interface.	30	0-500
Active Sessions	Maximum number of concurrent sessions allowed, including sessions that are in progress (being established).	3600	0-100,000
Active Sessions Incoming	Maximum number of concurrent sessions allowed from a peer or at a SIP/H.323 Interface.	1800	0-100,000
Active Sessions Outgoing	Maximum number of concurrent sessions sent to a peer or at a SIP/H.323 Interface.	1800	0-100,000
Max Emergency Sessions	Maximum number of emergency sessions sent to a peer or at a SIP/H.323 Interface.		
Burst Rate	Burst rate is used to accommodate occasional traffic spikes and is defined as a percentage of the other defined constraints.	0	0-100
Burst Rate Interval	Defines how long the burst can be allowed (in seconds).	3	0-10
<b>Non-INVITE Txns</b>			

Constraint	Description	Default	Range
Txn Rate Incoming	Sustained rate for incoming transactions.	100	0-2,000
Txn Rate Outgoing	Sustained rate for outgoing transactions.	100	0-2,000
Burst Rate	Burst rate is used to accommodate occasional traffic spikes and is defined as a percentage of the other defined constraints.	0	0-100
Burst Rate Interval	Defines how long the burst can be allowed.	3	0-10
<b>Dynamic Blacklisting</b> - To prevent DoS attacks, the BorderNet SBC temporarily black-lists suspicious peers. For example, if a peer has high incoming session attempts/packet rate higher than expected/ or too many malformed messages, the BorderNet SBC adds the peer to the black-list and raises an alarm. The peer is automatically removed from the black-list after a specified amount of time. The application administrator can provision this functionality by setting the following three parameters:			
Malformed Message Count	Indicates the number of malformed messages that are received before a peer is blacklisted.	450	0-99,999
Threshold	Indicates the threshold above which the peer is blacklisted.	200	0-1,000
Blocking Period	Indicates the amount of time an entry is in the dynamic blacklist.	60 seconds	0-300
Is Packet Rate Dynamic?	Specifies whether a packet should be dynamically calculated.	Yes	Yes/No
Packet Rate	Indicates the packet rate expected from the peer.	0	0-9,999

Table 8: Security Profile

A security profile can be applied to an interface or a peer. If it is assigned to both interface and peer, first for the peer profile is used and then the interface profile. This is a session constraint security feature. When a security profile is applied, the BorderNet SBC rejects any session attempts beyond the configured value and raises an alarm if the session attempts are 10% greater than the security profile allows.

→ To add a security profile:

1. Select **Application** → **Security Configuration** → **Security Profile**.

The **Security - Profile Configuration** window opens.

Name	Network Type	Session Rate Incoming (spk)	Session Rate Outgoing (spk)	Active Sessions	Max Emergency Sessions	Active Sessions Incoming	Active Sessions Outgoing	Burst Rate (%)	Burst Rate Interval (secs)	Malformed Message Count	Threshold (%)	Blocking Period (secs)	Is PkRate Dynamic	Packet Rate
Load_Prof	Interconnect	1000	1000	100000	0	100000	100000	0	10	450	200	60	Yes	
Default	Interconnect	30	30	3000	0	1000	1000	10	3	450	200	60	Yes	120

2. Click **+Add New Security Profile** and edit the parameters described below.

The dialog box shows the following configuration:

- Name: sjc\_access
- Network Type: Local
- Network Property:  Subscriber Traffic

3. Click **Save** to create the Security Profile.

The security profile has now been added to the **Summary** screen and can be applied to peers and interfaces.

## General tab

- **Name.** The security profile name.
- **Network Type.** Select the network type using the drop-down menu:
- **Interconnect** indicates a public network.
- **Local** indicates a private network.
- **Access-Public** indicates a public network towards UEs.
- **Access-Local** indicates home access network.
- **Access-Interconnect** indicates visiting access network.
- **Network Property.** Check the **Subscriber Traffic** box to enable the connectivity with the User Equipment.

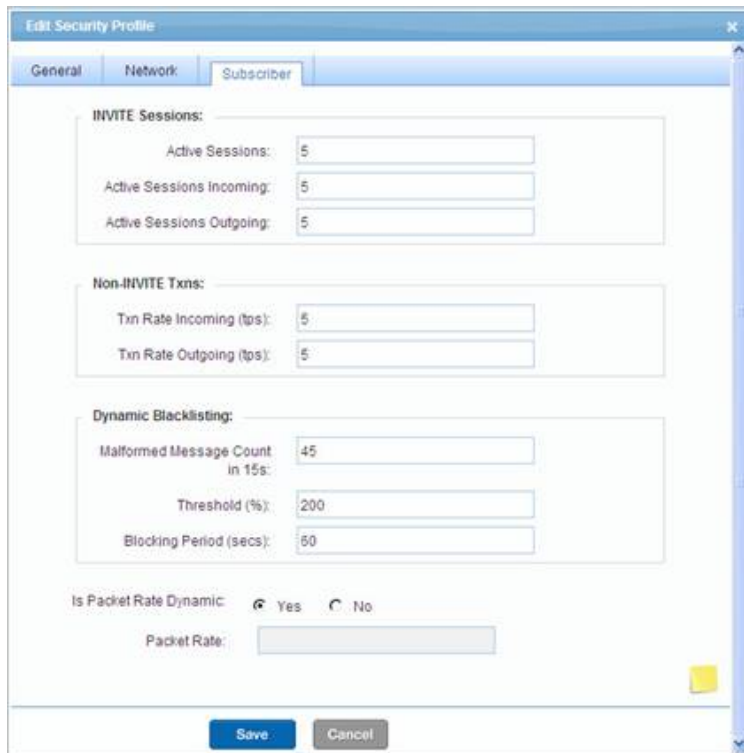
## Network tab

General	Network
<b>INVITE Sessions:</b>	
Session Rate Incoming(sps):	30
Session Rate Outgoing(sps):	30
Active Sessions:	3600
Active Session Incoming:	1800
Active Session Outgoing:	1800
Max Emergency Sessions:	0
Burst Rate (%):	0
Burst Rate Interval (secs):	3
<b>Non-INVITE Txns:</b>	
Txn Rate Incoming (tps):	100
Txn Rate Outgoing (tps):	100
Burst Rate (%):	0
Burst Rate Interval (secs):	3
<b>Dynamic Blacklisting:</b>	
Malformed Message Count in 15s:	450
Threshold (%):	200
Blocking Period (secs):	60

1. Edit the security specifications for the profile, based on the parameters in Table 6 above.

## Subscriber tab (for Access-Public only)





1. Enter the values based on the table below.

The following table shows the ranges and default values for the **Subscriber** tab:

Constraint	Description	Default	Range
<b>INVITE Sessions</b>			
Active Sessions	Maximum number of concurrent sessions allowed, including sessions that are in progress (being established).	5	0-100,000
Active Sessions Incoming	Maximum number of concurrent sessions allowed from a peer or at a SIP Interface.	5	0-100,000
Active Sessions Outgoing	Maximum number of concurrent sessions sent to a peer or at a SIP Interface.	5	0-100,000
<b>Non-INVITE Txns</b>			
Txn Rate Incoming	Sustained rate for incoming transactions.	100	0-2,000
Txn Rate Outgoing	Sustained rate for outgoing transactions.	100	0-2,000
<b>Dynamic Blacklisting</b>			
Malformed Message Count	Indicates the number of malformed messages that are received before a peer is blacklisted.	45	0-99,999
Threshold	Indicates the threshold above which the peer is blacklisted.	200	0-1,000
Blocking Period	Indicates the amount of time an entry is in the dynamic blacklist.	60 sec.	0-300

Table 9: Subscriber Parameters

→ To edit a security profile:

1. Select **Application** → **Security Configuration** → **Security Profile**.  
The **Security - Profile Configuration** window opens.

2. Select a profile from the list.
3. Choose **Edit** from the note icon drop-down menu.
4. Edit the relevant parameters.
5. Click **Save** to update the Security Profile.

## 6.2 Access Control List

BorderNet SBC employs application-aware **Access Control Lists (ACLs)** to selectively allow or drop traffic from remote entities. This is applicable for both signaling and management traffic.

ACLs can be automatically configured by the BorderNet SBC based on the signaling interface and peer associations, or an Application Administrator can configure ACLs for management or signaling traffic:

- The signaling IP and port of the SIP or H.323 interface forms the local IP and local port.
- The source list and protocol defined for a peer form the remote information.

To view all the ACLs (auto-generated and user-added), select **Diagnostics → System Status → ACL Status** (see *BorderNet SBC Maintenance Guide*).

By default, the BorderNet SBC drops all packets. Accept in ACL allows the traffic for any application (SIP/H.323/management).

When the system is initially deployed, the default ACLs are automatically added to allow GUI access (the ACLs allow HTTP access from same subnet as the management address).

→ To create a user-defined ACL:

1. Select **Application → Security Configuration → Access Control List**.

The **Security Access Control List Summary** window opens:

Status	Name	Application	Action	IP Address Type	Remote IP	Remote Netmask	Remote Port	Local Entity
<input checked="" type="checkbox"/>	SSH	Management	Accept	IPv4	0.0.0.0	0	0	SecureShell
<input checked="" type="checkbox"/>	DefaultManagement	Management	Accept	IPv4	0.0.0.0	0	0	https

2. Click the **+Add New ACL** button.

The **Add Security ACL** screen appears.

3. Enter the following parameters:
  - **Status**. Enable/disable the ACL by selecting **ON** or **OFF**.

- **Name.** Enter a name for the new ACL.
- **Application.** Select the application using the drop-down menu.
- Possible values: **SIP, H.323, Media, Management.**
- **Action.** Select an action using the drop-down menu.
- Possible values: **Drop/Allow.**

**Note:**

Drop ACLs take precedence over Allow ACLs.

- **Remote IP.** Select IPv4 or IPv6 and enter the Remote IP.
  - **Remote Netmask.** Enter the remote mask.
  - **Remote Port.** Enter the remote port.
  - **Local Entity.** Select the local entity to which the ACL applies, using the drop-down menu.
4. Click **Save**.

The new ACL appears in the **Security Access Control List Summary** window.

## 6.3 Digital Certificates

Digital certificates, which enable the secure transmission of data over the internet, use asymmetric encryption of data with **Transport Layer Security (TLS)**. They provide data integrity, user identification/authentication, user non-repudiation, data confidentiality and digital signatures.

BorderNet SBC provides self-signed digital-certificate creation or certificates signed by trusted authorities. The self-signed certificates are created using the RSA public key algorithms.

TLS certificates are used by BorderNet SBC web clients and peers to verify the identity to the connected end over HTTPS or SIP TLS.

BorderNet SBC is capable of generating a **CSR (Certificate Signing Request)** to be sent to a desired **Certificate Authority (CA)**. Once the certificate is signed by the CA it can be uploaded to the BorderNet SBC.

→ To create a self-signed certificate:

1. Select **Administration** → **Security Configuration** → **Digital Certificates**.

The **Digital Certificates** window opens.

Name	Type	Subject	Expiry Date	Fingerprint
Test_M_signed	Own	commonName=domain.com.organizationalUnitName=Dev,ou,Jan 18 14:17:41 2020 GMT	05:08:1E:13:12:9D:7F:15:93:F0:34:E4:47:EE:01:3F:3D:42:F3	
Demonstration	CSR	/C=br/CN=Dialogic		Demonstration
test8	CSR	/C=IL/L=Israel/CN=qa.test8		test8
test5	CSR	/C=IL/O=Dialogic/CN=lab.qa5		test5
test1	Own	commonName=lab.qa3.organizationalUnitName=Eng,org,Jan 15 08:36:46 2020 GMT	6D:3D:44:AB:4C:3E:06:3D:29:EB:F3:7C:C4:6F:E4:6E:2D:7F:49:14	
test2	CSR	/C=IL/L=Israel/O=Dialogic/OU=Eng/CN=lab.qa4		test2
root_CA_microsoft	Trusted	commonName=onse-ONCE-OC-CA,displayName=onse,Nov 30 11:43:35 2022 GMT	1A:4E:2C:69:56:31:7C:AC:4E:A7:23:66:93:73:09:D6:D6:93:93:88	

**Note:**

The type of certificate indicates how that certificate is used. The BorderNet SBC uses Own certificates to send to remote entities and Trusted certificates to validate certificates from remote entities.

2. Select **+Add New Certificate**.

The **Add Certificate** window opens.

3. Edit the fields accordingly.

4. Click **Save**.

The new certificate is added.

→ To create a new CSR:

1. Select **Create New CSR**.

The **Create CSR** window opens.

2. Enter the required fields, as illustrated in the following figure.

3. Click **Save**.

→ To upload a certificate:

1. Select **Upload Certificate File**.

The **Upload a Certificate File** window opens.

2. Insert the certificate name and browse to its file.

3. Click **Save**.

The certificate is uploaded.

---

**Note:**

The CSR cannot be edited. If there are errors in the CSR it is necessary to generate a new one.

---

## 6.4 TLS Profiles

BorderNet SBC supports **Transport Layer Security (TLS)** to secure SIP signaling and management traffic. TLS provides data confidentiality, integrity and authentication between the BorderNet SBC and remote end-points or management clients, supporting the following:

- TLS 1.0, TLS 1.2 and SSL 3.0 versions
- Self-signed certificate generation using RSA algorithm for use by the BorderNet SBC
- Importing x.509 certificates of Certificate Authorities and other trusted certificates in PEM/base 64 encoding
- RSA cipher suites - AES128, AES256 and RC4 encryption with SHA or MD5

The TLS profile allows attributes customization to establish a TLS connection with a remote entity, associated with a SIP Interface, when selected as the transport layer.

→ To set a TLS profile:

1. Select **Application** → **Security Configuration** → **TLS Profiles**.

The **Security - Profile Configuration** window opens:



2. Click **+Add New TLS Profile** to open the **Add TLS Profile** window.



3. Enter the following parameters:

- **Name.** The TLS Profile Name.
- **Local Certificate.** Certificate used by the BorderNet SBC to send to remote entities. These are self signed certificates, or CA signed certificates created from a BorderNet CSR.
- **Trusted Certificates.** Certificates used to validate certificates sent by remote entities.
- **Cipher Suites.** Select the cipher suites. RSA\_WITH\_NULL\_SHA, RSA\_WITH\_NULL\_MD5 do not provide encryption and should only be used for debugging.
- **TLS Protocol.** Select the TLS version.

- **Mutual Authentication.** Choose if client authentication is required.
4. Click **Save**.

## 6.5 IPsec Overview

**Internet Protocol Security (IPsec)** provides interoperable, cryptography-based security for IPv4 and IPv6 protocols as follows:

- Data origin authentication
- Connectionless integrity
- Confidentiality (via encryption)
- Access control
- Detection and rejection of replays
- Limited traffic flow confidentiality

IPsec is transparent to other applications, which means that applications can use IPsec without any modifications required.

The BorderNet SBC supports IPsec functionality to secure the signaling and media traffic for **SIP Interconnect** between the BorderNet SBC and remote endpoints.

The following functionality is supported:

- Authentication Headers (AH) and Encapsulation Security Payload (ESP)
- Tunnel and transport modes
- Key management using manual keys and IKE (with pre-shared secrets)
- IPsec for IPv4 and IPv6

IPsec is a licensed feature. The licenses are created in accordance with maximum number of tunnels allowed. Each **IPsec Key Policy** has two tunnels. Contact Dialogic Customer Support for information on licensing.

→ To configure the IPsec:

1. Create an IPsec Key Profile.

The IPsec Key Profile captures parameters that are required to create and negotiate a key for a **Security Association (SA)**. T BorderNet SBC supports:

- **Manual Key.** IPsec Manual Key Profiles capture algorithms and keys required for the IPsec SA.
- **IKE Key.** IPsec IKE Key Profiles capture the parameters required to negotiate both IKE and IPsec SA.

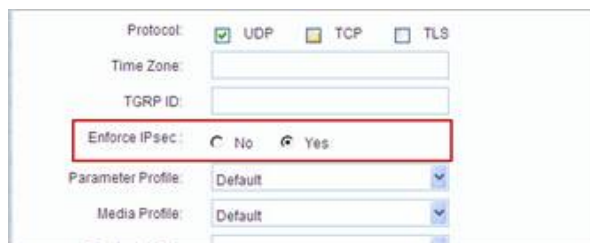
2. Associate the **Profile** to an **IPsec Policy**.

3. Enforce the IPsec.

IPsec is enforced for **SIP Interfaces and Peers**.

4. When creating/editing a SIP interface or peer, set the **Enforce IPsec** to **Yes**.

This ensures that traffic is always IPsec (if no IPsec Policy is configured for the peer/interface, then incoming UDP traffic from this peer/interface will be dropped).



The screenshot shows a configuration form with the following fields and values:

- Protocol:  UDP  TCP  TLS
- Time Zone: [Empty text box]
- TGRP ID: [Empty text box]
- Enforce IPsec:  No  Yes
- Parameter Profile: Default [Dropdown arrow]
- Media Profile: Default [Dropdown arrow]

---

**Note:**

To access the IPsec, System Manager Permission is mandatory.

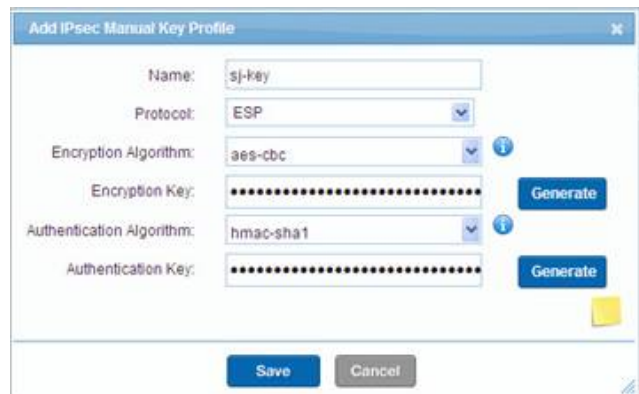
---

## 6.6 IPsec Manual Key Profiles

→ To create an IPsec manual key profile:

1. Select **Application** → **Security Configuration** → **IPsec Manual Key Profiles**.
2. Select the **+Add New IPsec Manual Key Profile** button.

The **Add New IPsec Manual Key Profile** window opens.



3. Enter the following parameters:
  - **Name**. The name of the IPsec manual key profile.
  - **Protocol**. Select the Protocol using the drop-down menu.
  - Possible values: **ESP, AH**.
  - **Encryption Key**. Enter an Encryption Key or click the **Generate** button to automatically create an encryption key.
  - **Authentication Algorithm**. Select the Authentication Algorithm from the drop-down list.
  - **Authentication Key**. Enter an Authentication Key or click the **Generate** button to automatically create an authentication key.
4. Click **Save**.

## 6.7 IPsec IKE Profiles

→ To create an IPsec IKE key profile:

1. Select **Application** → **Security Configuration** → **IPsec IKE Key Profiles**.
2. Select the **+Add New IPsec IKE Profile** button.

The **Add New IPsec IKE Profile** window opens.

3. Enter the following parameters:

- **Name.** The name of the IPsec IKE key profile.
- **IKE version.** Select the IKE version using the drop-down menu.
- Possible values are:
  - oIKEv1
  - oIKEv2
  - oIKEv2 compatible
- **Mode.** Select the **Mode** using the drop-down menu.
- Possible values are:
  - oMain Mode. Uses 3 two-way exchanges to authenticate the IPsec peers during ISAKMP negotiation.
  - oAggressive Mode. Makes fewer exchanges and uses fewer packets than main mode for faster authentication **during ISAKMP negotiation.**
- **Cipher Suites.** Use the **Add** button to move **Available** cipher suites to the **Selected** column.
- If explicit ciphers are selected, then the ISAKMP negotiation with the far end will be limited to the ciphers in the selected column. If no ciphers are selected, then the ISAKMP negotiation with the far end includes all supported ciphers.
- **IKE Lifetime.** Enter the **IKE Lifetime** in minutes.
- Indicate the **Dead Peer Detection (DPD)** parameters.

4. Enter the **DPD Timeout** in seconds.

5. Enter the **DPD Delay** in seconds.

6. Select the **DPD Action** from the drop-down menu. Values are **Hold, Clear, Restart, Restart by Peer.**

- **Encryption Algorithm.** Select the Encryption Algorithm from the drop-down list. This algorithm is used for encryption in the IPsec SA.
- **Authentication Algorithm.** Select the Authentication Algorithm from the drop-down list. This algorithm is used for authentication in the IPsec SA.
- **SA Lifetime.** Enter the SA Lifetime in minutes.

---

**Note:**

The SA Lifetime renegotiates the connection after a timeout.

---

7. Click **Save.**

The **IPsec IKE Key Profile** is added to the **Summary** screen.



## 6.8 IPsec Policies

IPsec Policy is a set of rules that determines how an associated **IPsec Key Profile** operates. Multiple IPsec Policies can be created, and individual policies can be activated or deactivated at any time.

Prior to creating an IPsec Policy, the following requirements should be met:

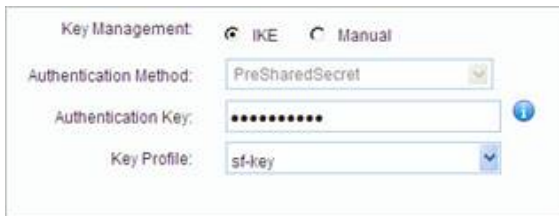
Requirement	Description
General	<ul style="list-style-type: none"> <li>Maximum number of IPsec policies, configured in the system is 25K.</li> <li>The Field Name and the 6-tuple (LocalIPId, Local Mask, Local Port, Remote IP, Remote Mask, Remote Port) must be unique.</li> </ul>
Tunnel Type Policy	<ul style="list-style-type: none"> <li>If two IPsec Policies are of type Tunnel, have identical LocalTunnelIPId and RemoteTunnelIP, then the two policies must have the identical Key Management type (IKE or Manual) and identical Key Profile.</li> <li>If two IPsec Policies are of type Tunnel, have identical LocalTunnelIPId, RemoteTunnelIP, and have the same IKE Key profile, then the two policies must have the identical Authentication Keys.</li> <li>For IPsec Policies that are of type Tunnel and have a Manual Key profile, the 3-tuple (LocalTunnelIPId, RemoteTunnelIP, and SecurityParameterIndex) must be unique.</li> </ul>
Transport Type Policy	<ul style="list-style-type: none"> <li>If two IPsec Policies have Transport Type and have identical LocalIPId and RemoteIP, then the two policies must have the identical Key Management type (IKE or Manual) and identical Key Profile.</li> <li>If two IPsec Policies have Transport Type, have identical LocalIPId and RemoteIP, and have the same IKE Key profile, then the two policies must have the identical Authentication Keys.</li> <li>For IPsec Policies that are of type Transport and have Manual Key profile, the 3-tuple (LocalIPId, RemoteIP, and SecurityParameterIndex) must be unique.</li> </ul>
Remote IPs	<ul style="list-style-type: none"> <li>If an IPsec Policy's RemoteIP comes from a SIP or H.323 Peer, make sure IPAddressType (IPv4, IPv6) and RemoteIP matches either an entry in SourceAddressList or HostAddressType and Host on a SIP or H.323 Peer. This constraint is also checked when a SIP or H.323 Peer is modified or deleted.</li> <li>If an IPsec Policy's RemoteTunnelIP is indicated to come from a SIP or H.323 Peer, make sure IPAddressType (IPv4, IPv6) and RemoteTunnelIP matches either an entry in SourceAddressList or HostAddressType and Host on a SIP or H.323 Peer. This constraint is also checked when a SIP or H.323 Peer is modified or deleted.</li> </ul>
Foreign Keys	<ul style="list-style-type: none"> <li>LocalIPId is a foreign key referring to NpIPCfg.</li> <li>LocalTunnelIPId/FKId is a foreign key referring to NpIPCfg.</li> <li>KeyMgmtId/IKE is a foreign key referring to NpIPSecIKeyProfile.</li> <li>KeyMgmtId/Manual is a foreign key referring to NpIPSecManualKeyProfile.</li> </ul>

Table 10: IPsec Policy Requirements

→ To create an IPsec Policy:

1. Select **Application** → **Security Configuration** → **IPsec Policies**.
2. Select the **+Add IPsec Policy** button.  
The **Add IPsec Policy** window opens.
3. Edit the parameters.

- o **Status.** Enable/disable the policy by selecting **ON** or **OFF**.
- o **Name.** The IPsec Policy unique name.
- o **Mode.** Select the mode.
- o Possible values:
  - o**Tunnel.** Encapsulates the entire packet
  - o**Transport.** Encapsulates only the IP packet's payload
  - o**IP Address Type.** Select IPv4 or IPv6.
- o **Local IP.** Select the local IP (on the BorderNet SBC), using the drop-down menu.
- o **LocalPort.** Enter the local port.
- o **Remote IP.** Enter the remote IP (indicates a peer) - can be selected from the drop-down menu or entered manually.
- o **Remote Mask.** Enter the remote mask.
- o **Remote Port.** Enter the remote port.
- o **Transport Protocol.** Select the transport protocol (Any, UDP or TCP). The Any option will not configure transport protocol for the policy.
- o **Local Tunnel IP.** Select the Local Tunnel IP using the drop-down menu. The Local IP and the Local Tunnel IP must be configured on the same VLAN - for Tunnel mode only.
- o **Remote Tunnel IP.** Select the Remote Tunnel IP using the drop-down menu (any VLAN can be selected) -for Tunnel mode only.
- o § **Key Management.** Select the Key Management type.
- o **IKE.** Select the Authentication Method and enter the Authentication Key.
- o The Authentication Key can be a hexadecimal number or an ASCII string. A hexadecimal number must have a 0x prefix. The ASCII string can have a maximum length of 256 characters. The widely accepted Authentication Methods are Pre-Shared Secret and RSA signatures.
- o Currently only Pre-Shared Secret (Pre-Shared Key) is supported.



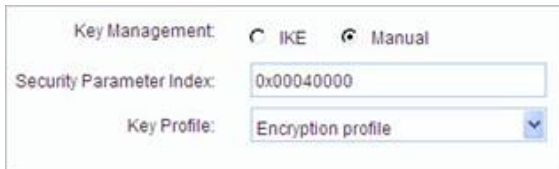
Key Management:  IKE  Manual

Authentication Method: PreSharedSecret

Authentication Key: [Redacted]

Key Profile: sf-key

- **Manual.** Enter the **Security Parameter Index**. The SPI is a hexadecimal number and must have a 0x prefix. The SPI must be identical on both the local and the remote side configured in the IPSec policy.



Key Management:  IKE  Manual

Security Parameter Index: 0x00040000

Key Profile: Encryption profile

- **Key Profile.** From the drop-down menu, select the Key Profile to be associated to the policy.
4. Click **Save**.

# 7. Policy Configuration

## 7.1 Overview

Routing policies are established by applying parameters and global variables to a configured policy to route traffic.

Policy-based routing rules are based on:

- Call parameters, which are derived directly from the message
- Non-call parameters, which are derived from:
  - Service profile time zone attached to the incoming peer or interface
  - Global variables that store intermediate results used in routing decisions
  - Incoming interface and peers

Routing policies are created by first creating the policy, and then adding a rule. The rule is an instruction for the policy that provides directions for routing and returns a route result for where to route a call.

Once a rule has been added to the policy, the routing policy can be further refined by adding siblings before or after the rules, or by adding a child to the rule.

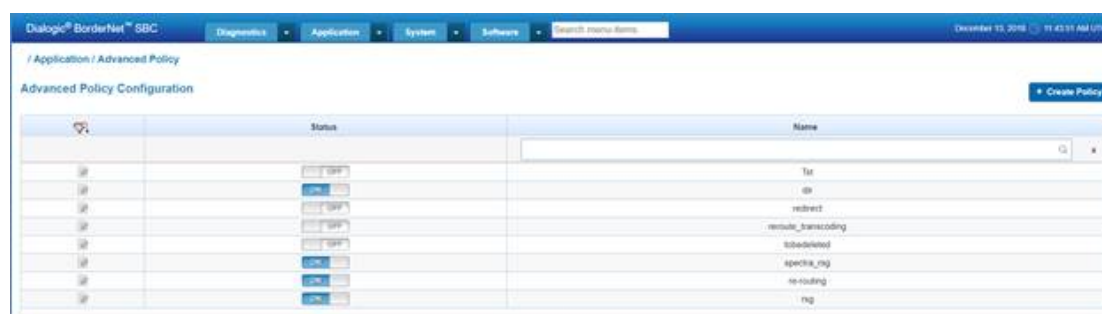
## 7.2 Advanced Policy

→ To create a routing policy:

1. Select **Application** → **Policy Configuration** → **Advanced Policy**.

The **Advanced Policy Configuration** window opens.

This window displays the BorderNet's routing policies list.



2. Click the **Create Policy** button.

The **Add Advanced Policy** window opens.



3. Set the status to **ON** and enter a Name for the routing policy.

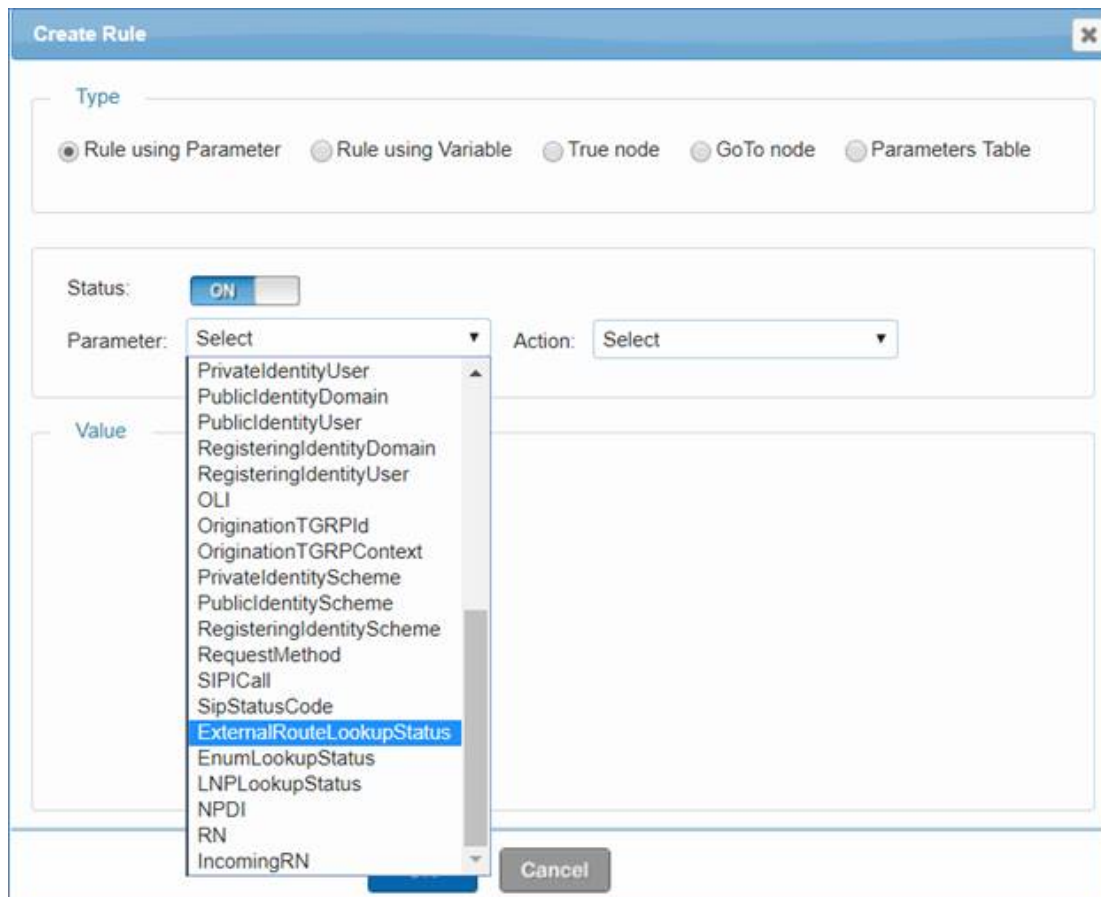
4. Click **Save**.

5. From the **Advanced Policy Configuration** screen, select the **Action List** icon  and click **Edit**.

1. The routing policy is shown in the tree view.



2. Select the **Add Child** icon  to open the **Create Rule** window.



The **Create Rule** window provides options for:






- Rule using Parameter. Establishes a rule based on the BorderNet SBC [Routing Parameters](#). The Parameter and Action drop-down lists are automatically populated with the available options.
- In addition, the values appear in the Parameter drop-down menu can be customized based on SIP or ISUP (for SIP-I) headers and parameters, available in [SIP Profilers](#) and [ISUP Profilers](#) windows. Customization can be applied by selecting the Parameter: GenericParameter (for SIP). After creating a parameter, it shall automatically be added to the drop-down menu and will be selectable to complete the rule.
- Rule using Variable. Establishes a rule using the user-created [Global Variables](#). The Parameter and Action drop-down lists are automatically populated with the available options.
- True node. Establishes that the rule is always true and moves to the next sibling.
- GoTo node. Selects another policy to which to route the call. The Policy drop-down list is automatically populated with the available policies.

3. Select the desired options and click **Save**.

Once the rule has been added, the tree expands and additional options become available.



After the first rule has been added, the routing policy can be configured with the following options:

- Add a sibling after the rule. 
- Add a sibling before the rule. 
- Add a **treatment**. 
- Modify the node. 
- Delete the node. 

## 7.2.1 Routing Parameters

The System Administrator uses routing parameters to create policy rules based on specific criteria.

BorderNet SBC supports the following routing parameters.

Parameter	Action	Token List
CallingPartyUserId CallingPartyDomainName CalledPartyUserId CalledPartyDomainName	BeginsWith EndsWith	StringList
	IsEqualTo IsNotEqualTo	StringList ParameterList VariableList
	AssignsFrom CriteriaBeginsWith CriteriaContains BelongsTo	String Parameter Variable
CallingPartyUserId CalledPartyUserId	Number Translation	Number Translation Profile
CallingPartyURIScheme CalledPartyURIScheme	IsEqualTo IsNotEqualTo	EnumList
CPC	IsEqualTo IsNotEqualTo	EnumList
CurrentTime	MatchesDayOfMonth	Integer between 1-31
	MatchesDayOfWeek	EnumList (Sunday-Friday)
	MatchesDayOfYear	Integer between 1-366
	MatchesMonthOfYear	EnumList (Jan-Dec)
	MatchesWeekOfMonth	Integer between 1-5
	IsInTimeBand IsNotInTimeBand	TimeBandList
DestinationDomain DestinationTGRPID DestinationTGRPContext	BeginsWith EndsWith	StringList
	IsEqualTo IsNotEqualTo	StringList ParameterList VariableList
	Assigns From Contains	Parameter Variable String
Emergency Call	IsPresent	Boolean (YES or NO)

Parameter	Action	Token List
From	AssignsFrom BeginsWith BelongsTo CriteriaBeginsWith CriteriaContains DoesNotBelongTo EndsWith IsEqualTo IsNotEqualTo AddPrefix AddSuffix DeletePrefix DeleteSuffix NumberTranslation	Parameter StringList Variable NumberTranslationProfile ParameterList VariableList
GenericParameter	BeginsWith EndsWith Contains	StringList
	IsEqualTo IsNotEqualTo	StringList ParameterList VariableList
IncomingPeer IncomingInterface	IsEqualTo IsNotEqualTo	UnsignedList ParameterList VariableList
MessageRouting	IsPresent	Boolean (YES or NO)
PrivateIdentityDomain PrivateIdentityUser	BeginsWith EndsWith	StringList
	IsEqualTo IsNotEqualTo	StringList ParameterList VariableList
PublicIdentityDomain PublicIdentityUser	BeginsWith EndsWith	StringList
	IsEqualTo IsNotEqualTo	StringList ParameterList VariableList
RegisteringIdentityDomain RegisteringIdentityUser	BeginsWith EndsWith	StringList
	IsEqualTo IsNotEqualTo	StringList ParameterList VariableList
OLI	BeginsWith EndsWith	StringList
	IsEqualTo IsNotEqualTo	StringList ParameterList VariableList
OriginationTGRPID OriginationTGRPContext	BeginsWith EndsWith	StringList
	IsEqualTo IsNotEqualTo	StringList ParameterList VariableList
	Assigns From Contains	Parameter Variable String
PrivateIdentityScheme PublicIdentityScheme RegisteringIdentityScheme	IsEqualTo IsNotEqualTo	SIP SIPS TEL
RequestMethod	IsEqualTo IsNotEqualTo	INVITE MESSAGE NOTIFY OPTIONS PUBLISH REFER REGISTER SUBSCRIBE
SIPICall	IsPresent	Boolean (YES or NO)
SipStatusCode	IsEqualTo IsNotEqualTo	ENUM (4xx, 5xx, 6xx)
ExternalRouteLookupStatus	IsEqualTo IsNotEqualTo	ENUM
EnumLookupStatus	IsEqualTo IsNotEqualTo	ENUM
LNPLookupStatus	IsEqualTo IsNotEqualTo AssignsFrom	ENUM
NPDI	IsPresent	Boolean (YES or NO)
RN	AssignsFrom BeginsWith Contains EndsWith IsEqualTo IsNotEqualTo	String StringList ParameterList VariableList
IncomingRN	AssignsFrom BeginsWith Contains EndsWith IsEqualTo IsNotEqualTo	String StringList ParameterList VariableList

Table 11: Routing Parameters

## 7.2.2 Treatment

Treatments are applied to routing policies. Treatments always return a route result indicating where to route the call.

There are five types of treatments:

- Route. Identifies the remote point where the call has to be directed. For each treatment priority and weight can be set.
- The following lists the routes types:
  - Interface
  - Peer
  - Interface-Peer
  - Interface-FQDN
  - RegisteredId
- Reject. Rejects the call.
- Apply Rule. Applies a configurable profiler.
- Media:
  - **Intercept. Enable/Disable** media interception per call (see Interception Treatment Configuration).
  - Transcoding options: **Activate Transcoding/ForceTranscodingOff**.
- Flow Class. Applies a white or gray flow.

These treatments can be applied individually or in **Route/Apply Rule** or **Reject/Apply Rule** combinations.

→ To apply a treatment:

1. Set the status to **ON** and enter a **Name** for the routing policy.
2. Click **Save**.

The screenshot shows the 'Create Treatment' dialog box with the following configuration:

- Type:** Route (selected), Reject, Apply Rule, Media, Flow Class.
- Treatment Parameter:** Interface (selected in dropdown).
- Input:**
  - Interface: Access-Interconnect2 (dropdown)
  - Destination TGRP ID: (empty text box)
  - Destination TGRP Context: (empty text box)
  - Priority: 1 (text box)
  - Weight: 50 (text box)
- Display:**
  - Route: interface:intf=Access-Interconnect2,priority=1,weight=50


**Note:**

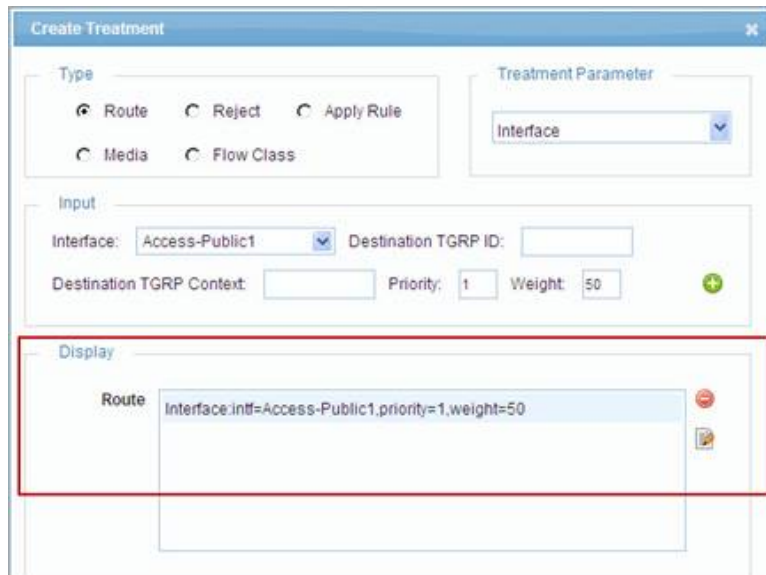
When a Route is selected, the Input selections change based on selected treatment parameters.




In the **Create Treatment** window, the user can set priority and weight for peers and interfaces.

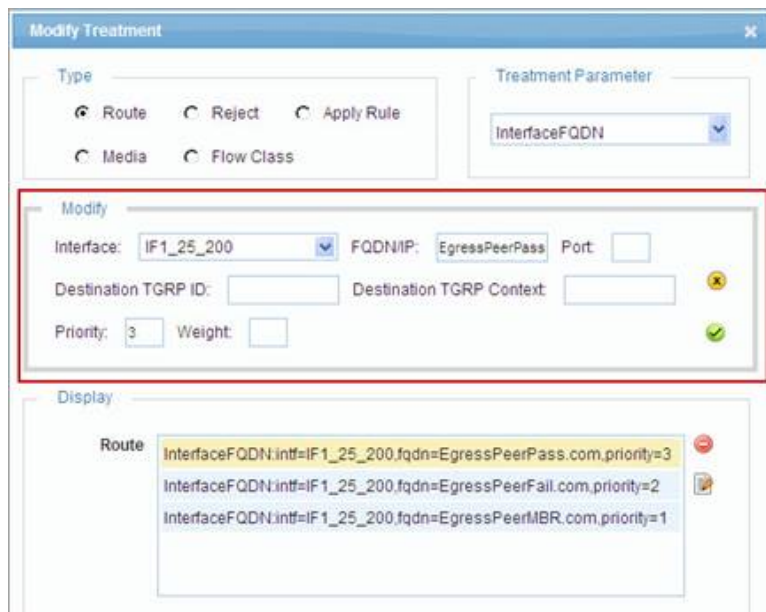
Routes are executed based on the priority and weight, and the highest priority route is tried first. If the priority is the same for multiple routes, the distribution is based on the weighted average of the routes.



1. In the **Input** section, enter the **Priority** and **Weight** [1 - 100],
2. Click the green plus sign  to add the **Input** to the **Display** section of the window.



→ To modify the treatment parameters:

1. Double-click the **Treatment** to open the **Modify Treatment** window.
2. Select the desired **Route** and click the **Modify**  icon.

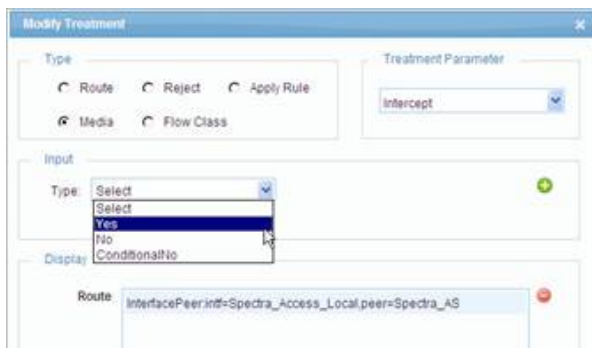


3. Set the parameters and edit the **Priority** and **Weight**.
4. Click the check mark  to save the changes or click the "x"  to cancel your changes.
5. Click **OK** to save the changes.

### 7.2.2.1 Interception Treatment Configuration

→ To enable media interception per call:

1. In the **Create Treatment** window, select **Media** as the **Type**.
2. Select Intercept as the Treatment Parameter,
3. Choose one of the following values from the **Input** drop-down menu:
  - **Yes**. Media is intercepted (to be selected in this case).
  - **No**. Media is not intercepted.
  - **Conditional No**. The interception depends on the media [transparency](#) selection.



4. Click **OK**.

---

#### Note:

The Intercept Treatment can exist independently or be part of the route. When sessions are processed and Advanced Policy is selected, the Intercept parameter alone can be a treatment, or the Intercept parameter can go with the routes identified in the treatment.

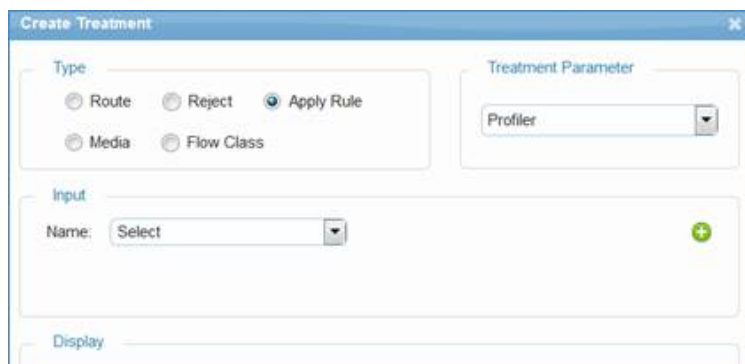
---

### 7.2.2.2 Enterprise Routing

BorderNet SBC's Enterprise Routing achieves the maximum manipulation of digits, especially in the **Called Party Identity** (the user part of the Request-URI). This is accomplished via the SIP Profiler, which can be applied as part of the treatment parameter in the Advanced Policy.

→ To enable enterprise routing:

1. In the **Create Treatment** window, select **Apply Rule** as the **Type**.
2. Select **Profiler** as the Treatment Parameter.
3. Select the desired profiler using the **Input** drop-down menu.



4. Click **OK**.

The BorderNet SBC leverages codec filtering offered in the Media Profile for pruning out the codecs prior to sending out to the destination. For destination-based codec manipulations, the SDP Profiler (which is part of the SIP Profiler) can be associated as part of the treatment in advanced policies.

---

**Note:**

A single profiler introduced as part of the treatment in an Advanced Policy can manipulate digits in SIP headers as well as the codecs in the SDP of the SIP message.

---

See the [SIP Profiler](#) section for more information.

## 7.3 Rerouting

Rerouting a session means trying a new destination when the session initialization attempt towards an existing server fails. Before v3.8.0, BorderNet retries all existing Egress servers when there is a failure, except for 486. To enable operators to control this behavior and provide alternate actions like REJECT, SKIP\_CARRIER, REDIRECT or CONTINUE, rerouting is used.

The following session rerouting options are available:

Treatment	Functionality
Reject	Rejects the call and stops attempting further routes. An optional cause value can also be set
SKIP Carrier	SKIPS the carrier associated to the current Egress point and jumps to the next
Continue Next Route	Continues to attempt the next route identified earlier
Redirect	Ability to drop existing routes and reanalyze new routes
LNP	Dips to an External server for LNP
External Routes	Dips to an External server specified for routing destinations
Try with Transcoding	Triggers a reattempt to the existing route with transcoding
ENUM Lookup	Dips to an External ENUM Server for TEL>SIP URI translations

### 7.3.1 External Route Server (SIP Redirect Server)

Interconnection to an **External Route Server** is available. With this feature, operators can configure the BorderNet SBC to consult an external routing engine via the SIP INV/3xx method to receive call routing instructions in the form of route lists.

The **ExternalRoutes** treatment provides information about routing the call towards the External Route Server, which itself provides the routes. These External Routing servers are SIP-based and in response to a request on INVITE provide routes in the form of a 302 response.

Additional features available include the following, which are applicable for the entire release and not just for the specific External Route Server:

- Ability to control rerouting based on Cause Codes.
- Ability to lookup into the External Route Server for routes/destinations.
- Ability to identify a group of peers as carriers.
- Ability to skip peers for a given carrier.
- Ability to lookup into the external server for Local Number Portability (LNP).
- Ability to lookup into an in-switch LNP.
- Ability to define routing templates for a large data of rules (Matrix Feature).
- Policy control to reattempt a destination with transcoding.

- A failback mechanism for external route server failures.

To support this feature, the BorderNet SBC WebUI enables the modification of SIP Profiler entries and parameters to provide access and route traffic to the External Route Server.

The BorderNet SBC also supports routing using trunk group parameters as part of this feature.

This indicates the action which needs to be taken after an **ExternalRoute** lookup can be configured by the operator based on the **ExternalRouteLookupStatus** field.

The possible values of the **ENUMLookupStatus** are:

- DipNotDone
- Failure
- NoRoutes
- RoutesAvailable

The **External Routing** process is illustrated in the following diagram.

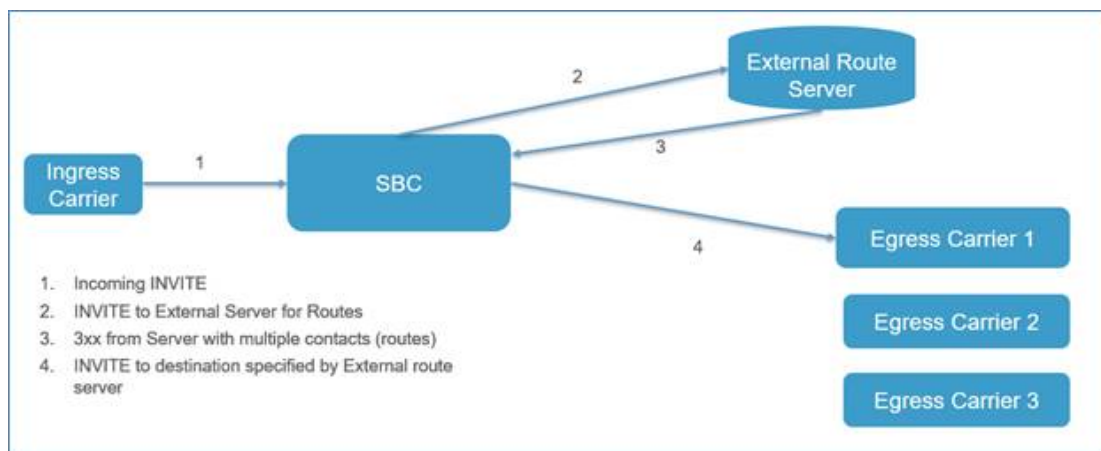


Figure 9: External Routing Process

## 7.3.2 Local Number Portability (LNP)

**Local Number Portability (LNP)** is a service that allows subscribers to switch local or wireless carriers and still retain the same telephone number.

BorderNet performs external lookups for LNP and one or more peers can be configured as LNP servers. If one server times out them the lookup is referred to another server. When all external servers are exhausted, the system will lookup into the advanced policy with the parameter **LNP lookup failed**.

In the case of a 302 response where the LNP Dip is performed and the number is not translated, after a response from the LNP Server (even on timeout), an additional advanced policy is performed. Based on the value of the **LNPLookupStatus** parameter value, the operator can decide on any action.

The possible values of the **LNPLookupStatus** parameter are:

- Dip Done Not translated
- DipNotDone
- Failure
- DipDoneTranslated

Where the LNP lookup leads to a new translated number, then an advanced policy lookup is performed to 're-analyze' the new data and any treatment if present, will be discarded.

The **LNP Lookup** process is illustrated in the following diagram.

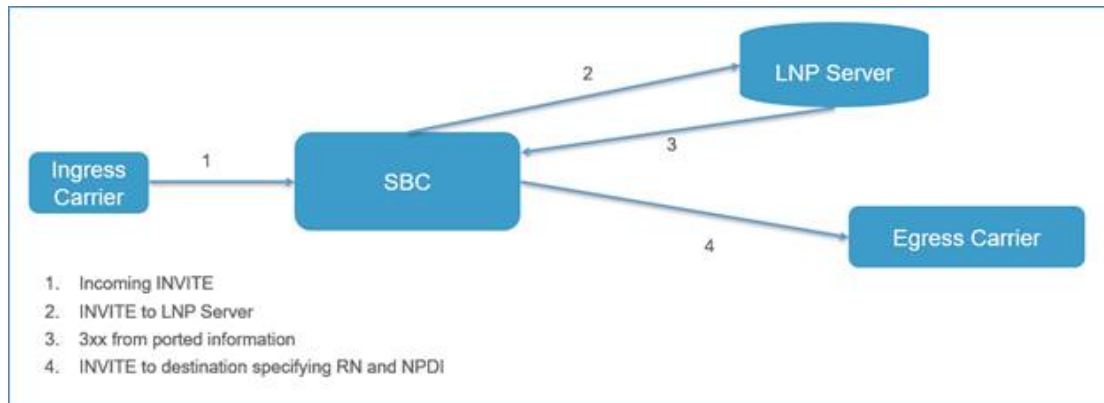


Figure 10: LNP Lookup Process

## 7.3.3 Matrix

All advanced policies define rule parameters and data for making policy decisions. With the introduction of features like **Number Translation**, **Criteria Lookup** and **Directory Lookup**, in BorderNet large data can be bulk-loaded and data kept in isolation to the policy rules, making it quick for access. Criteria Lookup allows rule parameters to have data outside of the policy.

The **Matrix** feature of the Control Switch allows multiple criteria fields defined in the policy to use bulk data configured separately. The Criteria Lookup feature allows the same, but is limited to only one field.

All rule parameters that contain 'Criteria belongs to' and 'Criteria doesn't belong to' actions support the 'Lookup into Matrix' parameter. The Matrix lookup is a Rule type, whose values are in the configured Matrix table. The treatment could be any of the possible values, thereby leaving the Matrix lookup for extracting data, rather than being limited to providing routes. Matrix allows the creation of a template of rules, where the data can be separate from the rules.

## 7.3.4 ENUM

BorderNet supports DNS functionality, and is also able to parse NAPTR and SRV records. This functionality has been enhanced to support **ENUM** routing and ENUM LNP functionality. The user is able to choose by configuration to apply either a SIP LNP or an ENUM LNP. An ENUM server is actually a DNS server, holding NAPTR records with E.164 to URI mappings. When LNP information is queried from the ENUM LNP server, an optional 'rn' parameter can be added in order to indicate the desired routing for the ported number.

When sending an ENUM query to a configured ENUM server, BorderNet uses an NAPTR record type as the record requested. When an NAPTR response is received from the ENUM DNS server, BorderNet verifies that it contains the proper service parameters for ENUM, namely either 'E2U+SIP', 'E2U+pstn:tel' or 'E2U+pstn:sip'.

If no service is present in the answer, or the service is different than the above types, BorderNet shall lookup the advanced policy by setting the LNP lookup failed parameter. The action which needs to be taken after an ENUM lookup can be configured by the operator based on the **ENUMLookupStatus** field.

The possible values of the **ENUMLookupStatus** are:

- DipNotDone
- Failure
- NoRoutes
- RoutesAvailable

The **ENUM Lookup** process is illustrated in the following diagram.

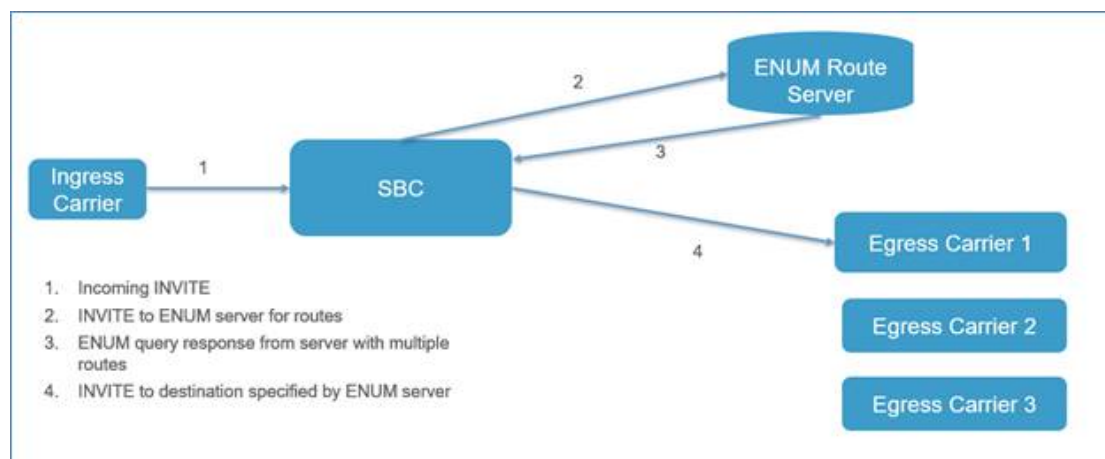


Figure 11: ENUM Lookup Process

## 7.4 Number Translation

Number translation is a service enabling BorderNet SBC to replace the calling/called party number of the call with another number, before the call is sent out to the next hop. The service uses a translation table which is stored in the memory or database.

Number Translation can also be used as a form of **LNP (Local Number Portability)**, such as from Freefone 1800/0800 numbers to real numbers.

The service is part of the **Advanced Policy** routing mechanism applied as an action type which is available for either the called or calling number parameters. The number translation profile depends on the order of rules in the **Advanced Policy**.

The Advanced Policy can then be assigned to a service profile which is then assigned to a SIP interface or peer.

In order to provision a number translation action using the Advanced Policy, the following steps should be performed:

1. Add a rule in an existing or a new Advanced Policy.
2. For a rule type, choose the **Rule using Parameter** option (this is also the default option).
3. In the Parameter drop-down list, select either CalledPartyUserId, CallingPartyUserId or FROM.
4. In the **Action** drop-down list, select **Number Translation**
5. For the **Value** drop-down list, select the desired number translation profile as configured beforehand in **Application** → **Common** → **Translation Profile**.

See the **Translation Profile** section for more information.

The screenshot shows a 'Create Rule' dialog box with the following configuration:

- Type:** Rule using Parameter (selected), Rule using Variable, True node, GoTo node
- Status:** ON
- Parameter:** CalledPartyUserId
- Action:** NumberTranslation
- Value:** To\_Carrier\_A

The maximum number of characters allowed per field is 50. Empty fields are not allowed.

Permitted characters are shown in the table below.

<b>Alphanumeric:</b>	A-Z, a-z, 0-9
<b>Mark:</b>	"_", "_", ".", "!", "~", "*", ""
<b>User unreserved:</b>	"&", "=", "+", "\$", ",", ";", "?", "/"

Each table is stored in a database file for backup and a copy is stored in the RAM for runtime processing.

The string used for the translation is the URI part, both **SIP type URI** and **Tel type URI**.

The "user part" of the **Request-URI** is the key for searching for a match on the translation table:

- If there is a match the Request-URI and the To headers are modified accordingly.
- If there is no match the Number Translation will not do anything and the original number will be used for the routing decision.

After the routing decision has been made BorderNet SBC won't make any extra number translations. The translation will be executed only on the Ingress side.

## 7.4.1 Number Translation on High Availability (HA) Deployments

Number translation tables, both those stored on the database file and the in-memory tables are transparently mirrored between active and standby systems. If a failover occurs all the new calls will then be served by the standby system which has become the active system.

## 7.4.2 Number Translation and the Session Description Record (SDR)

The table below shows the fields which relate to the number translation on the SDR.

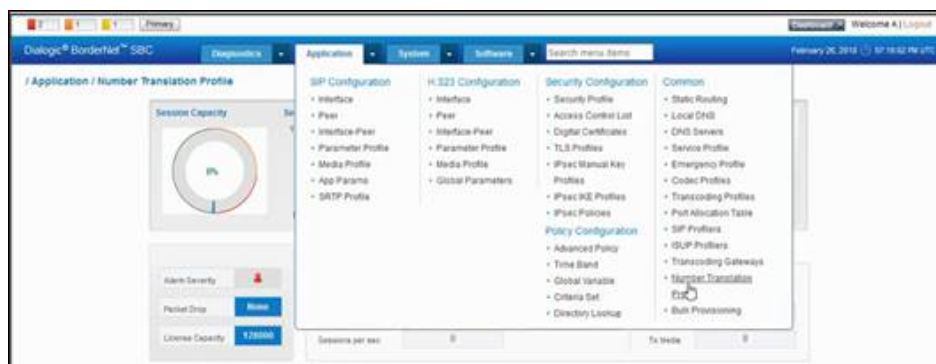
Field Name	Content Description
OutSigReqLine	Translated number in the URI user part
OutSigTo	Translated number in the URI user part
CalledPartyUser	translated number in the URI user part Note: This field contains only the user part of the URI and not the full URI
OrigCalledPartyUser	<ul style="list-style-type: none"> <li>Field population <b>will not</b> be modified</li> <li>This field contains only the user part of the URI and not the full URI</li> </ul> Note: This is the user part of the URI as received on the ingress peer before applying the profiler. Therefore, if a Profiler is used, then this is the pre-profiler number, and if Profiler is not used then this is the number entering the number translation mechanism

→ To access the Number Translation profile:

1. Open up the **Application** tab.
2. Select the **Common** pane within this tab.
3. Scroll down to **Number Translation Profile** in this pane.
4. Click **Number Translation Profile**.

The **Number Translation Profile** window opens.

5. Various actions can be performed within this window such as import of NT tables (in CSV format), export of NT tables, editing of tables, deleting tables, searching for numbers, adding translation sets etc.



### Note:

When choosing CallingPartyUserId, the modified FROM header number is overriden with the P-Asserted-ID value, if such exists. When choosing FROM, the modified FROM header is not dependent on the P-Asserted-ID, so the translated number is present in the FROM header in any case.

Each database should be limited to:

- max 200 tables per database.
- max 100,000 records per table.
- max 4,000,000 records per database.

## 7.5 Directory Lookup



**Directory Lookup** is a service enabling the parameters of the call to be matched/looked up with data in a directory list to determine if the call parameter belongs to that group.

This lookup is performed as part of the policy execution. It is extremely useful in the implementation of logic for functions such as subscriber checking, screening, whitelist, black list etc. These functions often require lookup of call parameters in a directory list.

The policy lookup provides tools for the operator to look up any call parameters into the directory list data, that would enable operators to reject the call, reject calls with announcements/tones, pick alternative routes, or in general for any policy decision.

The requirements for **Directory Lookup** are as follows:

- Every directory list shall be identified by a unique name.
- Each directory list/table shall allow data that shall be stored in a persistent memory (regular XML configuration files or an internal database), and shall also be loaded to the BorderNet's memory (RAM) for runtime processing.
- Each Directory Lookup table shall be limited to a maximum of 100,000 records.
- The total number of lookup tables shall be limited to 100.
- Two routable parameters, BelongsToList and NotBelongsToList allow the policy to check if a given number belongs to a given directory list. The directory list is evaluated by checking to see if the given routable parameters are contained in the directory.



Each database should be limited to:

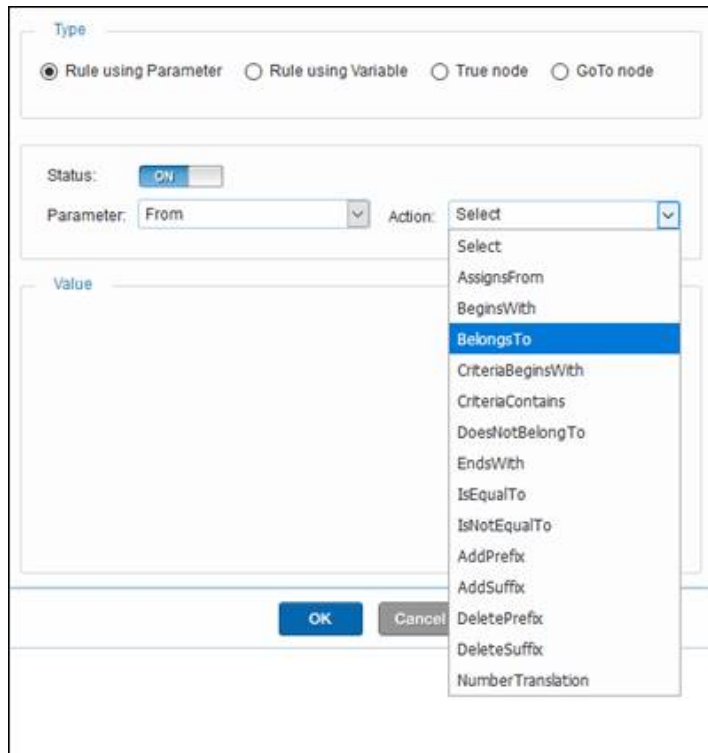
- max 200 tables per database.
- max 100,000 records per table.
- max 4,000,000 records per database.

## 7.6 Criteria Sets

A **Criteria Set** is a list of criteria prefixes along with the numbers that need to be checked against that criteria:

- Each Criteria Set has a unique name.
- Criteria Sets can be added/modified/deleted.
- Each Criteria Set can be defined as a list of numbers similar to the Directory Lookup list of numbers.
- Each Criteria Set can be used as a lookup table for routing criteria.
- The Criteria Set can be used with any operator for lookup with any call parameter. For example, the call parameters could be CalledPartyNumber and the operator could be beginswith or contains which would point to the Criteria Set.
- The Criteria Set enables reuse of rule data multiple times within or across multiple plans.





Each database should be limited to:

- max 200 tables per database.
- max 100,000 records per table.
- max 4,000,000 records per database.

## 7.7 Time Band

Time-based policy rules facilitate traffic during specific times on the network, such as during a peak busy period or to direct calls to a less-expensive route.

Time bands provide configurable start and end times, which can be customized to the year, month, week, day and minute. The System Administrator can create multiple time bands, and time bands can overlap or duplicate other time bands.

→ To create a new time band:

1. Select **Application** → **Policy Configuration** → **Time Band**.

The **Time Band Configuration** window opens, which provides a list of configured time bands.



2. Select the **+Add New Timeband** button.

3. Enter the desired information for the time band, as follows:

- **Status.** By default, the status is set to **OFF**.
- **Name.** The name of the time band.
- **Month of the Year.** Select the start and end months from the drop-down lists.
- **Week of the Month.** Select the start and end weeks for the selected months [1 - 5].
- **Day of the Month.** Select the start and end days for the selected months [1 - 31].
- **Day of the Week.** Select the start and end day of the week from the drop-down lists.
- **Hour of the Day.** Select the start and end hours of the day [0 - 23].
- **Minute of the Hour.** Select the start and end minutes of the hours [0 - 59].

4. Click **Save**.

→ To apply a time band to a policy:

1. Once a time band has been created, it is added to the **CurrentTime** parameter drop-down menu. A rule must be added to apply a time band to a policy.
2. From the **Create Rule** window:
  - Select **Rule** node using **Parameter** as the **Type**.
  - Select **Current Time** from the **Parameter** drop-down menu.
  - Select the appropriate operation from the **Action** drop-down list.
  - Use the **Add >** button to move the desired time band from **Available** to **Selected** windows:

1. Click **OK**.

The incoming call arriving on the specified time is routed appropriately.

→ To delete a time band:

1. Remove all policy references to the time band before deleting the time band.  
Only time bands that are not used in a policy can be deleted.
2. In the **Time Band Configuration** window, click the **Action List** icon next to the selected time band and choose **Delete**.
3. Click **Confirm** to delete the time band.

## 7.8 Global Variables

Global variables are used in policies to store information that is used for route determination. When a global variable is created, it is only a container. The value is assigned to the global variable when the global variable is assigned to a policy.

→ To add a new global variable:

1. Select **Application** → **Policy Configuration** → **Global Variable**.

The **Global Variable Configuration** window opens.

Global Variable - Policy Configuration		+ Add New Global Variable	
	Name		Data Type
	<input type="text"/>		<input type="text"/>
	float-test		Float
	unsigned-test		Unsigned
	int_gv		Integer
	test		String

1. Select the **+Add New Global Variable** button.

The screenshot shows a dialog box titled "Add Global Variable". It has two input fields: "Name" with the value "ny/c\_gv" and "Data Type" with a dropdown menu set to "Float". At the bottom, there are two buttons: "Save" and "Cancel".

2. Enter the **Name** of the global variable.
3. Select the **Data Type** from the drop-down menu.  
Possible values are: **Float, Integer, String, Unsigned**.
4. Click **Save**.

→ To assigning a global variable to a policy

1. After creating the global variable, it is automatically added to the **Variable** drop-down menu.  
A rule must be added or modified to apply a global variable to a policy.
2. In the **Create Rule** window select **Rule** node using **Variable** as the **Type**.
3. Select the desired variable using the **Variable** drop-down menu.

The screenshot shows a dialog box titled "Create Rule". It has several sections:
 

- Type:** Four radio buttons: "Rule node using Parameter", "Rule node using Variable" (selected), "True node", and "GoTo node".
- Status:** A toggle switch set to "ON".
- Variable:** A dropdown menu set to "dusk\_redirect".
- Action:** A dropdown menu set to "AssignsFrom".
- Value:** A section with "Value Type" dropdown set to "Parameter" and "Value" dropdown set to "IncomingPeer".

 At the bottom, there are "OK" and "Cancel" buttons.

4. Select the appropriate operation using the **Action** drop-down menu.
5. Assign the appropriate **Values**.
6. Click **OK**.

→ To delete a global variable:

1. Remove all policy references to the global variable before deleting the time band.  
Only global variables that are not used in a policy can be deleted.
2. In the **Global Variable Configuration** window, click the **Action List** icon next to the selected time band and choose **Delete**.
3. Click **Confirm** to delete the global variable.

## 8. Common Features

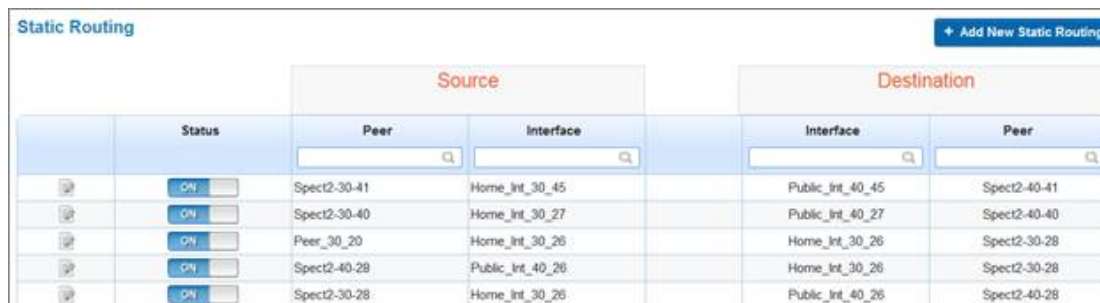
### 8.1 Static Routing

Static routing provides a set route for messages, based on pre-configured routing tables that are set up by the System Administrator.

→ To manually add static routes to the routing table:

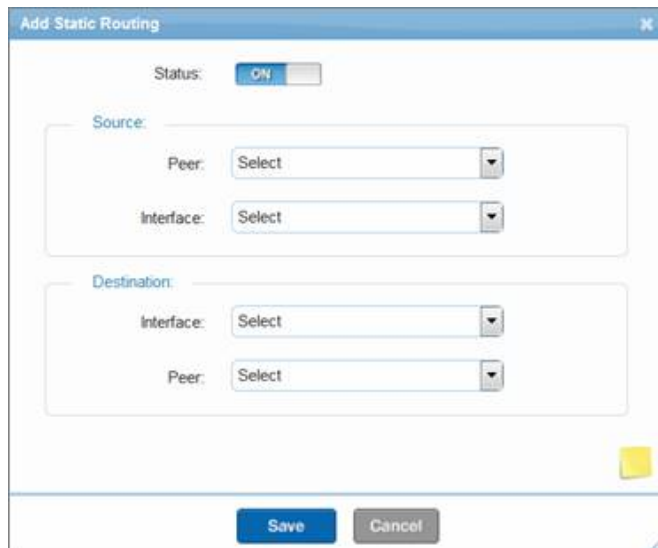
1. Select **Application** → **Common** → **Static Routing**.

The **Static Routing** window opens.



	Status	Source		Destination	
		Peer	Interface	Interface	Peer
	<input checked="" type="checkbox"/>	Spect2-30-41	Home_int_30_45	Public_int_40_45	Spect2-40-41
	<input checked="" type="checkbox"/>	Spect2-30-40	Home_int_30_27	Public_int_40_27	Spect2-40-40
	<input checked="" type="checkbox"/>	Peer_30_20	Home_int_30_26	Home_int_30_26	Spect2-30-28
	<input checked="" type="checkbox"/>	Spect2-40-28	Public_int_40_26	Home_int_30_26	Spect2-30-28
	<input checked="" type="checkbox"/>	Spect2-30-28	Home_int_30_26	Public_int_40_26	Spect2-40-28

2. Click **+Add New Static Routing** to open the **Add Static Routing** window.



**Add Static Routing**

Status:  ON

Source:

Peer: Select

Interface: Select

Destination:

Interface: Select

Peer: Select

Save Cancel

3. Add the following parameters:
  - Status. Enable/disable the static route, by selecting **ON** or **OFF** (default).
  - Select the peers and interfaces for the **Source** and **Destination** of the static route, using the drop-down menus.
4. Click **Save**.

Example: Routing an H.323 Interface-Peer:

→ To establish a route for an H.323 Interface-Peer association:

1. Select **Application** → **Common** → **Static Routing** to open the **Static Routing** window.
2. Click +Add New Static Routing.
3. Set the **Static Routing Status** to **ON**.
4. From the **Source** box, select the **H.323 Peer** and **Interface**, as the peer-interface association that sends traffic.
5. From the **Destination** box, select the **H.323 Interface** and **Peer**, as the interface-peer association that receives traffic.
6. Click **Save**.

Example: Routing a SIP Interface-Peer

SIP routes are included in the incoming SIP message via the **Route** header or **SIP Request-URI**.

SIP routing can be configured to use pre-determined routes in the SIP message. When this configuration is enabled, SIP routing overrides the destination given from a **Point-to-Point (P2P)** routing lookup.

---

**Note:**

When SIP message routing is selected from a Service Profile, BorderNet SBC performs only SIP message routing. Static routing is necessary to determine an outgoing interface for the session and could return a destination Peer, but in this case, SIP message routing determines the destination by the contents of the SIP message.

---

→ To establish a route for a SIP Interface-Peer association:

1. Select **Application** → **Common** → **Static Routing** to open the **Static Routing** window.
2. Click +Add New Static Routing.
3. Set the **Static Routing Status** to **ON**.
4. From the **Source** box, select the **SIP Peer** and **Interface**, as the peer-interface association that sends traffic.
5. From the **Destination** box, select the **SIP Interface** and **Peer**, as the interface-peer association that receives traffic.
6. Click **Save**.

## 8.2 Local DNS

BorderNet SBC supports **Local DNS (Domain Naming Service)**.

→ To establish a local DNS:

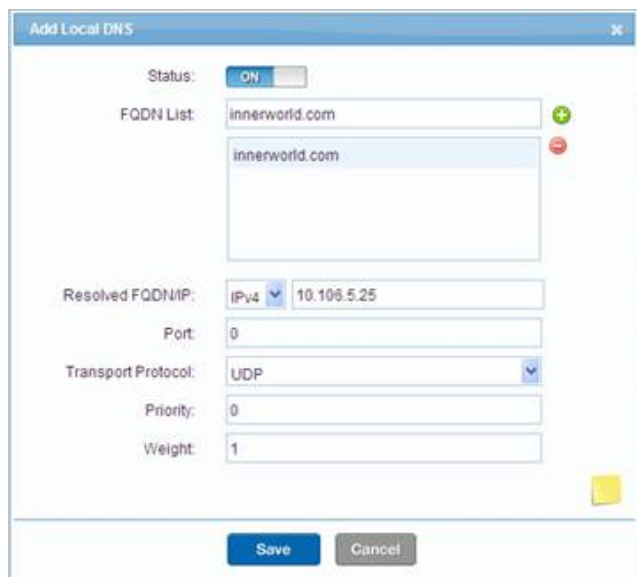
1. Select **Application** → **Common** → **Local DNS**.

The **Local DNS Configuration Summary** window opens.



	Status	FQDN	Resolved IP	Resolved FQDN/IP	Port	Transport Protocol	Priority	Weight
	OFF	aaaa	IPv4	1.1.1.1	0	UDP	0	99
	ON	efgh abcd efgh	IPv4	10.106.5.25	0	None	0	1
	ON	aaaa	IPv4	2.2.2.2	0	None	0	1

1. Click **+Add New Local DNS** to open the **Add Local DNS** window.



2. Enter the following parameters:
  - **Status**. Enable/disable the local DNS, by selecting **ON** or **OFF** (default)
  - Select the **Resolved FQDN/IP** type from the drop-down list (**IPv4**, **IPv6**, **FQDN**).
  - Enter the appropriate **IP address** and **Port**.
  - Select the **Transport Protocol** from the drop-down list (**None**, **UDP**, **TCP**, **TLS**).
  - Enter the desired **Priority** and **Weight**.
3. Click **Save** to add the local DNS.

## 8.3 DNS Servers

The external DNS server is used to resolve the SIP URI into an IP address, port and transport protocol for the next hop to the contact.

- Up to four servers can be configured as DNS servers: one primary and three secondary servers.
- The Primary external DNS server is queried first.
- If the Primary external DNS server is unreachable, secondary DNS servers are tried.

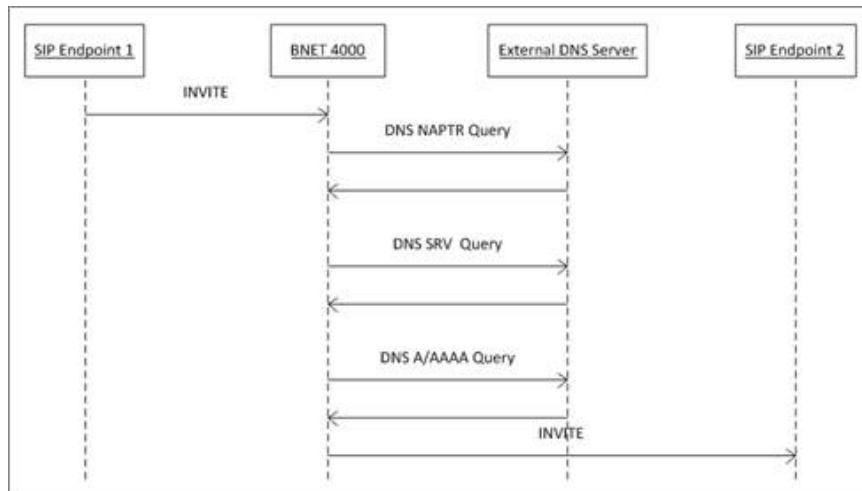
The BorderNet SBC first tries to resolve a domain name with the [Local DNS](#). If there is no resolution, the external DNS server is queried.

If a peer has FQDN to be resolved using the external DNS server and has the connectivity check enabled, BorderNet SBC performs external FQDN resolution and sends a **SIP OPTIONS** ping to the resolved addresses.

The following shows the workflow. BorderNet SBC:

1. Receives an INVITE and determines that the next hop's FQDN does not exist in the local DNS (FQDN can be part of the message, in message-based or message hop routing, or returned by the policy).
2. Communicates with the external DNS Server to find the resolution (multiple DNS queries can be performed).
3. Performs an NAPTR DNS query to get the transport of the next hop that matches its local outgoing interface transport, and SRV DNS query to get the port.
4. Performs the A or AAAA DNS query to get the IP address of the next hop.
5. Forwards the INVITE message to the next hop.





→ To configure an external DNS server:

1. Select **Application** → **Common** → **DNS Servers**.  
The **DNS Servers Configuration** window opens.

### DNS Servers Configuration

**Status:**  ON  OFF

**Name:**

**IP Address Type:**  ▼

**Primary DNS Server:**


**Alternate DNS Server(s):**  +

**Local DNS Interface:**  ▼

2. Enter the following parameters:
  - **Status.** Enable/disable the DNS Servers, by selecting **ON** or **OFF** (default).
  - **Name.** The name of the DNS server.
  - **IP Address Type.** Select IPv4 or IPv6.

#### Note:

- If IPv4 is selected, all subsequent IP addresses must be entered in IPv4 format.
- If IPv6 is selected, all subsequent IP addresses must be entered in IPv6 format.

- **Primary DNS Server.** Enter the primary DNS server IP address and port.
- **Secondary DNS Server.** Enter the secondary DNS server IP address and port.
- Click the green plus  icon to add the information to the field below. Up to three Secondary DNS Servers can be added to this field.

- **Local DNS Interface.** Select from the drop-down menu. This list is automatically populated with the configured local interfaces.
3. Click **Save**.

## 8.4 Service Profiles

A service profile is applied to interface or peer, when the peer service profile is applied first.

Service profiles define the behavior of the BorderNet SBC, by providing operation such as connection between the incoming and outgoing SIP profiles, routing methods and various thresholds definitions. When a service profile is applied, session attempts beyond the configured value are rejected, and an alarm is raised if the session attempts are 10% greater than the threshold the service profile allows.

The EMS manages multiple BorderNets. With the EMS in place, the configurations are implemented on BorderNet through EMS only.

Configurations such as Media Profile and Service Profile are created and pushed to all the managed BorderNets, so that the configuration of a specific BorderNet on EMS will always be in sync with the configuration on the specific BorderNet.

The exact same Media Profiles are configured on all BorderNets, but one of the parameters of the Media Profile is Port Allocation. This parameter uses a VLAN name which is specific to the BorderNet.

Similarly, in Service Profiles, there are other parameters such as Advanced Policy and Sip-Rec Peer, which refer to certain interfaces/peers of the specific BorderNet. These BorderNet-specific configurations prevent generalizing the Media Profile and Service Profile configurations at the EMS level. The way to deal with this problem is:

- to remove the Port Allocation configuration from the Media Profile setting
- remove Advance Policy and Sip-Rec Peer configurations from the Service Profile setting
- re-add these configurations at the Peer & Interface level.

→ To configure a service profile:

1. Select **Application → Common → Service Profile**.

The **Service - Profile Configuration** window opens.

Name	Network Type	Incoming Message Profiler	Outgoing Message Profiler	Emergency Profile	SIP Message Routing	Max Routing Reattempts	Redirect Response Handling
Default	Interconnect	SystemInMsgProfiler	SystemOutMsgProfiler		No	5	Forward

2. Click **+Add New Service Profile** to open the **Add Service Profile** window.

General tab

- Name. The service profile's name.
- Network Type. Select the network type using the drop-down menu:
  - **Interconnect.** Indicates a public network.
  - **Local.** Indicates a private network.
  - **Access-Public.** Indicates a public network towards UEs.
  - **Access-Local.** Indicates home access network.
  - **Access-Interconnect.** Indicates visiting access network.

- **SipRec.** Indicates SIP recording.
- Network Property. Check the Subscriber Traffic box to enable communication with User Equipment.
- Select the Incoming and Outgoing SIP profilers.
- Select the Emergency Profile.

Routing tab



- SIP Message Routing. To enable the SIP message routing, select Yes.
- Max Routing Reattempts.
- Redirect Response Handling. Possible values are Forward or Redirect.
- TGRP Mapping. Select Yes to enable the TGRP mapping (No is the default value).
- Destination Path. Select the Destination Path using the drop-down menu and enter the IP address and Port.

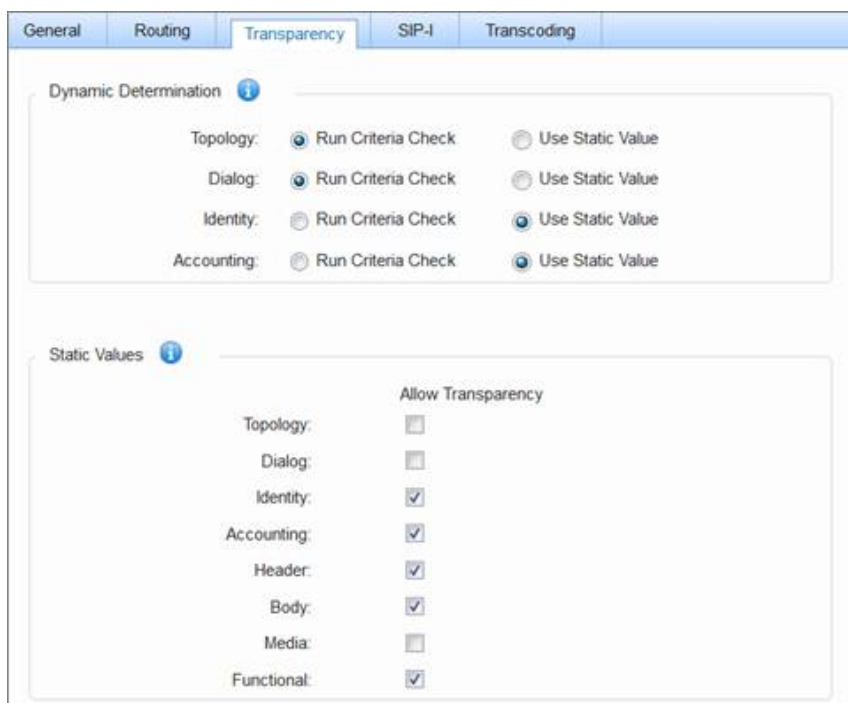
Transparency tab

BorderNet SBC provides transparent interworking with the Dialogic® ControlSwitch™ System and other core switches as well as peers. Two independent SIP dialogs are managed per session, one on each side of the BorderNet SBC. This model provides a controlled transparency, enabling the operator to selectively choose information for transparency, as described below.

Type	Information	Transparency
Topology	Via, From, Contact, P-Charging-Vector	<ul style="list-style-type: none"> <li>• VIA header information is passed through.</li> <li>• Contact header information is sent in another private header.</li> <li>• From header host information is passed through.</li> <li>• P-Charging-Vector is passed transparently.</li> </ul>
Identity	From, Contact, P-Asserted	User information is passed through, but headers carrying user identity may be dropped.
Dialog	Call-ID	The Call-ID alone can be preserved.
Accounting	P-Charging-Vector	Accounting and billing information headers may be dropped.
Header	Unknown Headers	When there are headers that have not been dropped for other reasons, based on header transparency, these unknown headers may be passed through or dropped.
Body	Content Types, such as ISUP, QSIG, Simple-Message-Summary	SDP is not used here, but these specific bodies can be dropped or allowed.

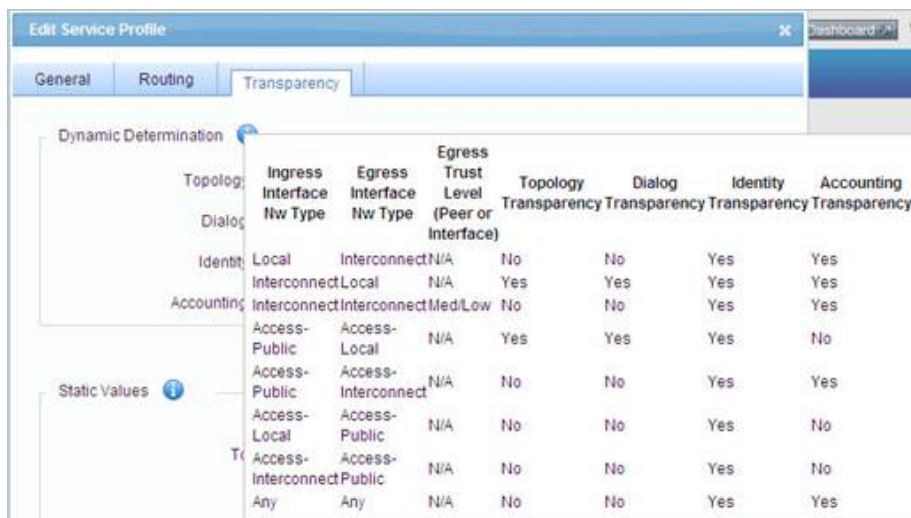
Type	Information	Transparency
Media	SDP	If unchecked, media is intercepted (takes precedence). If checked, media is not intercepted. Applicable if the Media Intercept parameter in SIP/H.323 configuration is set to Flexible.
Functional	SIP Option Tags	Provides feature transparency between two endpoints of the BorderNet SBC. SIP Option Tags take the form of certain headers and parameters. Typically, this transparency is desirable, but some deployments may not encourage this kind of transparency.

Table 12: Transparency Information

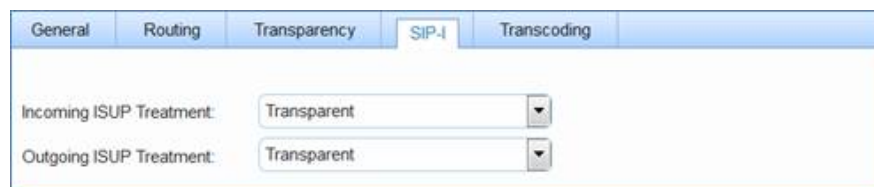


- Select the appropriate Dynamic Determination for Topology, Dialog, Identity, Accounting.
- Select the desired transparency options.

The default settings incorporate dynamic conditions, such as the interface types and trust levels. In most configurations this should be sufficient. In the default settings shown below, only the topology and dialog transparency values are computed dynamically. The remaining values are determined statically.



## SIP-I tab



General Routing Transparency SIP-I Transcoding

Incoming ISUP Treatment: Transparent

Outgoing ISUP Treatment: Transparent

- Incoming ISUP Treatment. Select the incoming ISUP treatment profiles, using the drop-down menu: Possible values:
  - **Transparent.** Allows the current behavior to be retained while passing the SIP-I encapsulated body transparently to the next leg. This is the default value.
  - **Reject.** Rejects the Incoming INVITE with the internal cause code *ISUPMessageBodyNotSupported*, regardless of whether the content handling is required or optional in the SIP Headers.
  - **SIP Only.** Removes the ISUP encapsulated body and continues with the normal forwarding for the SIP message.
  - **SIP-I if available.** Decodes the ISUP body using the ETSI protocol for the received INVITE with the IAM. For more information, see [SIP-I Support](#).
- Outgoing ISUP Treatment. Select the outgoing ISUP treatment profiles, using the drop-down menu: Possible values:
  - **Transparent.** Passes the SIP-I encapsulated body transparently to the network.
  - **Reject.** Rejects the call from the Egress leg.
  - **SIP Only.** Removes the ISUP encapsulated body before sending the SIP message to the network.
  - **SIP-I if available.** Encodes the ISUP body based on the input message to the outgoing INVITE message.
  - **Always SIP-I.** Appends an ISUP body to the outgoing message. For more information, see [SIP-I Support](#).

## Transcoding tab



General Routing Transparency SIP-I Transcoding

Transcoding Profile: Select

Enter the transcoding profile.

## Access tab

If **Access-Public** was the selected **Network Type**, select the **Access tab**.

---

**Note:**

If Access-Public was not the selected Network Type, go to the next step.

---

The screenshot shows the 'Add Service Profile' dialog box with the 'Access' tab selected. The fields are as follows:

- Max Allowed Registrations: 32000
- Min Registration Interval: 1800
- Max Registration Interval: 3600
- Forward Registration on Expiry (%): 50
- Release Sessions & Subscriptions upon Registration Expiry:
- Far-End NAT Traversal Mode: ShortReg
- Far-End NAT Traversal Interval: 60
- Anti-Tromboning:  Yes  No

- Max Allowed Registrations (default value is 32000 seconds)
- Min Registration Interval (default value is 1800 seconds)
- Max Registration Interval (default value is 3600 seconds)
- Forward Registration on Expiry (default value is 50%)
- Subscribe to Registration Events (currently not supported)
- Release Sessions and Subscriptions upon Registration Expiry (default value is No)
- Far-End NAT Traversal Mode (default value is ShortReg)
- Far-End NAT Traversal Interval (default value is 60 seconds).

#### LRBT tab

BorderNet SBC, during the call setup, can be provisioned to play a local **Ring Back Tone (RBT)**, notifying the originator that the terminator's device is ringing.

The BorderNet SBC includes a sample LRBT package (non-licensed).

The system operator can replace the sample tones with customized LRBT files.

These files must include a default country-specific tone (**defLRBT**), and additional optional customized tones.

- The LRBT tab allows the setting of the following parameters:

- Generate LRBT. Determines whether the LRBT is enabled, and if enabled it allows the user to select either the default LRBT or any customized LRBT tones, using a drop-down menu. The LRBT tones (files) are displayed in the list, only when they have been prepared and uploaded, to the system (see section *Generate Local Ring Back Tone Packages* in *BorderNet SBC Maintenance User's Guide* document, and [Customized LRBT](#)).
- Max play time for LRBT. The maximum time interval, to play the LRBT. Range: [30 to 300] seconds, default: 60 seconds.
- 180/183 type to orig. The message sent back to the originating peer to notify the LRBT.
- Possible values:
  - **Default - transparent.** The same 180/183 message (without SDP), received from the terminating peer, is sent to the originating peer (including SDP).
  - **Always 180.** 180 message with SDP is sent to the originating peer to notify the LRBT.
  - **Always 183.** 183 message with SDP is sent to the originating peer to notify the LRBT.

## 8.4.1 SIP-REC

The SIP-based protocol is used for the control of a session media recorder in the recording and storage of media sessions. It is primarily used by call centers, financial tradings etc i.e. anywhere where voice recording is required for regulatory compliance. Calls can also be recorded for quality control or business analytics purposes.

The **Session Recording Solution** includes a centralized recording server with centralized storage.

There are two elements involved in session recording:

- the Session Recording Client (SRC)
- the Session Recording Server (SRS)

The SRC-SRS interface is standardized by IETF.

The SIP-REC is a session recording according to the **Session Recording Protocol**. BorderNet serves as an SRC towards a third-party SRS supplied by the customer.

SIP-REC consists of SIP extensions to deliver RTP media and related information to a recording device, using SIP, SDP, RTP.

The system layout of the BorderNet SBC SIP-REC support is shown in the diagram below which illustrates a communication session between two SIP **User Agents (UAs)**.

The link between the BorderNet SRC and the SRS shall be secured externally using a router or using IPsec. The SRC can deliver to the SRS multiple media streams (audio and video) in a given communication session.

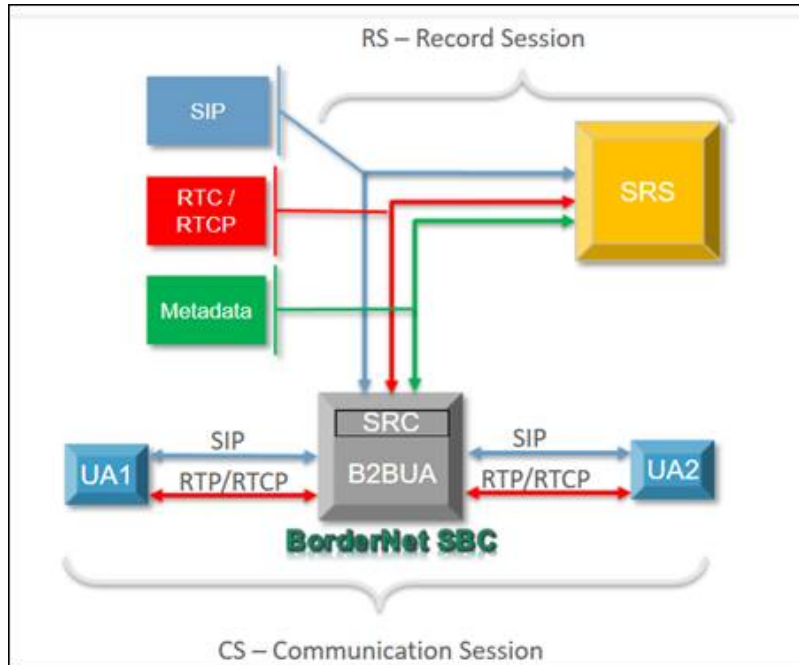


Figure 12: BorderNet System Layout

## 8.4.2 SIP Protocol Requirements

A BorderNet SRC supports an option tag called "siprec".

This tag:

- indicates to both sides of the call that the session is being recorded.
- is included at the request SIP header (i.e. INVITE) as require=siprec.
- will reject requests from the SRS if require=siprec is not included.

BorderNet SRC detects a "record aware" user agent option tag and sends a recording indication via the SDP **a=record** attribute. BorderNet SRC is the only entity allowed to initiate a recording session.

If a session is initiated by the SRS it will be rejected by the **403=forbidden** warning message.

The following protocol extensions are available in BorderNet:

- siprec option tag - used for requests to the SRS.
- +sip.src feature tag - signals that BorderNet is a SIP-Rec SRC and identifies a SIP dialog as a recording session.
- a=record SDP attribute - indicates if recording is on, off or paused, and is used as a recording indication to the recording-aware user agent.
- a=recordpref SDP attribute - allows a recording-aware user agent to inform the BorderNet on its recording preference.
- Metadata conveys information about the session being recorded and BorderNet generates a full metadata snapshot on session establishment, as well as partial metadata updates.

SIP-REC tab

The SIP-REC tab is provisioned when a call recording using a SIP-REC framework is desired.



- Recording. If set to Yes then SIP recording will be activated. When set to No then SIP-REC is set to OFF and there will be no attempt to create a recorded session (RS) with an SRS.
- Recording Preference. If set to Support then the BorderNet SBC will honor the preference indicated in the SDP "a=recordpref:" attribute of a recording-aware UA. When set to Ignore then the "a=recordpref:" attribute is not considered.
- SRS Peer. Select one of the SRS peers available, which are configured with an SRS type in the peer configuration screen. This is the recording server for which the RS (recorded session) will be created.

## 8.5 Emergency Profiles

### 8.5.1 Overview

BorderNet SBC recognizes emergency calls based on their **Req-URI**.

If the call's Req-URI is configured in an emergency profile, configured for the peer/interface in the Service Profile, then the call is marked as "**Emergency**", and the resource priority header is added to the message.

---

**Note:**

System Administrator privileges are required to enable Emergency Service.

---

**Note:**

The SDR for the call is marked as emergency to be used for billing.

---

The BorderNet SBC allows Emergency Calls to exceed the licensed call limit by up to 300 calls.

---

**Note:**

The maximum call capacity of the BorderNet SBC is 32,000 calls. Emergency Calls cannot exceed this limit. If a customer has a license for 31,700 calls, the addition 300 Emergency Calls will apply. If a customer has a license for 31,800 calls, only 200 additional calls will be available in case of an emergency.

---

### 8.5.2 Emergency Call Configuration

→ To configure an emergency service:

1. Review the BorderNet's default emergency profiles, including the standard emergency URI, as shown in the table below. Emergency profiles can be customized, and new emergency profiles can be added or cloned from the default profile to suit the geographical location:

URI/Number	Description	Resource Priority
counseling	Counseling services	ets.2
counseling.children	Counseling for children	ets.1
counseling.mental-health	Mental health counseling	ets.1
counseling.suicide	Suicide prevention hotline	ets.0
sos	Emergency services	ets.0
sos.ambulance	Ambulance service	ets.4
sos.animal-control	Animal control	ets.0
sos.fire	Fire service	ets.0
sos.gas	Gas leaks and gas emergencies	ets.0
sos.marine	Maritime search and rescue	ets.4
sos.mountain	Mountain rescue	ets.4
sos.physician	Physician referral service	ets.1
sos.poison	Poison control center	ets.1
sos.police	Police, law enforcement	ets.1

Table 13: Default Emergency Tables

1. If the default profiles meet your needs, skip to **Assigning the Emergency Profile**. Otherwise proceed to the next step.
2. **Create a New Emergency Profile**
3. Assign the emergency profile to the service profile, used for the call, by selecting the **Emergency Profile** displayed in the **General** tab, as the service profile.
4. Set the **Maximum Emergency Sessions** parameter in the ingress security profile to a non-zero number (see **Security Profile, Network** tab).
5. Create a routing policy (see **Advanced Policy**).
  - In the **Create Rule** window, select the **Rule using Parameter**.
  - Set the **Parameter** to **Emergency Call**.
  - Set the **Action** to **IsPresent**.
  - Select **Yes** for the **Value**.
6. In the Service Profile, **Routing** tab, select the the newly created advanced policy for the incoming Peer service profile.
7. Set the **Maximum Emergency Sessions** parameter in egress security profile to a non-zero number (see **Security Profile, Network** tab).

### 8.5.3 Creating a New Emergency Profile

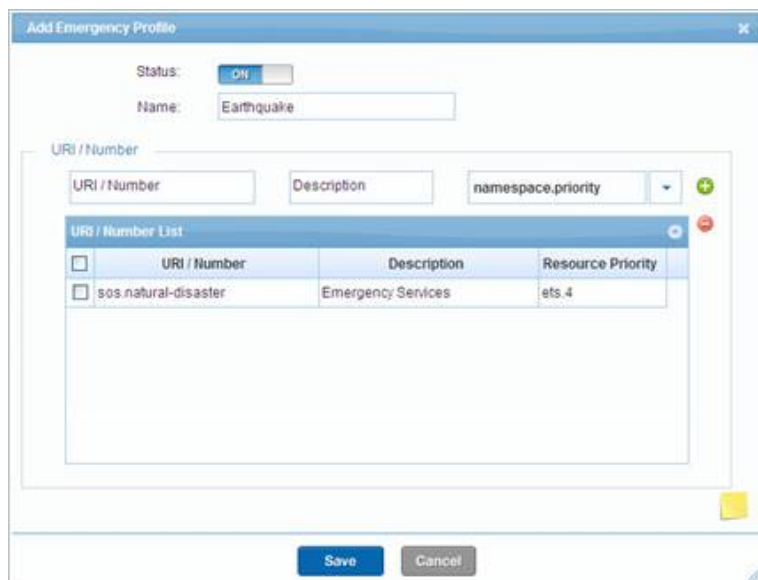
→ To create a new emergency profile:


1. Select **Application** → **Common** → **Emergency Profile**.  
The **Emergency Profile** window opens.

Emergency Profile				+ Add Emergency Profile
	Status	Name	URI / Number	
	<input type="checkbox"/>		spectra	911:sos:
	<input type="checkbox"/>	Default:	counseling.counseling.children;counseling.mental-health;counsel	

2. Select **+Add Emergency Profile**.

The **Add Emergency Profile** window opens.



3. Enter the **Name**, **URI/Number**, **Description**.
4. Select the **Resource Priority** from the drop-down list.
5. Click the green plus  icon to add the **Emergency Profile** to the **URI/Number** list.
6. Click **Save**.

The **Emergency Profile** has been created.

## 8.6 Codec Profiles

BorderNet SBC contains a comprehensive set of codec profiles that include audio, video and image media types. The operator can also create customized codec profiles or modify the existing profiles.

Each codec profile has additional criteria, such as FMTP parameters, RTP clock rate, channels or a combination of these. BorderNet SBC can detect variants within a codec family and apply more relevant profiles, which can impact the estimated packet rate and bandwidth.

Using codec profiles, BorderNet SBC sets a specific codec notation when sending out the SDP, to enable interoperability.

→ To configure a codec profile:

1. Select **Application** → **Common** → **Codec Profiles**.

The **Codec Profile - Configuration** window opens.

Profile Name	Display by Default	Codec Type	Media Type	Media Identity	Media SubType (format)	Description
AnyAudio	Yes	Any	audio	Any	Any	
clearmode	Yes	clearmode	audio	CLEARMODE	CLEARMODE	64 kbit/s channel data (all possibl
tone	Yes	tone	audio	tone	tone	
telephone-event	Yes	telephone-event	audio	telephone-event	telephone-event	
t38_audio	Yes	t38_audio	audio	t38	t38	
t38_image	Yes	t38_image	image	t38	t38	
VDV1	No	VDV1	audio	VDV1	VDV1	Variable-rate version of D/V14. RFC
RED_Forward	No	Custom	audio	fedred	fedred	
RED	No	RED	audio	RED	RED	A mechanism to issue multiple rec Atlist the addition of low-bandwid packet loss. application designers of large amounts of redundancy as hence packet loss, leading to a wo use of redundancy was intended to excessive network congestion a
L24	No	L24	audio	L24	L24	Linear PCM 24-bit audio
L20	No	L20	audio	L20	L20	Linear PCM 20-bit audio
Sample_AMR-WB+	No	Custom	audio	AMR-WB+	AMR-WB+	Extended Adaptive Multi Rate Wid

2. To edit a codec profile, select the **Edit** icon  or double-click the codec entry.

1. The **Edit Codec Profile** window opens.

**Edit Codec Profile**

Profile Name:

Display by Default:  Yes  No

Codec Type:

Media Type:

Media Identity:

Media SubType (format):

RTP Clock Rate:

Channels:

fmp Parameters:

Static Payload Type:

Bit Rate (kbps):

Ptime:

Redundancy Factor:

Wire BitRate (kbps):

Identification Format:

Description:

2. Edit the Codec and click **Save** to save the changes.

### DTMF Codec Profile Customization

BorderNet SBC has the ability to perform RFC 2833-based **Dual Tone Multi-Frequency (DTMF)** payload conversions for telephone events. In this scenario, the BorderNet SBC acts a bridge between two Peers.

BorderNet SBC intercepts the INVITE from the offering Peer, changes the payload type number of the telephone event in the SDP, and sends the offer to the answering Peer. The answering Peer's response is sent back to BorderNet SBC, which converts the number of the telephone event back to the original number and sends it to the offering Peer.


→ To customize a DTMF codec profile:

1. Select **Application** → **Common** → **Codec Profiles**.

The **Codec Profile - Configuration** window opens.

Codec Profile - Configuration							
	Profile Name	Display by Default	Codec Type	Media Type	Media Identity	Media SubType (format)	Description
	telephone						
	telephone-event-ss	Yes	Custom	audio	telephone-event	telephone-event	
	telephone-event-tp	Yes	Custom	audio	telephone-event	telephone-event	
	telephone-event-s	Yes	Custom	audio	telephone-event	tone telephone-event	
	telephone-event	Yes	telephone-event	audio	telephone-event	telephone-event	

2. Create a custom **Codec Profile**.

- Select the default telephone-event.
- Select **Clone** from the **Action List** icon .

1. The **Clone Codec Profile** window opens.

**Clone Codec Profile**

Profile Name:

Display by Default:  Yes  No

Codec Type:

Media Type:

Media Identity:

Media SubType (format):

RTP Clock Rate:

Channels:

fmt Parameters:

Static Payload Type:

Set Dyn Payload Type:

Bit Rate (kbps):

Ptime:

Redundancy Factor:

Wire BitRate (kbps):

Identification Format:

Description:

2. Enter a **Profile Name**.
3. Enter the dynamic payload type (possible values: 96 - 127) in the **Set Dyn Payload Type** field.
4. Click **Save**.
5. Verify that the **Custom** profile has been added the **Codec Profile Configuration** window.

Profile Name	Display By Default	Codec Type	Media Type	Media Identity	Media SubType (format)	Description
telephone						
telephone-event-ss	Yes	Custom	audio	telephone-event	telephone-event	
telephone-event-tp	Yes	Custom	audio	telephone-event	telephone-event	
telephone-event-s	Yes	Custom	audio	telephone-event	tone telephone-event	
telephone-event	Yes	telephone-event	audio	telephone-event	telephone-event	

6. Select the newly created telephone event for the egress **Media Profile** (see SIP [media profile](#), audi tab).

## 8.7 Transcoding Profiles

### 8.7.1 Transcoding Overview

Transcoding refers to the conversion of one media type to another.

In the BorderNet SBC transcoding the following media type conversions are provided:

- Codecs
- Packetization
- DTMF
- Fax

High Availability is supported for transcoding sessions. **Transcoding information is mirrored between the active and standby systems. Upon a failover the transcoded sessions are maintained** in the same manner as in the regular sessions.

Transcoding is implemented for SIP only. BorderNet supports both the **G.722** and the **G.726** codes for transcoding (Beta version).

In this release the Software transcoding is applied. The Hardware and the External Platform transcoding methods are not in the scope of this document, and this release.

For detailed information on BorderNet SBC Transcoding, see the *BorderNet SBC Transcoding User's Guide* document.

### 8.7.2 Transcoding Activation

To activate the transcoding, the service should be enabled per SIP interface/peer (see [Transcoding Configuration](#)). The peer profile overwrites the interface profile (used only if a peer profile does not exist).

A call is transcoded only if both the ingress and egress call legs are assigned with a transcoding profile.

Transcoding is enabled only if a transcoding license is loaded to the BorderNet SBC. The license is provided per session, limiting the total number of allowed transcoded sessions.

If the number of transcoded sessions reaches the license limit, then any additional call which requires transcoding is rejected

A transcoding license updates the maximum number of allowed sessions with transcoding. If the current number of transcoding sessions exceeds the new license limit, then the calls are gracefully terminated. Additional license is needed for AMR-WB/G.722.1 up to the number of sessions purchased.

Licensing state is viewed in the license status window, located in System →License.

---

**Note:**

BorderNet fully supports AMR-NB on all its rates.

---

## 8.7.3 Transcoding Configuration

The transcoding profile configuration is provided either via assignment to a service profile or through the advanced policy.

→ To configure transcoding:

The first step in configuring transcoding is to create a **Transcoding Profile** and associate it with a Service Profile.

You can do one of the following:

- Select the Service Profile in the Transcoding tab.
- In the Advanced Policy **Treatment** window, select Media as Type, ActivateTranscoding as the Treatment parameter, and in the Input select the transcoding profile.

---

**Note:**

A transcoding profile assigned via an advanced policy is assigned to both the ingress and egress legs. The transcoding profile selected in the Advanced Policy treatment overrides the transcoding profile selected in the service profile.

---

→ To start or stop transcoding:

1. In the System Services, select the Transcoding Service.
2. Click the **Edit** icon to start or stop transcoding.

---

**Note:**

The **Transcoding Service Unavailable Alarm** is raised if transcoding is stopped.

---

→ To disable transcoding, using the Advanced Policy treatment window:

1. In the **Advanced Policy Treatment** window, select **Media** as **Type**.
2. Select **ForceTranscodingOff** as the **Treatment** parameter.
3. In the **Input** field select **Yes**.

## 8.7.4 Fax Transcoding

The BorderNet supports fax transcoding between T.38 fax relay and G711 Fax Pass-through.

Fax transcoding shall be performed only if a transcoding profile is assigned. Fax transcoding with T.38 is supported only with a single image line, using a re-invite to indicate a fax image type.

Multiple m-lines in an offer means multiple (concurrent) media streams are being offered and used. Using multiple 'm' lines with one of them being T.38 will not trigger transcoding.

Using RTP as a transport for fax is not supported (rfc-4612, definition of the audio-RTP type for T.38). Only UDPTL is used for transcoding.

The BorderNet SBC supports High Availability configurations for transcoding sessions.

Fax transcoding is activated by enabling a checkbox by a parameter in the **Transcoding Profile**.

For detailed information on the BorderNet SBC Transcoding see the *BorderNet SBC Transcoding User's Guide* document.

## 8.7.5 DTMF Transcoding

**Dual Tone Multi-Frequency (DTMF)** transcoding allows the interoperability of different methods of transporting the DTMF when regular transcoding is not necessary.

There are two methods for triggering DTMF transcoding:

- When regular codec transcoding is used the DTMF transcoding is handled as part of the transcoding process.
- When regular transcoding is not required as both legs of the call are using the same codec, DTMF transcoding may be required if there is a mismatch of the DTMF transporting method. DTMF only transcoding can be triggered as long as the ingress and egress use the same coding.

There are 3 common ways to convey DTMF in a call:

- In-band - tones are encoded as regular speech with no extra handling (G.711 codec only).
- Out of band - tones are not carried in the regular RTP audio stream but sent in a call control signaling protocol. For example: SIP INFO messages with Content Type: application/dtmf-relay.
- RFC-2833 - tones are transported in the RTP stream but encoded differently to the codec used for the voice stream. The RTP packets sending the tones have a different payload type and a dedicated DTMF encoding method as per the RFC-2833 definition. If both call legs support DTMF via RFC-2833 there is no need for DTMF transcoding as it would be a waste of resources.

---

### Note:

The DTMF only transcoding will be triggered only if the field **Force DTMF Interworking** is set to **Yes**.

---

BorderNet SBC supports DTMF transcoding between each pair of the following DTMF methods:

- SIP INFO (out-of-band)
- In-band
- RFC-2833 (telephony events)



The following table illustrates the situations in which DTMF transcoding will be used:

	SIP INFO	In-band	RFC 2833
SIP INFO		transcode	transcode
In-band	transcode		transcode
RFC 2833	transcode	transcode	

For example, on a call from RFC-2833 to in-band or SIP INFO, the DTMF transcoding will be triggered when the RFC-2833 telephony events are indicated in the ingress received SDP and the SDP answer received from the egress doesn't include an RFC-2833 reference, then DTMF transcoding will be triggered.

If the transcoding profile assigned in the **DTMF Non-RFC2833 Preference** field is set to **Inband** and the negotiated codec is NOT G.711, then no DTMF transcoding will be performed.

Transcoding Mode: Unconditional

DTMF Non-RFC2833 Preference: Inband

Force DTMF Interworking:  No  Yes

Save Cancel

→ To configure DTMF transcoding:

1. Open the **Application** tab and move to the **Common** pane.
2. Select **Transcoding Profiles**.  
The **Transcoding Profile** window opens.
3. Click the blue **+Add Transcoding Profile** button at the right-hand side of the screen.  
The **Add Transcoding Profile** window opens.

Add Transcoding Profile

Status:  On

Name:

Transcoding Module: TclModule\_1

Destination FQDN:

Audio Codec List

Available: SW\_AMRNB, SW\_AMRWB, SW\_CH\_VAD, SW\_G722, SW\_G723\_5.3\_NoVAD

Selected:

Transcoding Mode: 4xMediaUnsupported

DTMF Non-RFC2833 Preference: Inband

Enable Fax Transcoding:  No  Yes

Force DTMF Interworking:  No  Yes

Save Cancel

4. Use the **Audio Codec List** to filter your selected codecs.
5. In the **Force DTMF Interworking** field, check the **Yes** button.
6. In the **DTMF Non-RFC2833 Preference** field, select either **Inband** or **SIP INFO**.
7. Click **Save**.

The DTMF transcoding profile is created.

## 8.7.6 Creating a Transcoding Profile

→ To configure a Transcoding profile:

1. Select Application → Common → Transcoding Profiles.

The Transcoding Profile window opens.

Status	Name	Audio Codecs	Transcoding Mode
ON	Trans_Prof1	SW_G723_Michael,SW_PCMA,SW_PCMU,SW_G729	Unconditional

This window displays the list of the existing profiles, including the following parameters:

- **Status.** The transcoding profile is enabled (on) or disabled (off). Status. The transcoding profile is enabled (on) or disabled (off).
- **Name.** The name of the transcoding profile. Name. The name of the transcoding profile.
- **Audio Codecs.** The list of the codecs allowed for the selected transcoding profile. Audio Codecs. The list of the codecs allowed for the selected transcoding profile.
- **Transcoding mode.** The condition applied on the transcoding. Transcoding mode. The condition applied on the transcoding.
- **Possible values:**
  - o4xxMediaUnsupported
  - oCodecListMismatch
  - oCodecPriorityMismatch
  - oUnconditional (default)
- See Codec Transcoding Triggers in *BorderNet SBC Transcoding User's Guide*.
- To edit an existing profile, double-click on the selected transcoding profile.
- To create a new profile click +Add Transcoding Profile.
- Enter the following parameters in the Transcoding Profile window:

- **Status.** Enables (on) or disables (off) the transcoding profile. Status. Enables (on) or disables (off) the transcoding profile.
- **Name.** The name of the transcoding profile. Name. The name of the transcoding profile.
- **Transcoding Module.** Not selectable. Transcoding Module. Not selectable.

- Destination FQDN. Not relevant for software transcoding. Destination FQDN. Not relevant for software transcoding.
- Audio Codec List. The list of the codecs allowed for transcoding. Audio Codec List. The list of the codecs allowed for transcoding.
- Select a codec and click Add>> to add it to the Selected list of codecs. Select a codec and click Add>> to add it to the Selected list of codecs.
- Transcoding mode. Select the conditions to apply the transcoding. Transcoding mode. Select the conditions to apply the transcoding.
- Possible values:
  - o4xxMediaUnsupported
  - oCodecListMismatch
  - oCodecPriorityMismatch
  - oUnconditional (default)
- See Codec Transcoding Triggers in *BorderNet SBC Transcoding User's Guide*.
- DTMF Non-RFC2833 Preference. DTMF Non-RFC2833 Preference.
- Possible values:
  - oInband. If there is no DTMF signaling (no RFC-2833 indication), then prefer to transcode to inband DTMF (assuming G.711 is used).
  - oSIP\_INFO. If there is no DTMF signaling (no RFC-2833 indication), then prefer to transcode the DTMF into SIP INFO signaling.
  - oDTMF Trigger Transcoding. Select **Yes** to trigger DTMF transcoding also when codec transcoding is not used (same codec but different DTMF transport method) or select **No** to disable it and use transcoding only when different codecs are used.

## 8.8 Diameter Profile

Diameter is an **Authorization, Authentication and Accounting (AAA)** protocol, providing a framework for applications requiring such AAA services.

IMS & 3GPP are using the Diameter protocol as the transport of charging events, in order to provide offline and online charging services, using the **Rf** (offline charging reference point between a 3G network element and the CDF) and **Ro** (online charging reference point between a 3G network element and the OCS) reference points respectively.

Diameter is also used for other reference points in the IMS architecture.

### 8.8.1 Diameter Overview

Diameter creates a framework for **Authentication, Authorization and Accounting (AAA)** transport. It details a base protocol that defines the minimum mandatory set of AAA operations, which can then be enriched with additional capabilities by using specific Diameter applications. These capabilities can be developed by extending existing applications or by creating new ones. For example, the **IETF Diameter Credit Control Application** creates a new application for online charging, and the 3GPP extends this application with additional AVPs to support the exchange of charging information for the **Ro** reference point.

All data delivered by the protocol is in the form of **Attribute-Value Pairs (AVPs)**, which is basically a flexible container of data. It is constructed of an attribute name and its value *<attribute name, value>*, making it a "pair" representing the data.

Diameter is a peer-to-peer protocol that uses a request-answer transaction format, in which any node can initiate a request. Every request sent from a Diameter client must be replied to with a Diameter answer from the server side.

- A Diameter client is a device at the edge of the network that performs access control and generates Diameter messages to request AAA services for the user.
- A Diameter server is a Diameter node that handles AAA requests from clients, although it also supports server-initiated requests.

A **Diameter** message consists of a fixed-length header followed by a variable number of AVPs. The amount and type of AVPs attached to each message (**Request/Answer**) is dependent on the command associated with the message.

## 8.8.2 Offline & Online Charging

Offline charging is a process where charging information for network resource usage is collected concurrently with that resource usage. The charging information is used to construct CDR files, which are then transferred to the **Network Operator's Billing Domain (BD)** for the purpose of subscriber billing, and/or inter-operator accounting.

Online charging is a process where charging information for network resource usage is collected concurrently with that resource usage in the same fashion as in offline charging. However, authorization for the network resource usage must be obtained by the network prior to the actual resource usage to occur. This authorization is granted by the **Online Charging System (OCS)** upon request from the network.

In offline charging, the resource usage is reported from the network to the BD after the resource usage has occurred. The charging information does not affect, in real-time, the service rendered. In online charging, a subscriber account, located in an OCS, is queried prior to granting permission to use the requested network resource(s). Charging information can affect, in real-time, the service rendered and therefore a direct interaction of the charging mechanism with the control of network resource usage is required.

In the case of **Offline Charging (Rf)**, the DCS will be able to connect to more than a single peer, allowing it to overcome failures on its connected peers. In case of a detected failure in the primary peer, the DCS will send the messages to the secondary peer IP address.

In the case of **Online Charging (Ro)**, since this is used for real-time applications, the behavior of a possible fail detection and failover mechanism should be defined separately. A failover of an ongoing real-time session to a secondary server must incorporate a complex updating solution between the primary and secondary servers.

## 8.8.3 Diameter Rx Interface

The **Diameter Rx** reference point is used for policy control of sessions on the **IP Connectivity Access Network (IP-CAN)** and is operated between the **P-CSCF (Proxy-Call Session Control Function)** and the **PCRF (Policy and Charging Rule Function)**. The **PCRF** provides network control regarding service data flow detection, gating (blocking or allowing packets), QoS control and flow-based charging towards the **PCEF (Policy and Charging Enforcement Function)**.

When the **Policy and Charging Control (PCC)** is used in the network the **P-CSCF** sends information obtained from SIP/SDP session setup signaling to the **PCRF** via the **Rx reference point**.

This information enables the **PCRF** to form authorized IP QoS data (e.g. maximum bandwidth and QoS class) and charging rules that will be delivered to the access gateway via the **Gx reference point**.

The **P-CSCF** is tasked to send policy information to the **PCRF** about every SIP message that includes an SDP payload. This ensures that the **PCRF** passes the proper information to perform policy and charging control for all possible IMS session setup scenarios.

Similarly, the **PCRF** utilizes the **Rx reference point** to send notifications of bearer events to the **P-CSCF**. For passing the information, the **P-CSCF** and **PCRF** use a **Diameter protocol** as defined in **3GPP TS 29.214**.

The **Diameter Rx** interface therefore relies mainly on the following standards:

- IETF rfc6733 - Diameter Base Protocol
- IETF rfc7155 - Diameter Network Access Server Application
- 3GPP 29.214 - Policy and Charging Control over Rx reference point

The **Diameter Rx** properties use the existing Diameter profile configuration screen, which is available for **Rf & Ro**. An **Rx** interface activation checkbox is available in the **SIP interface configuration** screen.

The Rx uses message types as defined in **RFC6733 Diameter Base Protocol**, with the addition of the **AAR (Authentication Authorization Request)** message type defined in **RFC7155 Diameter NASREQ**.

The license for **Diameter Rx** is per Diameter feature. The entire feature is either enabled or disabled regardless of the number of concurrent sessions using it.

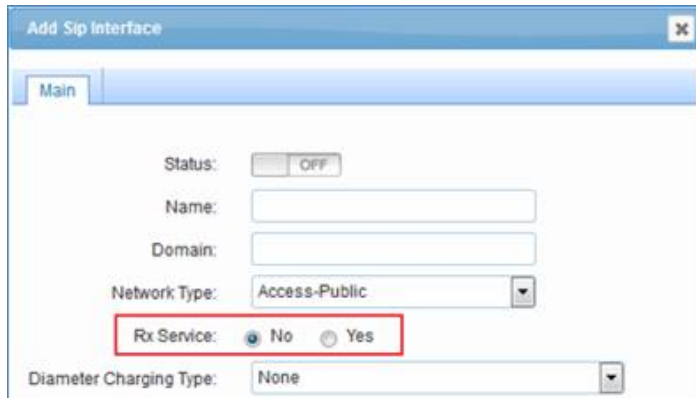
---

#### NOTES:

- **Rx** Diameter connections to the **PCRF** shall be independent of the **Ro** and **Rf** Diameter connections used for the **OCS/OCF** and the **CDF** accordingly.
  - **Rx** user validation shall be enabled only for the **Access-Public** interface type.
  - **Rx** validation shall be performed only if the **Rx Interface** parameter is set on the **SIP interface** configuration screen. Otherwise no **Rx** handling is required.
  - **Rx** messages are sent by the BorderNet to the Diameter server (**PCRF**) only when the BorderNet receives a SIP message with SDP.
  - **Rx** authorization process shall be performed before the call is routed, and before any **Rf** or **Ro** messages are sent.
- 

The **Diameter Rx** interface:

- is not dependent on Rf/Ro
- is applied only on Access Public interfaces.
- sends an Rx message only when the BorderNet receives or sends a SIP message with SDP
- is handled per offer/answer, regardless of the message carrying it



The screenshot shows the 'Add Sip Interface' configuration window. The 'Main' tab is selected. The 'Status' is set to 'OFF'. The 'Name' and 'Domain' fields are empty. The 'Network Type' is set to 'Access-Public'. The 'Rx Service' checkbox is checked, and the 'Yes' radio button is selected. The 'Diameter Charging Type' is set to 'None'.

Activate Rx Service

## 8.8.4 Configuring a Diameter Profile

→ To configure a Diameter profile:

1. Select Application → Common → Diameter Profiles.

The Diameter - Profile Configuration window opens.

/ Application / Diameter Profile

Diameter - Profile Configuration

Status	Product Name	DCS Host	DCS Realm	Transport Protocol	DCS Local Interface	Dcs Source Port	Watch Dog timer
<input checked="" type="checkbox"/>	Dialogic	ctf.tek.com	tek.com	TCP	10.10.10.7 (lan1)	3868	30

This window displays the list of the existing profiles.

1. To edit an existing profile, double-click on the selected diameter profile.
2. Enter the following parameters in the Edit Diameter Profile window, General tab.

Edit Diameter Profile

General CDF CDF AVPs OCF OCF AVPs PCRF

Status:  ON

Product Name: Dialogic

Dcs Host: ctf.tek.com

Dcs Realm: tek.com

Transport Protocol:  TCP

DCS Local Interface: 10.10.10.24 (Interface-1 ▼)

Dcs Source Port: 3868

Watch-Dog Timer(Tw): 30

Watch-Dog No Response: 3

Save Cancel

3. Edit the following parameters:
  - Status. The diameter profile is enabled (on) or disabled (off). Default off.
  - Product Name. The name of the product. A user configured string.
  - DCS Host. A user configured string.
  - DCS Realm. A user configured string.
  - Transport Protocol. Set to TCP.
  - DCS Local Interface. Dropdown list of available signaling IP addresses.
  - DCS Source Port. Value from 1024 to 65535.
  - Watch Dog Timer. Value from 6-30 seconds. Default 30 secs.
  - Watch Dog No Response. Value from 1-5 seconds. Default 3 secs.
4. Click Save.

1. Move to the **CDF** tab.

The screenshot shows the 'Edit Diameter Profile' dialog box with the 'CDF' tab selected. The form contains the following fields and values:

Field	Value
Primary IP Address:	IPv4 10.10.30.188
Primary Destination Port:	3868
Primary Realm Name:	tek.com
Primary Host Name:	cdf.tek.com
Secondary IP Address:	IPv4 10.10.30.12
Secondary Destination Port:	3869
Secondary Realm Name:	tek.com
Secondary Host Name:	cdf.tek.com
ACR Retransmission Interval:	3
ACR Retransmission Attempts:	3

At the bottom of the dialog, there are 'Save' and 'Cancel' buttons. The 'Save' button is highlighted in blue.

2. Edit the following parameters:

- **Primary IP Address.** Indicate whether **IP4/IP6**. **User configured IP address.** **Primary IP Address.** Indicate whether **IP4/IP6**. **User configured IP address.**
- **Primary Destination Port.** Set to **3868**. **Primary Destination Port.** Set to **3868**.
- **Primary Realm Name.** A user configured string. **Primary Realm Name.** A user configured string.
- **Primary Host Name.** A user configured string. **Primary Host Name.** A user configured string.
- **Secondary IP Address.** Indicate whether **IP4/IP6**. **User configured IP address.** **Secondary IP Address.** Indicate whether **IP4/IP6**. **User configured IP address.**
- **Secondary Destination Port.** Value from **1024 to 65535**. **Secondary Destination Port.** Value from **1024 to 65535**.
- **Secondary Realm Name.** A user configured string.
- **Secondary Host Name.** A user configured string. **Secondary Host Name.** A user configured string.
- **ACR Retransmission Interval.** Value from **1-10** seconds. Default **3** secs. **ACR Retransmission Interval.** Value from **1-10** seconds. Default **3** secs.
- **ACR Retransmission Attempts.** Value from **1-6**. Default **3** attempts. **ACR Retransmission Attempts.** Value from **1-6**. Default **3** attempts.

3. Click **Save**.

4. Move to the **CDF AVPs** tab.

5. Indicate **Yes/No** by clicking the radio button for each of the fields in the tab.

Default values are as follows:

- Destination Host: **Yes**
- Service Context ID: **No**
- Subscription ID Data: **No**
- Subscription ID Type: **No**
- From Address: **Yes**
- SIP Method: **Yes**
- Originating IOI: **Yes**
- Terminating IOI: **Yes**
- SDP Media Description: **No**
- SDP Media Name: **No**
- SDP Type: **No**
- Event Timestamp: **No**

- Calling Party Address: **Yes**
- Called Party Address: **Yes**
- SDP Session Description: **No**
- Acct Application ID: **Yes**

**Edit Diameter Profile**

General | CDF | **CDF AVPs** | OCF | OCF AVPs

Destination-Host:  Yes  No

Service-Context-Id:  Yes  No

Subscription-Id-Data:  Yes  No

Subscription-Id-Type:  Yes  No

From-Address:  Yes  No

Sip-Method:  Yes  No

Originating-loi:  Yes  No

Terminating-loi:  Yes  No

Sdp-Media-Description:  Yes  No

Sdp-Media-Name:  Yes  No

Sdp-Type:  Yes  No

Event-Timestamp:  Yes  No

Calling-Party-address:  Yes  No

Called-Party-Address:  Yes  No

Sdp-Session-Description:  Yes  No

Acct-Application-Id:  Yes  No

6. Click **Save**.

7. Move to the **OCF** tab.

**Edit Diameter Profile**

General | CDF | CDF AVPs | **OCF** | OCF AVPs

Primary IP Address:

Primary Destination Port:

Primary Realm Name:

Primary Host Name:

Secondary Ip Address:

Secondary Destination Port:

Secondary Realm Name:

Secondary Host Name:

Response-Waiting-time(Tx):

**Save** **Cancel**

8. Edit the following parameters:

- **Primary IP Address.** Indicate whether IP4/IP6. User configured IP address.



- **Primary Destination Port.** Set to **3868**.
  - **Primary Realm Name.** A user configured string.
  - **Primary Host Name.** A user configured string.
  - **Secondary IP Address.** Indicate whether IP4/IP6. **User configured IP address.**
  - **Secondary Destination Port.** Value from **1024** to **65535**.
  - **Secondary Realm Name.** A user configured string.
  - **Secondary Host Name.** A user configured string.
  - **Response Waiting Time (Tx).** Value from **1-60** seconds. Default **10** secs.
9. Click **Save**.
  10. Move to the **OCF AVPs** tab.
  11. Indicate **Yes/No** by clicking the radio button for each of the fields in the tab.

Default values are as follows:

- Destination Host: **Yes**
- Service Context ID: **No**
- Subscription ID Data: **No**
- Subscription ID Type: **No**
- From Address: **Yes**
- SIP Method: **Yes**
- Originating IOI: **Yes**
- Terminating IOI: **Yes**
- SDP Media Description: **No**
- SDP Media Name: **No**
- SDP Type: **No**
- Event Timestamp: **No**
- Calling Party Address: **Yes**
- Called Party Address: **Yes**
- SDP Session Description: **No**
- Cause Code: **No**

The screenshot shows a window titled "Edit Diameter Profile" with a tabbed interface. The "OCF AVPs" tab is selected. The following table represents the configuration options visible in the screenshot:

Field Name	Yes	No
Destination-Host:	<input checked="" type="radio"/>	<input type="radio"/>
Service-Context-Id:	<input type="radio"/>	<input checked="" type="radio"/>
Subscription-Id-Data:	<input type="radio"/>	<input checked="" type="radio"/>
Subscription-Id-Type:	<input type="radio"/>	<input checked="" type="radio"/>
From-Address:	<input checked="" type="radio"/>	<input type="radio"/>
Sip-Method:	<input checked="" type="radio"/>	<input type="radio"/>
Originating-ioi:	<input checked="" type="radio"/>	<input type="radio"/>
Terminating-ioi:	<input checked="" type="radio"/>	<input type="radio"/>
Sdp-Media-Description:	<input type="radio"/>	<input checked="" type="radio"/>
Sdp-Media-Name:	<input type="radio"/>	<input checked="" type="radio"/>
Sdp-Type:	<input type="radio"/>	<input checked="" type="radio"/>
Event-Timestamp:	<input type="radio"/>	<input checked="" type="radio"/>
Calling-Party-address:	<input checked="" type="radio"/>	<input type="radio"/>
Called-Party-Address:	<input checked="" type="radio"/>	<input type="radio"/>
Sdp-Session-Description:	<input type="radio"/>	<input checked="" type="radio"/>
Cause-Code:	<input type="radio"/>	<input checked="" type="radio"/>

12. Move to the **PCRF** tab.

The screenshot shows the 'Edit Diameter Profile' dialog box with the 'PCRF' tab selected. The fields are as follows:

- Primary IP Address: IPv4 (dropdown), [text box]
- Primary Destination Port: 3868 (text box)
- Primary Realm Name: [text box]
- Primary Host Name: [text box]
- Secondary IP Address: IPv4 (dropdown), [text box]
- Secondary Destination Port: [text box]
- Secondary Realm Name: [text box]
- Secondary Host Name: [text box]
- Response-Waiting-time(Tx): 10 (text box)

Buttons: Save, Cancel

13. Edit the parameters.

14. Click **Save**.

## 8.9 Port Allocation Table

Dynamic pinholes are primarily used to allow traffic from a specified remote entity.

BorderNet SBC automatically allocates dynamic ports for signaling without operator intervention. The operator can configure dynamic port ranges for media traffic on a Peer or Interface using the **Port Allocation Table** assigned to **Media Profile**.

The **Port Allocation Table** allocates local media ports mapping for BorderNet SBC sessions. Ports are opened and closed on an as-needed basis to create dynamic pinholes, in the defined range.

For a media session, the media information is exchanged (**IP, Port, Codec properties**), and pinholes are opened to allow the media. When the session terminates, the pinholes will be closed.

→ To configure dynamic pinholes:

1. Create a port allocation table entry (range [7000 - 65535]) shown below.
2. In the **SIP Configuration Media Profile** window, select the created port allocation table entry.
3. Assign it to the appropriate media profile.

→ To create a Port Allocation Table entry:

1. Select **Application** → **Common** → **Port Allocation Table** to open the **Port Allocation** window:

Port Allocation							
<a href="#">+ Add New Port Allocation</a>							
Name	Application	IP Address Type	IP Interface	Vlan Name	Start Port	End Port	
Media_40_45	IPv4	Media	10.10.40.45	Vlan_40	7000	65535	
Media_30_45	IPv4	Media	10.10.30.45	Vlan_30	7000	65535	
Media_30_27	IPv4	Media	10.10.30.27	Vlan_30	7000	65535	
Media_40_27	IPv4	Media	10.10.40.27	Vlan_40	7000	65535	
Media_40_26	IPv4	Media	10.10.40.26	Vlan_40	7000	65535	
Media_30_26	IPv4	Media	10.10.30.26	Vlan_30	7000	65535	

2. Click the **+Add New Port Allocation** button.

The **Add Port Allocation** window opens:

**Add Port Allocation** ✕

Name:

Application:  ▼

IP Interface:   ▼

Start Range:

End Range:

*Note: A single call with a single media session uses 4 ports.*

3. Enter the following parameters:

- **Name.** The name of the allocation.
- **Application.**
- Possible values are:
- **Media.** Port allocation is used for all kind of media.
- **RegAddrMap.** Port allocation is used for **Registration Address Mapping (RegAddrMap)**. Ports are released when the port allocation is not in use. The IP and the port range are associated with only a single listening IP and port on either the **Access-Local** or **Access-Interconnect SIP Interface**, applicable only for IPv4 and UDP.
- **IP Interface.**
- Possible values: **IPv4/IPv6**.
- **Start Range.** The first port of the allocation (default **7000**).
- **End Range.** The last port of the allocation (default **65535**).

4. Click **Save**.

## 8.10 SIP Profilers

### 8.10.1 Overview

The BorderNet SBC SIP Profiler enables SIP headers and parameters manipulation for both the incoming and outgoing SIP messages, including the following SIP header operations:

- Modifying existing headers

- Deleting existing headers
- Adding new headers
- Storing header/parameter values in a variable
- Rejecting SIP Messages with custom warning codes

The Profiler applies the header manipulations when the message is received (incoming), and before the message is transmitted (outgoing).

Any SIP header field can be manipulated, such as a header value, header parameter and URI parameter, enabling interworking between SIP networks.

SIP headers and parameters are stored on the SIP Profiler and the rules impact on on the following levels:

- Local. Rules impact the SIP message (for example INVITE).
- Transaction. Rules from a SIP message are stored and impact another message in the same transaction (for example storing content from an INVITE and impacting a 200 OK to that INVITE).
- Session. The impact applies beyond a particular transaction (for example storing content from an INVITE and impacting the BYE message).

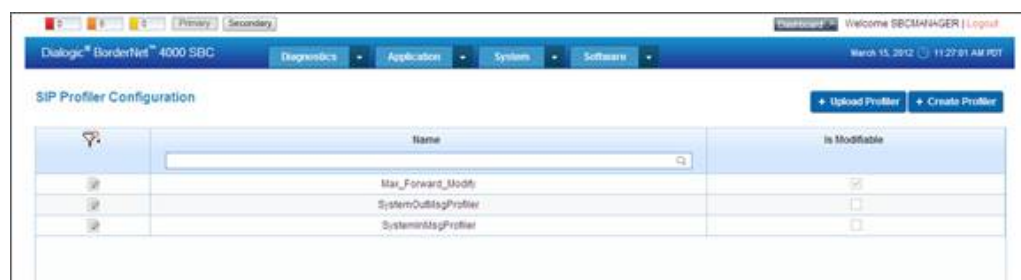
SIP Profilers are highly programmable but require caution and knowledge when creating and implementing rules. Only an Application Administrator can create, modify and delete SIP Profilers.

## 8.10.2 Conventions

→ To view the SIP Profiler Configuration menu:

1. Select Application → Common → SIP Profilers.

The SIP Profiler Configuration window opens.



The following icons facilitate **SIP Profiler** creation:

- The Add and Remove icons add or remove conditions or groups.

---

### Note:

Conditions within a Profiler group are either "and" conditions or "or" conditions. There cannot be "and" and "or" conditions within the same Profiler group.

---

- Gear icons are located next to some fields and indicate attributes to be added to the corresponding field.
- The Action List icon displays a drop-down menu when selected. The menu provides options for the selected SIP Profiler, such as Edit, +Rule and Delete.
- A red asterisk \* indicates a required field.

## 8.10.3 Creating a SIP Profiler

The following example shows how to create a Profiler that sets the "Max-Forwards" header with 85 in the INVITE message.

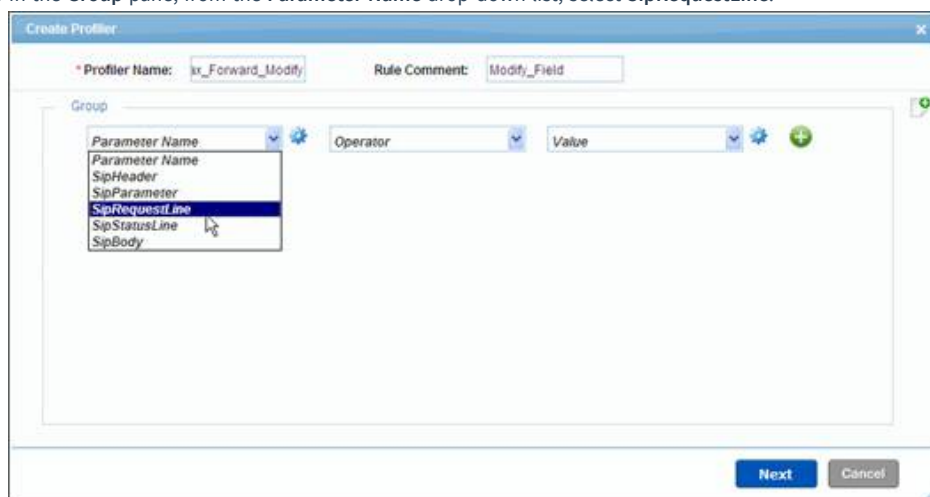
1. In **Application** → **Common** → **SIP Profilers**.
2. Click on the **Create Profiler** button.  
The **Create Profiler** window opens.
3. Enter a **Name** and **Rule Comment** for the new Profiler.



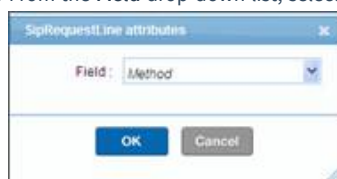
### Note:

The Profiler Name must be unique.

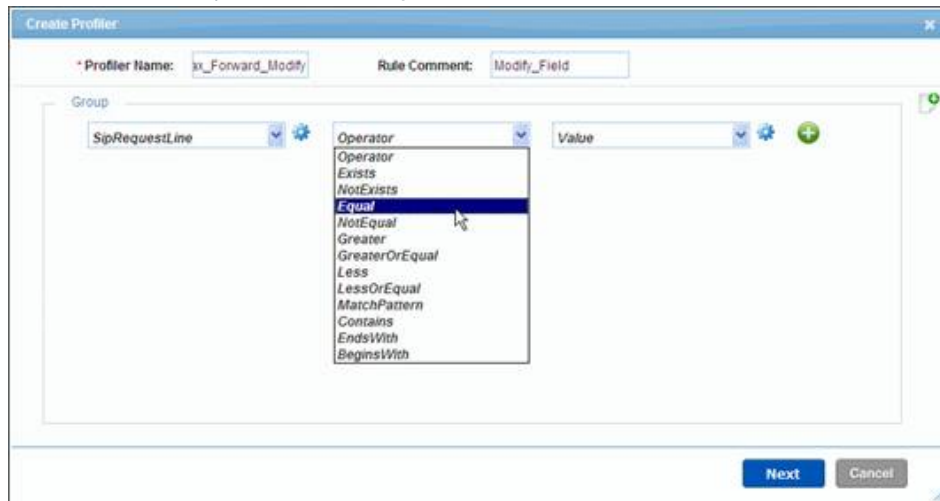
4. Set the **SIP Profiler** condition.
5. In the **Group** pane, from the **Parameter Name** drop-down list, select **SipRequestLine**.



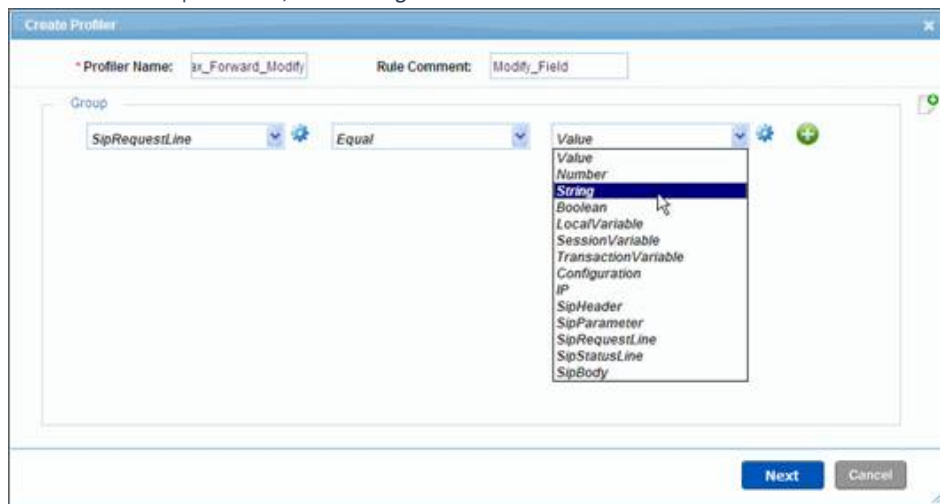
6. Select the **Gear** icon next to the **Parameter Name** field to open the **SipRequestLine Attributes** window.
7. From the **Field** drop-down list, select **Method** and click **OK**.




- From the **Operator** drop-down list, select **Equal**.



- From the **Value** drop-down list, select **String**.

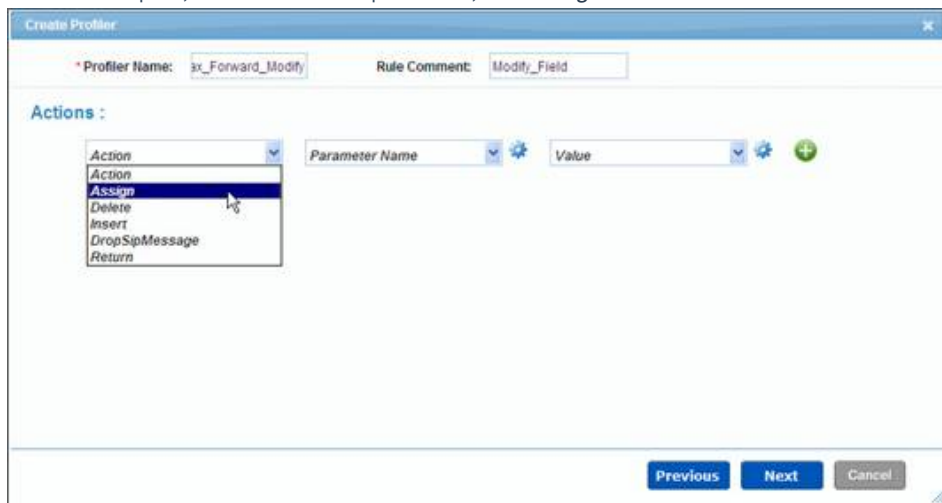


- Select the **Gear**  icon next to the **Value** field to open the **String Attributes** window.
- In the **Value** field, enter **INVITE** and click **OK**.

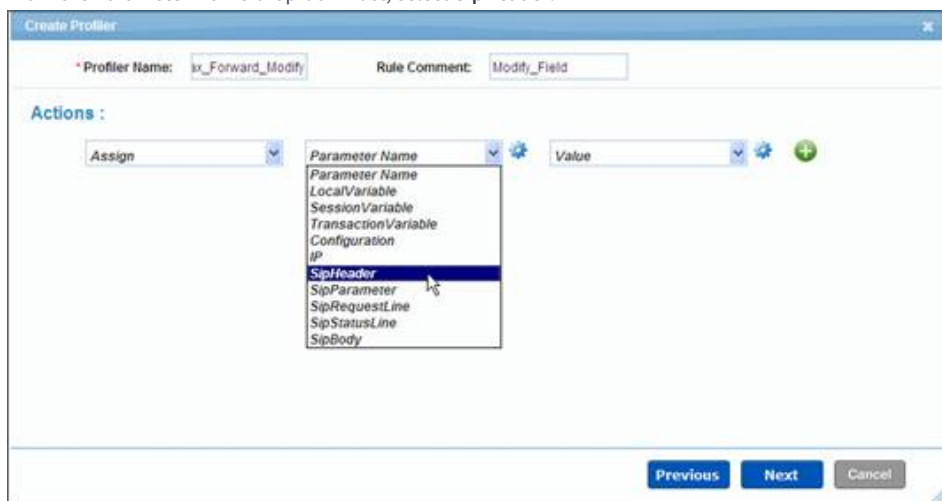



- Click **Next**.
- Set the **SIP Profiler** action.

6. In the **Actions** pane, from the **Action** drop-down list, select **Assign**.



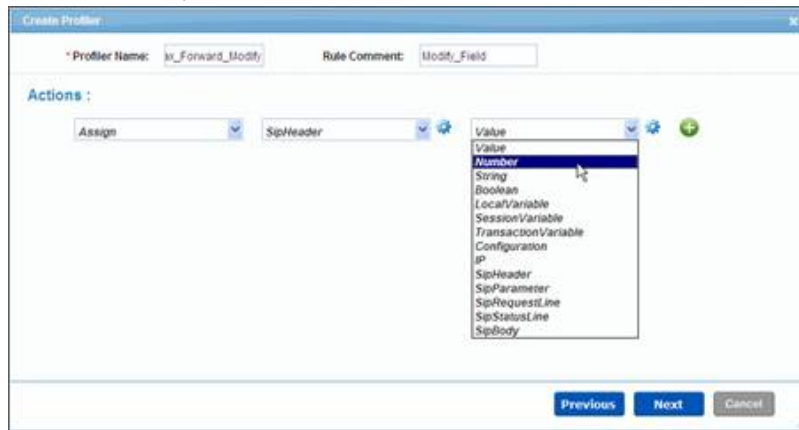
1. From the **Parameter Name** drop-down list, select **SipHeader**.



2. Select the  icon next to the **Parameter Name** field to open the **SipHeader Attributes** window.
3. In the **Header** field, select **Max-Forwards** from the drop-down list and click **OK**.



1. From the **Value** drop-down list, select **Number**.

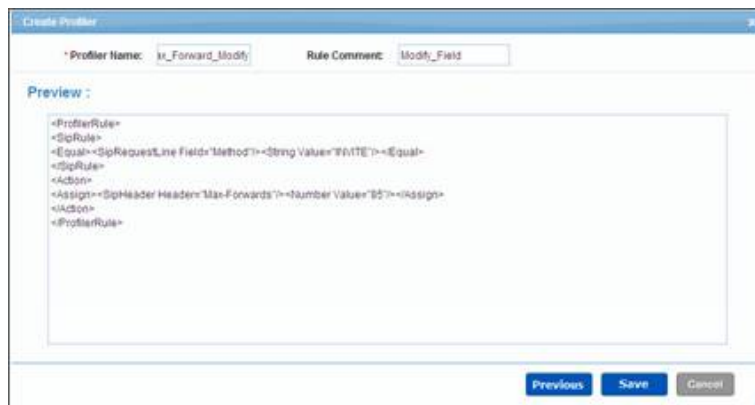


2. Select the **Gear** icon next to the **Value** field to open the **Number Attributes** window.
3. Enter **85** in the **Value** field and click **OK**.



4. Click **Next**.

The **Preview** window shows the **SIP Profiler**.



5. Click **Save**.

## 8.10.4 Editing SIP Profilers

→ To edit an existing SIP Profiler:

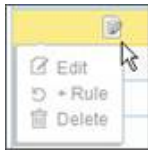
1. In **Application** → **Common** → **SIP Profilers**, select a **SIP Profile**.  
The checkmark in the **Is Modifiable** column indicates which profilers can be edited.
2. Click on the **Edit** icon.

## 8.10.5 Adding Rules



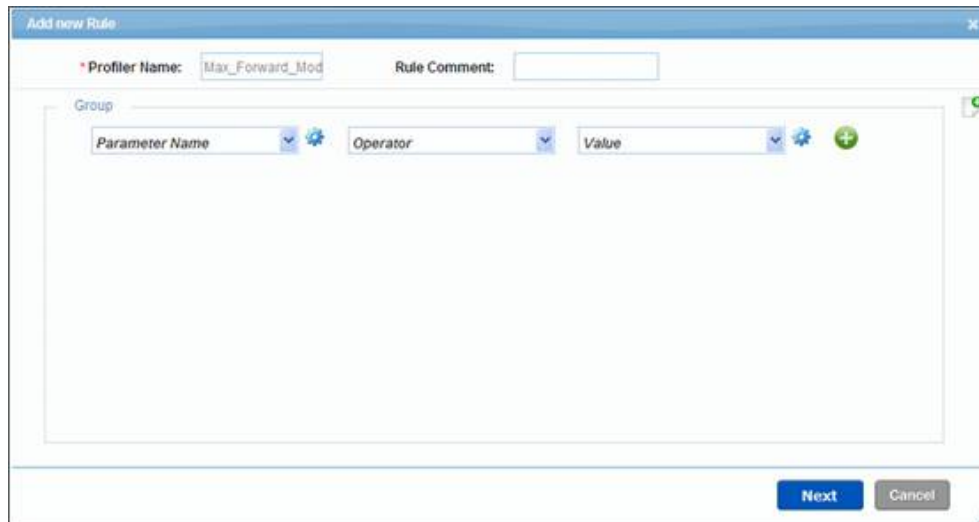
→ To add Rule+Action to an existing SIP profiler:

1. Select **+Rule** from the **Notepad** icon drop-down list, as shown below.



The **Add New Rule** window opens.

2. The **Rule ID** needs to be changed because it must be unique, and then additional rules and associated sets of actions can be added in the same manner they were created.



Once the profiler is created and saved, the new set of **Rule+Action** is appended below the original **Rule+Action**. This allows for the existence of multiple sets of **Rule+Actions** to coexist in the same profiler.

The **Return** statement allows to exit from the existing profiler after implementing all actions in the first set of **Rule+Actions** once the conditions have passed, and without executing any other **Rule+Action** sets that follow it. This is useful if you have to touch multiple messages with varied conditions and implement certain actions only if those conditions are met.

If the **Return** statement is not used, the sets of profilers are executed one after another, as shown in the following example:

```
<SipRule>
```

```
<Conditions>
```

```
</SipRule>
```

```
<Action>
```

```
<Action 1/>
```

```
<Action 2/>
```

```
<Return/>
```

```
</Action>
```

```
<SipRule>
```

```
<Conditions>
```

```
</SipRule>
```

<Action>

<Action 1/>

<Action 2/>

</Action>

---

**Note:**

If the <Return> statement is used, and the first set of conditions pass, the profiler will exit without executing the second set of Rule+Action. When the <Return> statement is not used, then both sets of Rule+Action will be implemented sequentially.

---

## 8.10.6 Creating a SIP Profiler with XML Files

For advanced users, a SIP Profiler can be written in XML and loaded directly from the user's system.

Once loaded, the XML file can be selected via a drop-down menu in the **SIP Profiler Configuration** window and applied to an incoming or outgoing SIP message on a SIP interface. XML syntax is automatically validated when a document is uploaded to the SCS using **XSD (XML Schema Definition language)**.

→ To upload an XML file:

1. Select **Application**→**Common**→**Profilers** to open the **SIP Profiler Configuration** window.
2. Click the **Upload Profiler** button.

The **Add SIP Profiler** window opens.

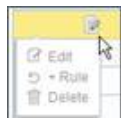


3. Click the **Browse** button to select an XML file to upload.
4. Name the file and click **Save**.

## 8.10.7 Deleting SIP Profilers

→ To delete an existing SIP Profiler:

1. In **Application**→**Common**→**SIP Profilers**, select a **SIP Profile**.
2. Click on the **Delete** icon, from the **Action List** icon drop-down menu.



3. Click **Confirm** in the dialog box to delete the SIP Profiler.



## 8.10.8 Profiler Document Hierarchy

The SIP Profiler consists of one or more SIP Profiler documents, which are individual XML files.

Each XML file contains rules with associated actions that are used to manipulate SIP headers and parameters. Multiple files can be linked together logically for enhanced organizational benefit and high efficiency.

For example, one XML file can be designed as a common building block that can be written once and called over and over on different SIP interfaces as part of more complex header manipulations that may vary only slightly from one another.

When a SIP message enters on a SIP interface with a configured SIP Profiler document, the Profiler execution begins at the invocation of the configured XML document called the root document. The outgoing SIP message on the same interface may have a different XML document assigned to complete the desired header manipulation.

More complex SIP Profiles may be defined by nesting multiple XML files in a pre-defined order to the root document. Each nested XML documents is executed in order.

The Profiler execution begins at the invocation of the root document associated with the SIP interface. During root document execution, the root document may call other SIP Profiler XML documents, shown below.

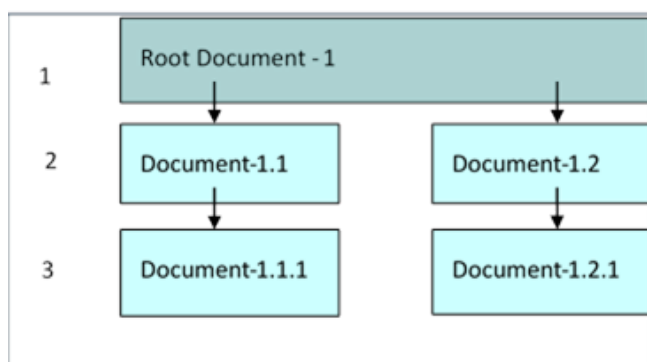


Figure 13: SIP Profiler Nested Documents

Any number of SIP Profiler XML documents can be called during Profiler execution as long as the total hierarchy levels do not exceed 5. The above figure illustrates a SIP Profiler that executes 5 XML files within 3 hierarchy levels.

The order of execution begins with **Root Document-1**, moves to **Document-1.1**, and then **Document-1.1.1** before returning to **Root Document-1** and then traversing **Document-1.2** and **Document 1.2.1**. Profiler execution is explained further in [Profiler Document Structure](#).

## 8.10.9 Profiler Document Structure

Each SIP Profiler XML document consists of one or more '**ProfilerRule**' elements. Each '**ProfilerRule**' element consists of a '**SipRule**' and '**Action**' element and is of the form:

**If** (Condition) **Then** (Statement).

There is no limit to the number of elements that may be contained within one SIP Profiler XML document.

The '**SipRule**' element contains the conditions and the '**Action**' element contains the statements to be executed if all conditions inside the '**SipRule**' element return "**true**".

See [Group Elements](#) for an XML syntax example.

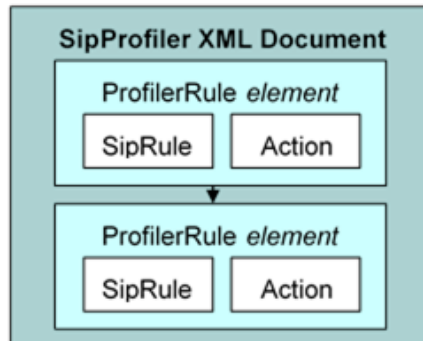


Figure 14: SIP Profiler Document Structure

Profiler execution begins with the execution of the first 'ProfilerRule' element in the root document and continues execution from this rule to the next rule until one of the following conditions is met:

- All the 'ProfilerRule' elements in the document are executed.
- An 'Action' element states to stop execution.
- A SIP message is rejected by the Profiler.

The Profiler enables an operator to reject a SIP message based on precisely defined criteria. For example, a call may be released early with a specific release code returned to the customer if a predefined mandatory SIP parameter is missing from an incoming message.

An 'Action' element inside a 'ProfilerRule' element may state to jump to another XML document, shown below. In that scenario all 'ProfilerRule' elements in the child XML document are executed before the rest of the 'ProfilerRule' elements are executed in the parent XML document.

Figure 15: Rule Processing Flow in Nested SIP Profiler Documents

## 8.11 ISUP Profilers

The **SIP Profiler** strips the encapsulated body of the SIP-I and executes the ISUP Profilers.

The **ISUP Profiler** allows the operator to add, delete, or modify any parameter of encapsulated ISUP message.

The **ISUP Profiler** executes parameter modification rules based on the message type, using XML rule files.

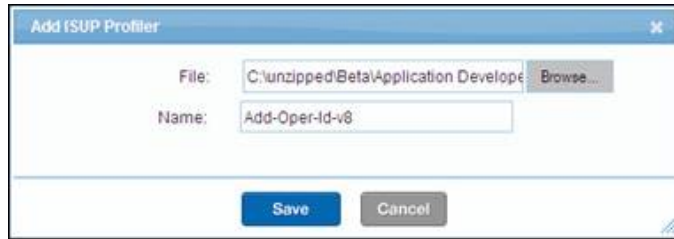
→ To upload an ISUP profiler:

1. Select **Application** → **Common** → **ISUP Profilers**, to open the **ISUP Profiler Configuration** window.



2. Click **Upload Profiler**.

The **Add ISUP Profiler** window opens.



3. Use the **Browse** button to select the desired **ISUP Profiler** file.
4. Enter the **Name** of the **ISUP Profiler**. The name must be unique to the BorderNet SBC.
5. Click **Save**.

## 8.12 Transcoding Gateways

Not relevant for this release.

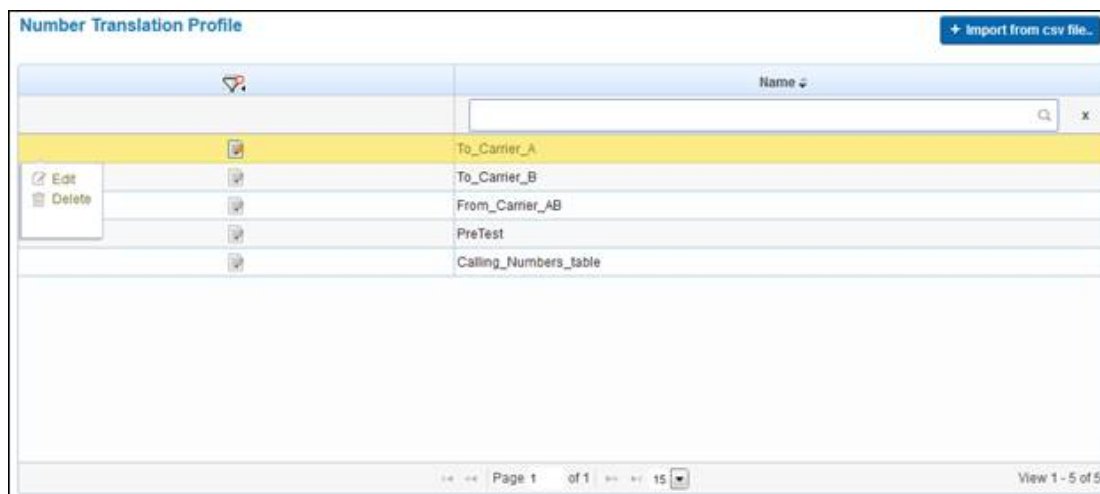
## 8.13 Number Translation Profile

Number translation profiles hold the translation tables which are used for searching and substituting the calling or called numbers.

→ To provision a Number Translation Profile:

1. Select **Application** → **Common** → **Translation profile**.

The **Number Translation Profile** window opens.



2. Select **+ Import from CSV file** to load a new translation table.
3. A new **Add Number Translation Profile** dialog will be displayed.



- 4. Enter a name for the new profile to be added.
- 5. Select a file to be loaded.

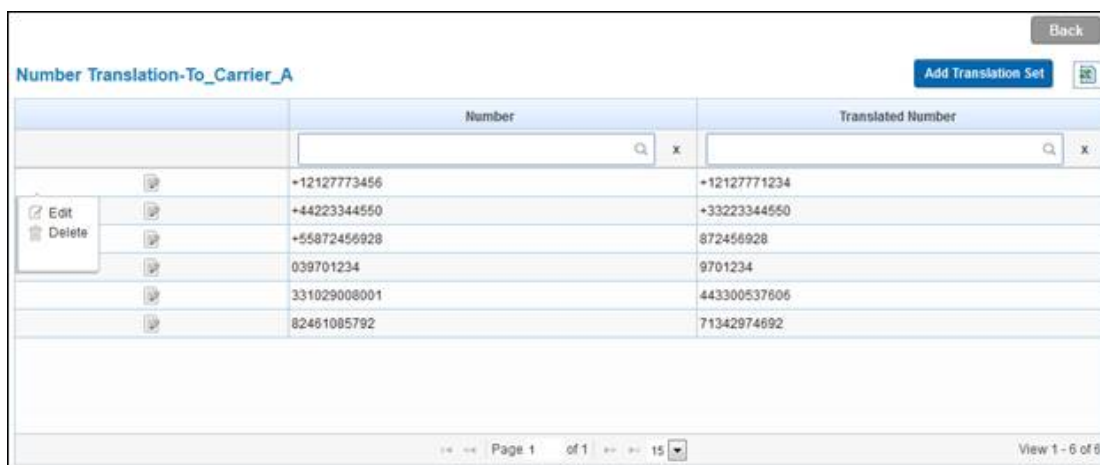
The translation table file should be in a CSV format, containing two columns separated by a single comma. The first column is the searched number and the second column is the translated number.

See the following printout of an example CSV file:

1	+12127773456,+12127771234
2	+44223344550,+33223344550
3	331029008001,443300537606
4	039701234,9701234
5	+55872456928,872456928
6	82461085792,71342974692

Each file/table can contain up to 200,000 records (up to 200,000 lines).

- 6. To delete an existing table, select **Delete** from the left column configuration button.
- 7. To edit/view an existing table:
  - Select **Edit** from the left column configuration button.
  - OR
  - Double click a desired profile line.
- 8. A new configuration window containing all the table's numbers will be displayed.



- 9. Select **Back** to return to the main **Number Translation Profile**.
- 10. Select **Add Translation Set** to add a new single line to the translation table.
- 11. Select **Export CSV** to export the translation table to a CSV file.

12. To delete a single existing record, select **Delete** from the left column configuration button.

13. To edit/view an existing record:

- Select **Edit** from the left column configuration button.
- OR
- Double click a desired profile line.

The BorderNet SBC uses the following logic to find and replace the calling/called numbers.

- CalledPartyUserId
  - The searched number is extracted from the user part of the **Request-URI** (request-line)
  - The headers to be modified if the number is translated include the **Request-URI** and the **TO** header.
- CallingPartyUserId
  - The searched number is extracted from the user part of the **FROM** header.
  - The headers to be modified include only the **FROM** header.
  - In case a **P-Asserted-ID** exists, it will override the manipulation performed on the **FROM** header. This means that the modification of the translated number in the **FROM** header will be lost, and the **P-Asserted-ID** will be copied to the **FROM** header.
- FROM
  - The searched number is extracted from the user part of the **FROM** header.
  - The headers to be modified include only the **FROM** header.
  - In case a **P-Asserted-ID** exists, the modification to the **FROM** header is still preserved. This means that the **FROM** header will contain the translated number.

## 8.14 Bulk Provisioning

You can load a pre-prepared Excel file, populated with all the relevant parameter values, per entity type, to save a time-consuming one-by-one configuration.

→ To configure bulk provisioning:

1. Select **Application** → **Common** → **Bulk Provisioning**.

The **Bulk Provisioning Configuration** window opens.

**Bulk Provisioning Configuration**

Bulk Provisioning Status:

Entity Type:

Template:

```
Add Interface
Id, Status, Name, Domain, Network Type, SIPconnect, SIPconnect Type, IMS, Subscriber
Traffic, Access Type, Transport IP Address Type, Signaling IP, VLAN Name, Signaling
Port, Signaling Protocol, Max Allowed UDP MTU, Signaling TOS, RegPortReuse, Port
Allocation, TGRP Context, Enforce ipsec, Parameter Profile, Media Profile, Service
Profile, Security Profile, Associated Peers, Trust Level, Local Operator Id, TLS Profile, SRTP
Profile, Time Zone
```

+ Upload Export Cancel

## 2. Enter **Entity Type**.

This refers to the item to be configured. By selecting this item the **Template** field is automatically populated.

When the *Advanced Policy* entity type is selected, you are required to enter the *Advanced Policy Name*.

If surrogate registration is enabled, for the *Peer* Entity Type the list of parameters on the template text box includes the surrogate peer parameters at the end of the line.

For more details, see *BorderNet SBC Bulk Provisioning User's Guide*.



The screenshot shows a dialog box titled "Bulk Provisioning Configuration". It contains a "Bulk Provisioning Status:" field at the top. Below that, there are two dropdown menus: "Entity Type:" with "Advanced Policy" selected, and "Advanced Policy Name:" with "demo" selected. A text area below these fields contains the prompt "Please select the policy name." At the bottom of the dialog, there are three buttons: "+ Upload", "Export", and "Cancel".

## 3. Click on the **+Upload** button to select the Excel files.

This includes the parameters rows based on the columns introduced in the template field, and to import the parameters.

## 4. To export the configured parameters to a CSV file, click the **Export** button.

# 8.15 Customized LRBT

Prior uploading the **Local Ring Back Tones (LRBT)** from your local terminal to the BorderNet SBC, see the section *Generate Local Ring Back Tone Packages* in the *BorderNet SBC Maintenance User's Guide* document.

To enable this capability and to provision its related parameters see [LRBT](#).

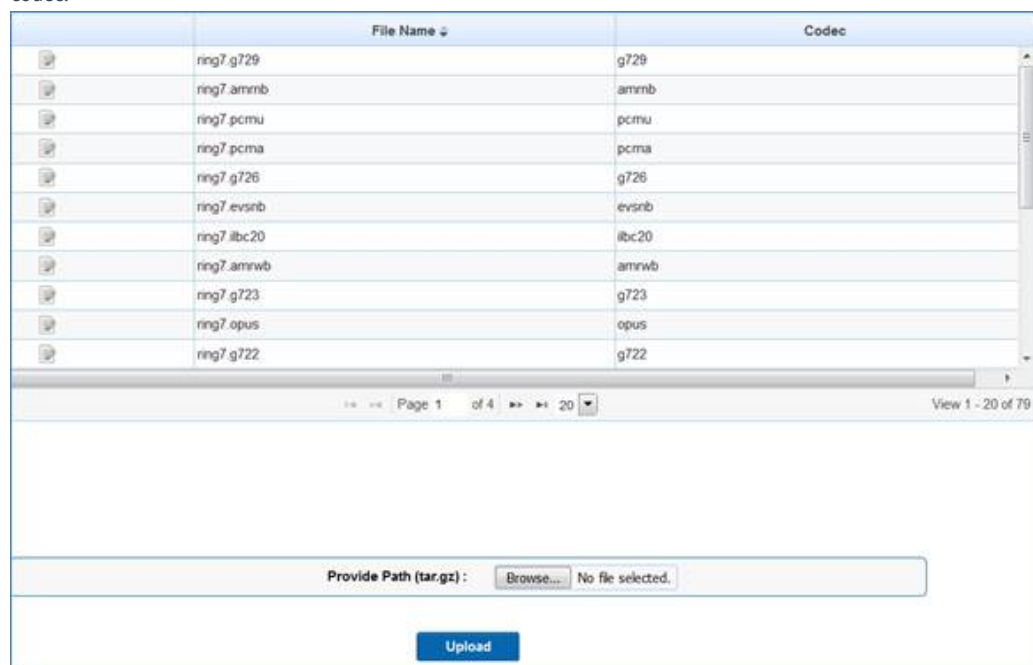
→ To upload an LRBT file:

### 1. Select **System** → **Administration** → **Customized LRBT**.












The **LRBT** window opens, presenting the list of the LRBT files, the time they have been uploaded to the system, and the used



codec.



The screenshot shows a web interface with a table of codecs and a file upload section. The table has two columns: 'File Name' and 'Codec'. Below the table is a pagination bar showing 'Page 1 of 4' and 'View 1 - 20 of 79'. At the bottom, there is a 'Provide Path (tar.gz):' field with a 'Browse...' button and 'No file selected.' text, and an 'Upload' button.

	File Name	Codec
	ring7.g729	g729
	ring7.ammb	ammb
	ring7.pcmu	pcmu
	ring7.pcma	pcma
	ring7.g726	g726
	ring7.evsnb	evsnb
	ring7.ilbc20	ilbc20
	ring7.amrwb	amrwb
	ring7.g723	g723
	ring7.opus	opus
	ring7.g722	g722

Page 1 of 4 View 1 - 20 of 79

Provide Path (tar.gz):  No file selected.

2. In the **Provide Path** field, select an LRBT file from the local terminal using the Browse facility.
3. Click **Upload**.

This enables the selected file's upload from the local terminal to the BorderNet SBC.

---

**Note:**

The Edit icon at the left side of the *Time* column, enables a selected LRBT file's removal from the system.

---

## 9. Definitions

### 9.1 Trunk Groups

A **SIP Trunk Group** is a set of destinations (IP addresses, ports, and transport types) that can be used to reach the same endpoint.

The following Trunk Group scenarios are supported:

- Pass-through SIP Trunk Group information without taking any action on the received information.
- Originating SIP Trunk Group information insertion for use by the Dialogic ControlSwitch™ System or another switch later in the call.
- Destination SIP Trunk Group information insertion returned by the PE for use by the CS System or another switch later in the call.
- Receive SIP 3XX Trunk Group information and redirecting calls passing on this Trunk Group information.
- Receive either a SIP INVITE or 3XX, consuming Trunk Group information terminating at the BorderNet SBC, and selecting the appropriate SIP or H.323 outgoing routes.

BorderNet SBC supports RFC 4904. The term "**TGRP**" identifies a string that represents a set of Trunk Groups that may be defined on the BorderNet SBC or on another switch. The BorderNet SBC also supports the proprietary terms "**otg**" (**originating trunk group**) and "**dtg**" (**destination trunk group**).

### 9.2 Network Types

BorderNet SBC supports access networks, providing three access network types while creating SIP-based interfaces, peers and profiles:

- Access-Public
- Access-Interconnect
- Access-Local

For Interconnect networks (such as a service provider network's CS, peering to another service provider):

- Interconnect indicates a public network.
- Local indicates a private network.

Access networks are defined in the following table.

Network Interface	Description	Usage
Access-Public	A public network interface type created towards the User Equipment (UE).	The Access-Public type interface is used for SIP communication towards the UE. The Access-Public interface address is advertised to the UE as an outbound proxy.
Access-Local	A network interface type created towards the network that provides access services. Access-Local indicates a home access network.	The Access-Local type interface is used for SIP communication towards a trusted home access network. Access-Local should be chosen for trusted servers.

Network Interface	Description	Usage
Access-Interconnect	A network interface type created towards the network that provides access services. Access-Interconnect indicates a visiting access network.	The Access-Interconnect type interface is created for SIP communication towards a visiting network on which Session Border Control functionality is desired. If Access-Interconnect is selected, the ControlSwitch System will disable topology and dialog transparency for traffic or traffic from Access-Public to Access-Interconnect interfaces.

Table 14: Access Network Types

The distinction between the **Access-Local** and **Access-Interconnect** interfaces is the dynamic transparency determination rules:

- The Topology and Dialog transparency are disabled for traffic from Access-Public to Access-Interconnect interfaces.
- The Topology and Dialog transparency are enabled for traffic from Access-Public to Access-Local interfaces.

---

**Note:**

Only the Peers and Interfaces with the same Network Type can be associated.

---

A **Network Property** field is included in all SIP-based interface, peers and profiles.

Check the **Subscriber Traffic** box to enable the interface to handle SIP and other messages related to the User Equipment (examples are REGISTER, SUBSCRIBE, NOTIFY, etc.).

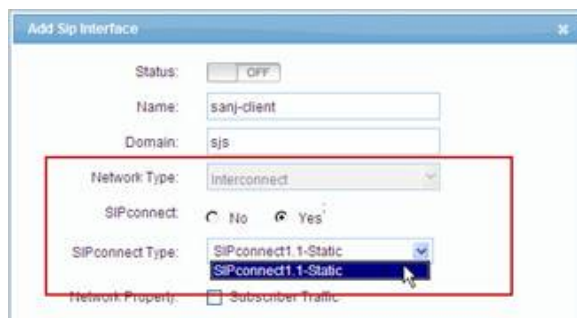
The **Network Property** default values depend on the type of network selected.

- For an Interconnect or Local Network Type, the default value for Subscriber Traffic is OFF (unchecked box).
- For an Access Network Type, the default value for Subscriber Traffic is ON (checked box).

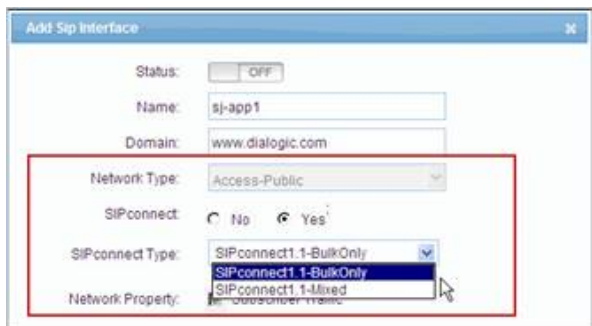
## 9.3 SIP Connect

If **Interconnect** or **Access-Public** network types are selected, then **SIPconnect** and **SIPconnect Type** fields shall be displayed and can be configured.

- **Interconnect.** If SIPconnect is set to Yes, the SIPconnect Type is SIPconnect1.1-Static. In this mode, the BorderNet SBC determines the SIP-PBX signaling address using statically configured data or the DNS.



- **Access-Public.** If SIPconnect is set to Yes, the SIPconnect Type options are:
  - **SIPconnect1.1-BulkOnly.** In this option, the **Bulk Number Contact (BNC)** notation is enforced per RFC 6140. Even individual registration appearances (such as in the Contact indicated in the example below) are treated as bulk mode.
  - **SIPconnect1.1-Mixed.** The standard mode when the BorderNet SBC supports both individual and bulk registrations indicated in RFC 6140.



Example, consider the following REGISTER:

*REGISTER sip:sse5.sipconnectit.com SIP/2.0*

*From: <sip:+18881005000@sse5.sipconnectit.com>;tag=IXh88gxBG0vOGJBws2AEFF1C7f960c5*

*To: <sip:+18881005000@sse5.sipconnectit.com;transport=UDP>*

*Call-ID: 1102318852@132.177.127.111*

*CSeq: 881 REGISTER*

*Via: SIP/2.0/UDP 132.177.127.101:5060;branch=z9hG4bK-15-536b-1f1fd203-7f7331836e80*

*Contact: <sip:+18881005000@132.177.127.101:5060;transport=UDP>*

*Via: SIP/2.0/TLS 132.177.127.111:5060;branch=z9hG4bK928121122*

*Max-Forwards: 69*

*Expires: 60*

*Content-Length: 0*

As per RFC 6140, this REGISTER is a registration of a single E.164 number and, therefore, a device behind the SIP PBX. This is indicated by the user part in the Contact URI.

However, per the current practice (and witnessed in interoperability cases), the identity (in TO) '5000@sse5' is considered as a PBX identity by the SIP-AS, and the user part in the Contact is not considered as a subscriber.

The '5000@sse5' is configured with a bulk of numbers, such as 18881005001 to 18881005015. The Contact is implicitly assumed as though with BNC, and its IP and Port are used for these bulk numbers. In this instance, BorderNet SBC has a situation where it has to treat the notation as bulk registration.

The **SIPconnect1.1-BulkOnly** option is used to treat such REGISTERS as bulk.

For the configuration of **SIPconnect1.1-Mixed** mode, the BorderNet SBC receives this SIP REGISTER and treats it as an individual registration.

---

**Note:**

When the BorderNet SBC processes the SIP register over the Access-Public interface, cache entries are created that correspond to the SIP-PBX identity that issued the entry. BorderNet SBC inserts the cache identity in the forwarded Register and also retrieves the cache that comes in the future INVITE from within the SP Network. This is a standard Access mechanism that is also utilized for SIPconnect Register mode.

---

## 9.4 Transport Protocol

Network Type	SIP Connect	Transport Protocol Options
<b>Interconnect</b> or <b>Access-Public</b>	Yes	<b>UDP-TCP, TLS</b>
<b>Interconnect</b> or <b>Access-Public</b>	No	UDP, UDP-TCP
Access-Interconnect		UDP, UDP-TCP, TLS
Access-Local		UDP, UDP-TCP, TLS

### 9.4.1 Supported Configurations of Transport Interworking

The supported configurations of transport interworking for interfaces involving **Access-Public** and **Access-Interconnect** are listed in the following table.

The supported configurations are independent of one another, as **Access-Public** and **Access-Interconnect** do not interwork directly.

Method	Transport with SIP-PBX	Transport with SP-Network
REGISTER/INVITE	TLS	TLS[1]
REGISTER/INVITE	TLS	TCP[2]
REGISTER/INVITE	TLS	UDP
REGISTER/INVITE	UDP	UDP
REGISTER/INVITE	TCP	TCP

### 9.4.2 UDP to TCP Automatic Transition

The **SIP UDP to TCP** transition feature enables BorderNet SBC to switch from UDP to TCP when packets require fragmentation. TCP provides transport-layer fragmentation while UDP has no such message fragmentation capability. When it is used, the fragmentation occurs at the IP layer instead. It is used because it is faster than TCP and its overhead is smaller and lighter.

TCP uses the sliding window mechanism for flow control. It adjusts the permitted window size in accordance with the underlying MTU and congestion control. For large SIP messages which are greater than the default value of 1500 bytes, the **RFC-3261** standard mandates the usage of a congestion control transport layer so that the message will pass successfully. Transition from UDP to TCP will therefore be applied specifically for larger messages.

## 9.5 SIP-I Support

BorderNet SBC supports sending and receiving encapsulated ISUP message in SIP messages.

BorderNet SBC encodes and decodes the encapsulated ISUP messages using the ETSI variant. The ISUP body is handled as part of the multi-part or as a single body in the SIP message. The ISUP body can also be handled with trailing `/r/n` or without trailing `/r/n`.

---

#### Note:

BorderNet SBC complies with the Profile-C requirements in the ITU spec Q.1912.5 in cases of encapsulated message available in the SIP Message.

---

The following combinations for ingress and egress SIP messages are supported:

Ingress Protocol	Ingress - ISUP Body	Egress Protocol	Egress - ISUP Body
SIP	Present	SIP	Present
SIP	NA	SIP	Present
SIP	Present	SIP	NA
SIP	NA	SIP	NA

**SIP-I** is a licensed feature.

**SIP-I** calls are supported up to the maximum license limit.

On unlicensed systems, **SIP-I** messages pass transparently, unless the operator opts to reject or strip the message.

→ To configure the SIP-I Profiler:

1. Configure the **Service Profile**.
2. In **Service Profiles**, select the **SIP-I** tab.
3. Enter the desired values for the incoming and outgoing ISUP treatments.
4. Associate the service profile with an interface or a peer.
5. In the **Advanced Policy** window, create a routing policy.
6. Select the **Rule using Parameters** type, and **SIPICall** parameter.
7. Upload the ISUP Profiler (see **ISUP Profilers**).

## 9.6 Traffic Policing

BorderNet SBC employs traffic policing to throttle incoming packet traffic from allowed sources so that each source can only use the bandwidth allocated for it. This allows the system to protect itself from a misbehaving host or flooding attacks like ICMP floods, UDP floods, TCP SYN floods, and so forth from a spoofed address.

Traffic policing prevents bandwidth theft or misuse for media traffic. For example, when a session is established, a codec is negotiated for that session. The codec has a pre-determined bandwidth, and any excess packets are dropped.

Traffic policing is based on the concept of flows, and the specified packet rate for a flow. A flow is a sequence of packets all sharing common properties, such as source IP, source port, destination IP, destination port, and transport. White list flows and gray list flows depend on the trust level of the peer.

- White list flows have assured bandwidth allocation and are used for peers with a higher trust level. Media flows are also a type of white list flows.
- Gray list flows initially provide limited bandwidth and are used with peers that have a lower trust level, (such as gateways behind a gatekeeper). Gray list flows can be promoted to white list flows once trust-like a successful call-is established.

The bandwidth allowed for media flows is determined based on the codec that is used for a given session. The bit-rate for the codecs is configured in the codec profile.

The BorderNet SBC dynamically determines the packet rates for signaling flows based on the session constraints defined in the security profile. The Application Administrator can set a fixed packet rate value by editing the security profile (shown below), and

this will override the dynamic packet rate adjustments.

Is Packet Rate Dynamic:  Yes  No  
 Packet Rate:

## 9.7 Surrogate Registration

Surrogate Registration is a SIP registration on behalf of the individual phones, behind an aggregation device (such as IP-PBX). This capability is available only for **Access-Public** interface types.

When the surrogate registration is enabled on an **Access-Public** interface, the BorderNet initiates a registration request on behalf of each SIP endpoint, configured in the associated surrogate peer.

→ To configure Surrogate Registration:

1. Select **Application** → **SIP Configuration** → **Peer** → **Add SIP Peer**.

The **Add SIP Peer** window opens.

2. Set the **Network Type** to **Access-Public**.

The following new parameters are displayed:

**Add Sip Peer**

Status:  ON

Name:

Class ID:

Network Type:

Network Property:  IMS

Surrogate Peer:  Surrogate-Peer

Peer AOR (user part):

Auth Username:

Auth Password:

Number List/Range:

Registration Type:  Regular  Bulk(RFC-6140)

- **Surrogate Peer.** If checked, the surrogate registration is enabled, and the following parameters (below) are displayed.
- **Peer AOR (user part).** The user part of the **To** header, sent from the peer (usually it is a phone number - string)
- **Auth. Username.** The username for the SIP authentication (if required by the registrar via a 401/407 response - string).
- **Auth. Password.** The password for the SIP authentication (if required by the registrar via a 401/407 response - string).
- **Number List/Range.** A list of numbers to be registered on behalf of the SIP peer (individual numbers separated by comma, or a range indicated by a hyphen).
- Registration Type.
- Possible values: **Regular/Bulk** (RFC-6140).

# 10. SIP Profiler Variables and Elements

## 10.1 SIP Profiler Variables

During header manipulation operations it may be advantageous to store certain header data from one operation and then call that data during another operation. The Profiler supports data value storage into three different types of temporary variables each with a different lifetime.

The following three types of variables are supported:

- Local variables
- Transaction variables
- Session variables

Variables have multiple properties, described in the following table.

Variable Property	Description
Value	A variable can be assigned an initial value using the <Assign> tag and can be re-assigned a data value any number of times using this same tag.
Scope	A child SIP Profiler document can access a variable defined by the parent document and vice versa.
Quantity	The Profiler supports a limited number of variables of each type. While storing a data value inside a variable, an index for that variable is assigned. An index number cannot exceed the maximum number of variables of each type and can be used to access the variable during its lifetime. See <b>Local Variables</b> for more details.
Name	A variable can be given a name that is used to access the variable value during the variable's lifetime. The variable name can be assigned in the same statement that is used to assign the variable value.

### 10.1.1 Examples

#### Local Variable

```
<Assign>
```

```
<!--Stores Sip Request Method name inside a Local variable stored at index 1, named Method -->
```

```
<LocalVariable name="Method" index="1"/>
```

```
<SipRequestLine Field="Method"/>
```

```
</Assign>
```

#### Session Variable

```
<Assign>
```

```
<!--Accesses local variable named Method, and stores it back in Session variable at index 1 -->
```

```
<SessionVariable index="1"/>
```

```
<LocalVariable index="1"/>
```

```
</Assign>
```



## 10.1.2 Local Variables

A local variable is accessible throughout one single Profiler execution. Once a SIP message leaves the Profiler (execution is complete) all local variables are destroyed.

There are a maximum of ten (10) local variables per Profiler execution.

## 10.1.3 Transaction Variables

A transaction variable is accessible throughout the entire SIP transaction lifetime. For example, a transaction variable created when an initial INVITE message is received can be accessed in 180 Ringing or 200 Ok for that INVITE.

A transaction variable created inside an incoming interface can be accessed from SIP Profiler assigned to outgoing interface for the same transaction.

There are a maximum of five (5) transaction variables per transaction.

## 10.1.4 Session Variables

A session variable is accessible throughout the entire SIP session lifetime. For example, a session variable created when an initial INVITE message is received can be accessed in 180 Ringing, 200 Ok, or BYE for that INVITE. Once the session is over all session variables are destroyed.

A session variable created inside an incoming interface can be accessed from SIP Profiler assigned to outgoing interface for the same sip session.

There are a maximum of five (5) session variables per SIP session.

---

**Note:**

One single Profiler execution is the timeline between execution of the first and the last 'ProfilerRule' element inside one SIP Profiler (including the root and all child SIPProfiler documents)

---

## 10.2 SIP Profiler Elements

This section describes the current SIP Profiler XML elements.

---

**Note:**

Element names are case sensitive. For example, 'SipHeader', 'sipheader', and 'SIPHEADER' are not same.

---

All elements in a SIP Profiler XML document can be broadly classified as one of the following:

- **Group Element**
  - Attributes: Groups elements together to create associations between them
  - Value Returned: None
- **Operator Element**
  - Attributes: Performs some type of operation on data. (e.g. 'And', 'Or', 'Concatenate', etc.)
  - Value Returned: The value returned depends on the type of operation performed. For example, an 'And' operation will return Boolean true or false while a 'Concatenate' operation will return a concatenated string.
- **Data Element**

- Attributes: References one of the following data values
- SIP data: SIP header/parameter values
- Configuration data: Data configured in SCS tables
- IP layer data: IP layer information e.g. incoming IP, port etc.
- Constant Data: String, Number, or Boolean constants
- Value Returned: The value returned depends on the data value being referenced by the element. For example, referencing an incoming IP address will return a string, while referencing an incoming port will return a number value.
- **Action Element**
  - Attributes: Performs an action such as
  - Rejecting a SIP message
  - Inserting/deleting a SIP header
  - Stop further execution of Profiler rules.

## 10.2.1 Group Elements

### 10.2.1.1 <SipProfiler>

Parent element: None

- 1st Child element: <Id>
- 2nd Child element: <Name>
- 3rd Child element: <ProfilerRule>

Attributes: None

**Usage Guidelines:** This is the root element of the SIP Profiler document. Other than the <Id> and <Name>, it contains only 'ProfilerRule' elements as its children.

At least one 'ProfilerRule' element should be present inside a 'SIPProfiler' element. There is no limit to the maximum number of 'ProfilerRule' child elements.

#### Example

```
<SipProfiler>
<Id>10</Id>
<Name>Example</Name>
<ProfilerRule id="Unique_Id_1" comment="Description of Rule 1">
<SipRule>
<!-- -->
</SipRule>
<Action>
<!-- -->
</Action>
</ProfilerRule>
```

```
<ProfilerRule id="Unique_Id_2" comment="Description of Rule 2">
```

```
<SipRule>
```

```
<!-- -- >
```

```
</SipRule>
```

```
<Action>
```

```
<!-- -- >
```

```
</Action>
```

```
</ProfilerRule>
```

```
</SipProfiler>
```

The example above shows one Profiler document that contains two 'ProfilerRule' elements.

### 10.2.1.2 <ProfilerRule>

Parent element: <SipProfiler>

- 1st Child element: <SipRule>
- 2nd Child element: <Action>

#### Attributes

Name	Required	Type	Description	Value Set	Default Value
Id	M	String	Unique id to identify the rule		
Comment	M	String	Description of rule		

**Usage Guidelines:** This is a container element for the 'SipRule' element and its corresponding 'Action' element. It contains exactly one Mandatory 'SipRule' and one Mandatory 'Action' element.

There is a one to one (1:1) relationship between a 'SipRule' and 'Action' element. Each SipRule has one corresponding Action element.

#### Example

```
<ProfilerRule id="Unique_Id_2" comment="Description of Rule 2">
```

```
<SipRule>
```

```
<!-- -- >
```

```
</SipRule>
```

```
<Action>
```

```
<!-- -- >
```

```
</Action>
```

```
</ProfilerRule>
```

The example above shows the structure of a single 'ProfilerRule' element. Each 'ProfilerRule' consists of one 'SipRule' and one 'Action' element.

### 10.2.1.3 <SipRule>

Parent element: <ProfilerRule>

Child elements: <And>, <Or>, <Not>, <Boolean>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <BeginsWith>, <EndsWith>, <Greater>, <Less>, <GreaterOrEqual>, <LessOrEqual>, <Contains>, <MatchPattern>, <ParseIPAddress>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

**Attributes:** None

**Usage Guidelines:** Use this element to encapsulate all the elements that define a single SIP rule.

The 'SipRule' Element has only one child element of type Boolean (Element that returns Boolean value after execution), so it could be either an operator element that returns a Boolean value after the operation is performed or it could be a data element that returns a Boolean data value.

#### Example

```
<SipRule>
<And>
<NotExists><SipRequestLine/></NotExists>
<Equal>
<SessionVariable index="1" />
<Number value="100"/>
</Equal>
</And>
</SipRule>
```

The example above checks, if 'SipRequestLine' does not exist in a Sip Message (i.e., it is a SIP response) 'And' the data value stored inside the 'SessionVariable' at index "1" is "100" (a number). Note that the 'SipRule' contains the child element 'And', which is a Boolean operator element.

### 10.2.1.4 <Action>

Parent element: <ProfilerRule>

Child elements: <Assign>, <Execute>, <Insert>, <Delete>, <Print>, <Return>, <DropSipMessage>, <RejectSipMessage>

**Attributes:** None

#### Usage Guidelines

Use this element to encapsulate all the 'Action' statements corresponding to one 'ProfilerRule' element.

**Example**

```

<Action>

<Assign>

<LocalVariable index="1"/>

<String value="Test"/>

</Assign>

<Return/>

</Action>

```

The example above stores a string "Test" inside a local variable at index "1" and then instructs the Profiler to stop execution of further 'ProfilerRule' elements using the 'Return' element.

## 10.2.2 Operator Elements

### 10.2.2.1 <And>

Parent element: <SipRule>, <And>, <Or>, <Not>, <Equal>, <NotEqual>

Child elements: <And>, <Or>, <Not>, <Boolean>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <BeginsWith>, <EndsWith>, <Greater>, <Less>, <GreaterOrEqual>, <LessOrEqual>, <Contains>, <MatchPattern>, <ParseIPAddress>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

**Returns:** Boolean True or False

**Attributes:** None

**Usage Guidelines:** Use this element to perform the 'And' operation on Boolean values.

All child elements of 'And' should be Boolean - i.e., either it should be an operator element that returns a Boolean result or it should be a data element that returns a Boolean data value.

---

**Note:**

There should be at least two child elements.

---

**Example**

```

<And>

<NotExists><SipRequestLine/></NotExists>

<Equal>

<SessionVariable index="1" />

<Number value="100"/>

</Equal>

```

</And>

The example above checks, if 'SipRequestLine' does not exist in a Sip Message (i.e. it is a SIP response) 'And' the data value stored inside the 'SessionVariable' at index "1" is "100" (a number).

### 10.2.2.2 <BeginsWith>

Parent element: <SipRule>, <And>, <Or>, <Not>, <Equal>, <NotEqual>

Child elements: <String>, <Concatenate>, <ReplaceString>, <RemoveString>, <SubString>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>

**Returns:** Boolean

**Attributes:** None

**Usage Guidelines:** Use this element to check if a string begins with a sub string.

This element has exactly two child elements. The first element is the string to be checked and second element is the sub string.

Both child elements of the 'BeginsWith' element should be string (i.e., they should be operator elements, which return a string result value, or they should be data elements that return a string data value).

#### Example

<BeginsWith>

<SipHeader Header="From" Field="Address\_User"/>

<String Value="test"/>

</BeginsWith>

The example above checks whether the user part of the Address field of a 'From' SIP header begins with string "test" or not. If so, then "True" is returned, otherwise "False" is returned.

### 10.2.2.3 <Concatenate>

Parent element: <Equal>, <NotEqual>, <Assign>, <ParseIPAddress>, <Concatenate>, <RemoveString>, <ReplaceString>

Child elements: <String>, <Concatenate>, <ReplaceString>, <RemoveString>, <SubString>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

**Returns:** String

**Attributes:** None

**Usage Guidelines:** Use this element to concatenate two or more strings.

All child elements of 'Concatenate' element should be string i.e. either they should be operator elements, which return string result value or they should be data elements that return string data value.

---

#### Note:

There should be at least two child elements.

---

#### Example

```
<Concatenate>
```

```
<SipHeader Header="From" Field="Address_User"/>
```

```
<String Value="@dialogic.com"/>
```

```
</Concatenate>
```

The example above performs a concatenate operation between the user part of an Address field of 'From' SIP header and a constant string "@dialogic.com" and returns the concatenated string as result of the operation. If the user part value is "test" then the resultant string will be [test@dialogic.com](mailto:test@dialogic.com)

### 10.2.2.4 <Contains>

Parent element: <SipRule>, <And>, <Or>, <Not>, <Equal>, <NotEqual>

Child elements: <String>, <Concatenate>, <ReplaceString>, <RemoveString>, <SubString>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

**Returns:** Boolean

**Attributes:** None

**Usage Guidelines:** Use this element to check if a string contains a sub string.

This element has exactly two child elements. The first element is the string to be checked and the second element is the sub string.

Both child elements of the 'Contains' element should be string, i.e. either they should be operator elements, which return a string result value or they should be data elements that return a string data value.

#### Example

```
<Contains>
```

```
<SipHeader Header="From" Field="Address_User"/>
```

```
<String Value="test"/>
```

```
</Contains>
```

The example above checks, whether the user part of the Address field of the 'From' SIP header contains a sub string "test" or not. If so, then "True" is returned, otherwise "False" is returned.

### 10.2.2.5 <Difference>

Parent element: <Equal>, <NotEqual>, <Assign>

Child elements: <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>, <Number>, <Sum>, <Difference>, <Product>

**Returns:** Number

**Attributes:** None

**Usage Guidelines:** Use this element to find difference between two number elements.

This element has exactly two child elements. Both child elements should be of type number - i.e., either they should be operator elements, which return number result value or they should be data elements that return number data value.

#### Example

```
<Difference>

<SipHeader Header="Max-Forwards" Field="Value"/>

<Number Value="10" />

</Difference>
```

The example above finds the difference between the Max-Forwards header value received in the SIP message and a number constant "10". If the Max-Forwards value received in the SIP message is "65" then <Difference> will return "55" as the result, since 65 - 10 = 55.

### 10.2.2.6 <EndsWith>

Parent element: <SipRule>, <And>, <Or>, <Not>, <Equal>, <NotEqual>

Child elements: <String>, <Concatenate>, <ReplaceString>, <RemoveString>, <SubString>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>

Returns: Boolean

Attributes: None

**Usage Guidelines:** Use this element to check if a string ends with a sub string.

This element has exactly two child elements. The first element is the string to be checked and the second element is the sub string.

Both child elements of the 'EndsWith' element should be string, i.e. either they should be operator elements, which return a string result value or they should be data elements that return a string data value.

#### Example

```
<EndsWith>

<SipHeader Header="From" Field="Address_User"/>

<String Value="test"/>

</EndsWith>
```

The example above checks whether or not the user part of the Address field of a 'From' SIP header ends with the string "test". If so, then "True" is returned, otherwise "False" is returned.

### 10.2.2.7 <Equal>

Parent element: <SipRule>, <Equal>, <NotEqual>, <And>, <Or>, <Not>

1st Child elements: <Number>, <Sum>, <Difference>, <Product>, <String>, <Concatenate>, <ReplaceString>, <RemoveString>, <SubString>, <Boolean>, <And>, <Or>, <Contains>, <BeginsWith>, <EndsWith>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <Not>



<Greater>, <GreaterOrEqual>, <Less>, <LessOrEqual>, <ParseIPAddress>, <MatchPattern>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

2nd Child element: Same as 1st or <List>

**Returns:** Boolean True or False

**Attributes:** None

**Usage Guidelines:** Use this element to check if two data values are equal.

This element has exactly two child elements. The data type of both child elements should be the same for this operation to be successful. For example, you can equate a Boolean value only to Boolean data.

#### Example

<Equal>

<SipRequestLine Field="Method"/>

<SipHeader Header="CSeq" Field="Method"/>

</Equal>

The example above checks whether the method name inside the SIP Request Line is equal to the method name in the CSeq Header. If it is equal then the <Equal> operation will return "True" otherwise it will return "False".

<Equal>

<SipRequestLine Field="Method"/>

<List>

<String Value="Subscribe"/>

<String Value="Notify"/>

<String Value="Refer"/>

<String Value="Update"/>

</List>

</Equal>

The example above checks if the method name inside the SIP Request Line is equal to any of the string constants defined inside <List>. If it is equal, then the <Equal> operation will return "True" otherwise it will return "False".

### 10.2.2.8 <Exists>

Parent element: <Equal>, <NotEqual>, <SipRule>, <And>, <Or>, <Not>

Child element: <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

**Returns:** Boolean True or False

**Attributes:** None

**Usage Guidelines:** Use this element to check if a data reference exists or not.

This element has only one child element. If the data exists, then this element returns "True" otherwise it returns "False".

**Example**

```
<Exists>
```

```
<SipRequestLine/>
```

```
</Exists>
```

The example above checks if the SIP message contains a Request Line (i.e., the SIP message is a SIP Request and not a SIP Response).

```
<Exists>
```

```
<SipHeader Header="Via" Index="1"/>
```

```
</Exists>
```

The example above checks if the SIP message contains at least one Via header.

### 10.2.2.9 <Greater>

Parent element: <Equal>, <NotEqual>, <Assign>, <And>, <Or>, <Not>, <SipRule>

Child elements: <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>, <Number>, <Sum>, <Difference>, <Product>

**Returns:** Boolean

**Attributes:** None

**Usage Guidelines:** Use this element to find if a number is greater than another.

This element has exactly two child elements. Both child elements should be of type number (i.e., either they should be operator elements, which return a number result value or they should be data elements that return a number data value).

**Example**

```
<Greater>
```

```
<SipHeader Header="Max-Forwards" Field="Value"/>
```

```
<Number Value="65" />
```

```
</Greater>
```

The example above checks if Max-Forwards header value received in a SIP message is greater than the number "65", which is a constant. If yes, then <Greater> will return true, otherwise it will return false.

### 10.2.2.10 <GreaterOrEqual>

Parent element: <Equal>, <NotEqual>, <Assign>, <And>, <Or>, <Not>, <SipRule>

Child elements: <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>, <Number>, <Sum>, <Difference>, <Product>

**Returns:** Boolean

**Attributes:** None

**Usage Guidelines:** Use this element to find if a number is greater than equal to another number.

This element has exactly two child elements. Both child elements should be of type number (i.e., either they should be operator elements, which return a number result value or they should be data elements that return a number data value).

#### Example

```
<GreaterOrEqual>
```

```
<SipHeader Header="Max-Forwards" Field="Value"/>
```

```
<Number Value="65" />
```

```
</GreaterOrEqual>
```

The example above checks if Max-Forwards header value received in a SIP message is greater than or equal to the number "65", which is a constant. If yes, then <GreaterOrEqual> will return true, otherwise it will return false.

### 10.2.2.11 <Less>

Parent element: <Equal>, <NotEqual>, <Assign>, <And>, <Or>, <Not>, <SipRule>

Child elements: <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>, <Number>, <Sum>, <Difference>, <Product>

**Returns:** Boolean True or False

**Attributes:** None

**Usage Guidelines:** Use this element to find if a number is less than another.

This element has exactly two child elements. Both child elements should be of type number - i.e., either they should be operator elements, which return a number result value or they should be data elements that return a number data value.

#### Example

```
<Less>
```

```
<SipHeader Header="CSeq" Field="Step"/>
```

```
<Number Value="1" />
```

```
</Less>
```

The example above checks whether CSeq header's step value is less than a number constant "1" or not. If yes, then <Less> will return true, otherwise it will return false.

### 10.2.2.12 <LessOrEqual>

Parent element: <Equal>, <NotEqual>, <Assign>, <And>, <Or>, <Not>, <SipRule>

Child elements: <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>, <Number>, <Sum>, <Difference>, <Product>

**Returns:** Boolean

**Attributes:** None

**Usage Guidelines:** Use this element to find if a number is less than or equal to another number.

This element has exactly two child elements. Both child elements should be of type number - i.e., either they should be operator elements, which return a number result value or they should be data elements that return a number data value.

#### Example

```
<LessOrEqual>
```

```
<SipHeader Header="CSeq" Field="Step"/>
```

```
<Number Value="2" />
```

```
</LessOrEqual>
```

The example above checks whether CSeq header's step value is less than or equal to number constant "2" or not. If yes, then <Less> will return true, otherwise it will return false.

### 10.2.2.13 <MatchPattern>

Parent element: <Equal>, <NotEqual>, <Assign>, <And>, <Or>, <Not>, <SipRule>

1st Child elements: <String>, <Concatenate>, <ReplaceString>, <RemoveString>, <SubString>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

2nd Child element: <RegExpr>

**Returns:** Boolean

**Attributes:** None

**Usage Guidelines:** Use this element to match a string value to a regular expression pattern.

This element has exactly two child elements:

The first child element is the string value that needs to be matched. This element should be of type string; i.e., either it should be an operator element, which returns a string result value or it should be a data element that returns a string data value.

The second child element is a constant regular expression pattern against which the string value will be matched.

#### Example

```
<MatchPattern>
```

```
<SipHeader Header="From" Field="Address_User"/>
```

```
<RegExpr Pattern="^test"/>
```

`</MatchPattern>`

The example above checks whether the user part of the Address field in the 'From' header starts with the string "test" or not. If yes, then `<MatchPattern>` will return "True", otherwise it will return "False".

### 10.2.2.14 `<Not>`

Parent element: `<Equal>`, `<NotEqual>`, `<SipRule>`, `<And>`, `<Or>`, `<Not>`

Child element: `<And>`, `<Or>`, `<Not>`, `<Boolean>`, `<Equal>`, `<NotEqual>`, `<Exists>`, `<NotExists>`, `<BeginsWith>`, `<EndsWith>`, `<Greater>`, `<Less>`, `<GreaterOrEqual>`, `<LessOrEqual>`, `<Contains>`, `<MatchPattern>`, `<ParseIPAddress>`, `<Configuration>`, `<IP>`, `<SessionVariable>`, `<TransactionVariable>`, `<LocalVariable>`, `<SipHeader>`, `<SipParameter>`, `<SipRequestLine>`, `<SipStatusLine>`, `<SipBody>`

**Returns:** Boolean

**Attributes:** None

**Usage Guidelines:** Use this element to negate a Boolean data value.

This element has only one child element. The data type of the child element should be of type Boolean - i.e., either it should be operator elements, which return a Boolean result value or it should be a data element that returns a Boolean data value.

#### Example

```
<Not><SipHeader Header="Contact" Field="Star"/></Not>
```

The example above negates the value returned by the Contact header value's field Star. If the Contact header's value is Star, then `<Not>` will return "False", otherwise it will return "True".

```
<Not><Exists>
```

```
<SipHeader Header="Via" Index="1"/>
```

```
</Exists></Not>
```

The example above negates the result returned by the `<Exists>` operation. If `<Exists>` element returned true, then `<Not>` will return false and vice versa.

---

#### Note:

`<Not><Exists>` is equivalent to the `<NotExists>` element in operation.

---

### 10.2.2.15 `<NotEqual>`

Parent element: `<SipRule>`, `<Equal>`, `<NotEqual>`, `<And>`, `<Or>`, `<Not>`

1st Child elements: `<Number>`, `<Sum>`, `Difference>`, `<Product>`, `<String>`, `<Concatenate>`, `<ReplaceString>`, `<RemoveString>`, `SubString>`, `<Boolean>`, `<And>`, `<Or>`, `<Contains>`, `<BeginsWith>`, `<EndsWith>`, `<Equal>`, `<NotEqual>`, `<Exists>`, `<NotExists>`, `<Not>`, `<Greater>`, `<GreaterOrEqual>`, `<Less>`, `<LessOrEqual>`, `<ParseIPAddress>`, `<MatchPattern>`, `<Configuration>`, `<IP>`, `<SessionVariable>`, `<TransactionVariable>`, `<LocalVariable>`, `<SipHeader>`, `<SipParameter>`, `<SipRequestLine>`, `<SipStatusLine>`, `<SipBody>`

2nd Child element: Same as 1st or `<List>`

**Returns:** Boolean

**Attributes:** None

**Usage Guidelines:** Use this element to check if two data values are not equal.

This element has exactly two child elements. The data type of both child elements should be the same for this operation to be successful - e.g., you can equate a Boolean value only to Boolean data.

**Example**

```
<NotEqual>
<SipHeader Header="Via" Index="1" Field="Transport"/>
<IP Field="InInterfaceProtocol"/>
</NotEqual>
```

The example above checks if the transport field inside the first Via header is not equal to the incoming interface's transport. If it is not equal then the <NotEqual> operation will return "True" otherwise it will return "False".

```
<NotEqual>
<SipHeader Header="Contact" Field="Address_Scheme"/>
<List>
<String Value="Sip"/>
<String Value="Sips"/>
<String Value="Tel"/>
</List>
</NotEqual>
```

The example above checks if the scheme field inside the first Contact header's Uri is not equal to all of the string constants defined in <List>. If it is not equal then the <NotEqual> operation will return "True" otherwise if scheme is equal to any one of the specified string constants it will return "False".

### 10.2.2.16 <NotExists>

Parent element: <Equal>, <NotEqual>, <SipRule>, <And>, <Or>, <Not>

Child element: <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

**Returns:** Boolean

**Attributes:** None

**Usage Guidelines:** Use this element to check if a data reference does not exist.

This element has only one child element. If the data does not exist then this element returns "True" otherwise it returns "False".

**Example**

```
<NotExists>
```

```
<SipRequestLine/>
```

```
</NotExists>
```

The example above checks if the SIP message does not contain a Request Line - i.e., the SIP message is a SIP Response and not a SIP Request.

```
<NotExists>
```

```
<LocalVariable Index="1"/>
```

```
</NotExists>
```

The example above checks if there is some data stored in a local variable indexed at '1'.

### 10.2.2.17 <Or>

Parent element: <SipRule>, <And>, <Or>, <Not>, <Equal>, <NotEqual>

Child elements: <And>, <Or>, <Not>, <Boolean>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <BeginsWith>, <EndsWith>, <Greater>, <Less>, <GreaterOrEqual>, <LessOrEqual>, <Contains>, <MatchPattern>, <ParseIPAddress>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

**Returns:** Boolean

**Attributes:** None

**Usage Guidelines:** Use this element to perform the 'Or' operation on Boolean values.

All child elements of 'Or' should be Boolean - i.e., either they should be operator elements that return a Boolean result or they should be data elements that return Boolean data value.

---

**Note:**

There should be at least two child elements.

---

**Example**

```
<Or>
```

```
<NotExists><SipHeader Header="Authorization" Field="UserName"/></NotExists>
```

```
<NotExists><SipHeader Header="Authorization" Field="Nonce"/></NotExists>
```

```
</Or>
```

In the example above, the <Or> operation will return "True" if either the UserName or Nonce field does not exist in the Authorization header in the SIP message.

### 10.2.2.18 <ParseIPAddress>

Parent element: <Equal>, <NotEqual>, <And>, <Or>, <Not>, <SipRule>

Child elements: <String>, <Concatenate>, <ReplaceString>, <RemoveString>, <SubString>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

**Returns:** Boolean

**Attributes**

Name	Required	Type	Description	Value Set	Default Value
Type	M	Enum	Type of IP address	IPv4   IPv6	

**Usage Guidelines:** Use this element to check if a string is an IP address or not.

This element has only one child element. The data type of this child element should be of type string - i.e., either it should be an operator element, which returns a string result value or it should be a data element that returns a string data value.

**Example**

```
<ParseIPAddress Type="IPv4">
```

```
<SipHeader Header="Record-Route" Field="Address_Host"/>
```

```
</ParseIPAddress>
```

In the example above, <ParseIPAddress> returns "True" if the host field of Record-Route header's Uri is an IPv4 address, otherwise it returns "False".

```
<ParseIPAddress Type="IPv6">
```

```
<SipHeader Header="Contact" Field="Address_Host"/>
```

```
</ParseIPAddress>
```

In the example above, <ParseIPAddress> returns "True" if the host field of Contact header's Uri is an IPv6 address, otherwise it returns "False".

## 10.2.2.19 <Product>

Parent element: <Equal>, <NotEqual>, <Assign>

Child elements: <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>, <Number>, <Sum>, <Difference>, <Product>

**Returns:** Number

**Attributes:** None

**Usage Guidelines:** Use this element to find the product of two or more number elements.

This element should have at least two child elements (there may be more). All child elements should be of type number - i.e., they should either be operator elements, which return a number result value or they should be data elements that return a number data value.

**Example**

```
<Product>
```

```
<SipHeader Header="Max-Forwards" Field="Value"/>
```

```
<Number Value="10" />
```



</Product>

The example above finds the product of Max-Forwards header value received in the SIP message and a number constant "10". If the Max-Forwards value received in the SIP message is "65" then <Sum> will return "650" as the result, since  $65 * 10 = 650$ .

### 10.2.2.20 <RegExpr>

Parent element: <MatchPattern>

Child elements: None

#### Attributes

Name	Required	Type	Description	Value Set	Default Value
Pattern	M	String	Regular expression pattern		

**Usage Guidelines:** Use this element to represent a constant regular expression pattern. This element has no child elements and is always used under the context of the <MatchPattern> element.

This element has one attribute 'Pattern' which contains the constant regular expression pattern.

The following Meta characters are supported for regular expression patterns:

#### Regular Expression Meta Characters

Char	Meaning
^	Beginning of string
\$	End of string
.	Any character except newline
*	Match 0 or more times
+	Match 1 or more times
?	Match 0 or 1 times
	Alternative
()	Grouping
[]	Set of characters
{}	Repetition modifier
\	Quote or special

#### Regular Expression Repetition Examples

Char	Meaning
a*	zero or more a's
a+	one or more a's
a?	zero or one a's (i.e., optional a)
a{m}	exactly m a's
a{m,}	at least m a's

Char	Meaning
a{m,n}	at least m but at most n a's

**Example**

```
<MatchPattern>
```

```
<SipHeader Header="From" Field="Address_Host"/>
```

```
<RegExpr Pattern="com$"/>
```

```
</MatchPattern>
```

The example above checks whether or not the 'host' part of the Address field in the 'From' header ends with the string "com". If yes, then <MatchPattern> will return true, otherwise it will return false.

**10.2.2.21 <RemoveString >**

Parent element: <Equal>, <NotEqual>, <Assign>, <ParseIPAddress>, <Concatenate>, <RemoveString>, <ReplaceString>

Child elements: <String>, <Concatenate>, <ReplaceString>, <RemoveString>, <SubString>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

**Returns:** String

**Attributes**

Name	Required	Type	Description	Value Set	Default Value
Offset	O	Number	Location of first character in string- where to start removing		1
Length	O	Number	Number of characters to be removed		Length of String

**Usage Guidelines:** Use this element to remove part of a string.

This element has the two attributes 'Offset' and 'Length'. These attributes mark the boundary in the string. This boundary is used to find the sub string which will be removed. Both of these attributes are optional. If *Offset* is not present, then the search for the sub-string starts from the 1st character of the parent string. If *Length* is not present, then the search for the sub string ends at the last character of the parent string.

This element has one mandatory child element and a second optional child element.

- The first child element is the string on which the remove operation will be performed (parent string).
- The second optional element is the sub string which needs to be removed. If this element is not present then the entire sub string starting from 'Offset' to 'Length' will be removed. If the second element is present then the string from 'Offset' to 'Length' is searched for the presence of this sub string and if found is then removed.

All child elements of the 'RemoveString' element should be string, i.e. either they should be operator elements, which return string result value or they should be data elements that return string data value.

**Example**

```
<RemoveString Offset ="5">
```

```
<String Value="testimony"/>
```

```
</RemoveString>
```

In the first example above, <RemoveString> will return the string "test", removing rest of string starting from 5th character through the end of the string

```
<RemoveString Offset ="6">
```

```
<String Value="Dialogician"/>
```

```
<String Value="ian"/> <!--Search for 'ian' starting from 6th character in Dialogician-->
```

```
</RemoveString>
```

In the next example above, <RemoveString> will return the string "Dialogic.".

### 10.2.2.22 <ReplaceString>

Parent element: <Equal>, <NotEqual>, <Assign>, <ParseIPAddress>, <Concatenate>, <RemoveString>, <ReplaceString>

Child elements: <String>, <Concatenate>, <ReplaceString>, <RemoveString>, <SubString>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

**Returns:** String

#### Attributes

Name	Required	Type	Description	Value Set	Default Value
Offset	O	Number	Location of first character in string- where to start removing		1
Length	O	Number	Number of characters to be removed		Length of String

**Usage Guidelines:** Use this element to replace part of a string.

This element has the two attributes "Offset" and "Length". These attributes mark the boundary in the string. This boundary is then used to find the sub string which will be replaced with a new string. Both of these attributes are optional. If *Offset* is not present, then the search for the sub string starts from the 1st character of the parent string. If *Length* is not present, then the search for the sub string ends at the last character of the parent string.

This element has two mandatory child elements and a third optional child element.

- The first child element is the string on which the replace operation will be performed (parent string).
- The second child element is the replacement sub string.
- The third optional element is the sub string which needs to be replaced. If this element is not present then the entire sub string starting from 'Offset' to 'Length' will be replaced. If the third element is present then the string from 'Offset' to 'Length' is searched for the presence of this sub string, and if found is then replaced with the replacement sub string.

All child elements of the 'ReplaceString' element should be string, i.e. either they should be operator elements, which return a string result value or they should be data elements that return a string data value.

#### Example

```
<ReplaceString Offset ="1" Length="4">
<String Value="testimony"/>
<String Value="matr"/> <!--Replace sub string starting from 1st character to length of 4 characters i.e 'test' in 'testimony' with 'matr'
-->
</ReplaceString>
```

In the example above, <ReplaceString> will return the string "matrimony".

```
<ReplaceString Offset ="6">
<String Value="Dialogic.com"/>
<String Value="net"/> <!--Replace with net-->
<String Value="com"/> <!--Search for com starting from 6th character in Dialogic.com-->
</ReplaceString>
```

In the example above, <ReplaceString> will return the string "Dialogic.net".

### 10.2.2.23 <SubString >

Parent element: <Equal>, <NotEqual>, <Assign>, <ParseIPAddress>, <Concatenate>, <RemoveString>, <ReplaceString>

**Child element:** <String>, <Concatenate>, <ReplaceString>, <RemoveString>, <SubString>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

**Returns:** String

#### Attributes

Name	Required	Type	Description	Value Set	Default Value
Offset	O	Number	Location of first character in string- where to start removing		1
Length	O	Number	Number of characters to be removed		Length of String

**Usage Guidelines:** Use this element to select part of a string.

This element has the two attributes 'Offset' and 'Length'. These attributes mark the boundary in the string. This boundary is used to find the sub string which will be selected. Both of these attributes are optional. If 'Offset' is not present, then the search for the sub string starts from the 1st character of the parent string. If 'Length' is not present, then the search for the sub string ends at the last character of the parent string.

This element has one mandatory child element, the string from which the substring will be selected.

The child element of the 'SubString' element should be string, i.e. either it should be operator elements, which return string result value or it should be data elements that return string data value.

#### Example

```
<SubString Offset="5">
<String Value="testimony"/>
</SubString>
```

In the example above, <SubString> will return the string "imony", selecting the string starting from 5th character through the end of the string

### 10.2.2.24 <Sum>

Parent element: <Equal>, <NotEqual>, <Assign>

Child elements: <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>, <Number>, <Sum>, <Difference>, <Product>

**Returns:** Number

**Attributes:** None

**Usage Guidelines:** Use this element to find the sum of two or more number elements. This element should have at least two child elements (there may be more). All child elements should be of type number - i.e., they should either be operator elements, which return a number result value or they should be data elements that return a number data value.

#### Example

```
<Sum>
<SipHeader Header="Max-Forwards" Field="Value"/>
<Number Value="10" />
</Sum>
```

The example above finds the sum of Max-Forwards header value received in the SIP message and a number constant "10". If the Max-Forwards value received in the SIP message is "65" then <Sum> will return "75" as the result, since 65 + 10 = 75.

## 10.2.3 Constant Data Elements

These elements are used to represent constant data elements - e.g., a constant string or a constant number.

### 10.2.3.1 <String>

Parent element: <BeginsWith>, <EndsWith>, <Equal>, <NotEqual>, <Assign>, <ParseIPAddress>, <Concatenate>, <List>, <Insert>

Child elements: None

#### Attributes

Name	Required	Type	Description	Value Set	Default Value
Value	M	String	Constant String		

**Usage Guidelines:** This element is used to represent a string constant.

---

**Note:**

There is a size limit of 255 characters for a string in the Profiler. A string larger than this will be truncated.

---

**Example:** `<String value ="user@dialogic.com"/>`

### 10.2.3.2 <Number>

Parent element: <Greater>, <Less>, <Sum>, <Difference>, <Equal>, <NotEqual>, <Assign>, <List>

Child elements: None

**Attributes**

Name	Required	Type	Description	Value Set	Default Value
Value	M	Number	Constant Number		

**Usage Guidelines:** This element is used to represent a number constant.

**Example:** `<Number value ="5060"/>`

### 10.2.3.3 <Boolean>

Parent element: <And>, <Or>, <Not>, <SipRule>, <Equal>, <NotEqual>

Child elements: None

**Attributes**

Name	Required	Type	Description	Value Set	Default Value
Value	M	Boolean	Constant Boolean	True   False	

**Usage Guidelines:** This element is used to represent a Boolean constant.

**Example:** `<Boolean value =True"/>`

## 10.2.4 SIP Data Elements

These elements are used to access SIP message data (Request-Line, Status-Line, headers and parameter values).

### 10.2.4.1 <SipHeader>

Parent element: <And>, <Or>, <Not>, <SipRule>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <Greater>, <Less>, <Sum>, <Difference>, <ParseIPAddress>, <Concatenate>, <List>, <Insert>, <Delete>, <Assign>

Child elements: None

**Attributes**

---

Name	Required	Type	Description	Value Set	Default
Header	M	String	Profiler known SIP header name ( <a href="#">appendix I</a> )	True   False	
OtherHeader	O	String	Unknown SIP header name		
Field	O	String	Predefined header field name		
Index	O	Number   String	Index of header to be retrieved in case of multiple headers with same name	1 -20   All	1
Location	O	Enum	Location of header in the header list, from where to start the search	Top   Bottom	Top

**Usage Guidelines:** This element is used to access SIP header values.

Attribute 'Header' contains known SIP header name ([appendix I](#)). This name uniquely identifies the header, whose value is being accessed. If there are multiple headers with the same name, the 'Index' attribute can be used to access the respective header. The 'Location' attribute specifies the location from which to start the search.

For example, if there are four 'Route' headers, and user wants to access the last 'Route' header, the user may use one of the following statements

```
<SipHeader Header="Route" Index="4">
```

```
<!--Location="Top" by default, so 4th Route header from top -->
```

Or

```
<SipHeader Header="Route" Location="Bottom">
```

```
<!--Index="1" by default, so first header from Bottom (faster way) -->
```

---

**Note:**

Index="All", is only used in the context of deleting headers.

---

If the header to be accessed is not a Profiler known header ([appendix I](#)), then the attribute 'OtherHeader' is used in combination with the 'Header' attribute. The 'OtherHeader' attribute value should be the name of the unknown header and value of the 'Header' attribute should be 'Other'.

For example, if the user wants to access a private header, let's say 'Dialogic-Pvt-Header', the user must use the following statement

```
<SipHeader Header="Other" OtherHeader="Dialogic-Pvt-Header"/>
```

The attribute 'Field' is used to access a specific field inside the header value. The Profiler does not support accessing fields of unknown headers. There is a predefined mapping of field names with their respective headers.

For example, to access the 'method' field inside the CSeq header

```
<SipHeader Header="CSeq" Field="Method"/>
```

To access the host inside the From header Uri

```
<SipHeader Header="From" Field="Address_Host"/>
```

**Example**

```
<!--Get Step field of CSeq Header -->
```

```

<SipHeader Header="CSeq" Field="Step"/>

<!--Get SIP URI of From Header -->

<SipHeader Header="To" Field="Address"/>

<!--Get User-Agent Header (Unknown Header) -->

<SipHeader Header="Other" OtherHeader="User-Agent"/>

<!--Get Second Record-Route Header -->

<SipHeader Header="Record-Route" Location="Top" Index="2" />

<!--Insert Max-Forwards header and initialize its value to 70 -->

<Insert>

<SipHeader Header="Max-Forwards"/>

<String value="70">

</Insert>

<!--Delete All Record-Route Headers -->

<Delete>

<SipHeader Header="Record-Route" Index="All"/>

</Delete>

```

### 10.2.4.2 <SipParameter>

Parent element: <And>, <Or>, <Not>, <SipRule>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <Greater>, <Less>, <Sum>, <Difference>, <ParseIPAddress>, <Concatenate>, <List>, <Insert>, <Delete>, <Assign>

Child elements: None

#### Attributes

Name	Required	Type	Description	Value Set	Default
Header	M	String	Profiler known SIP header name ( <b>appendix I</b> )		
OtherHeader	O	String	Unknown SIP header name		
Index	O	Number   String	Index of header to be retrieved in case of multiple headers with same name	1 -20   All	1
Location	O	Enum	Location of header in the header list, from where to start the search	Top   Bottom	Top
Parameter	M	String	Profiler known name of parameter to be accessed		
OtherParameter	O	String	Unknown parameter name		

#### Usage Guidelines

This element is used to access SIP header parameter values.



Attributes 'Header', 'OtherHeader', 'Index', and 'Location' are used to retrieve the respective header whose parameter needs to be accessed. The definition of these attributes is the same as in the <SipHeader> element.

The attribute 'Parameter' contains a known SIP header parameter name. This name uniquely identifies the parameter.

---

**Note:**

If a SIP header definition contains multiple parameters with the same name, the Profiler will always return the value of first such parameter. The Profiler does not have the ability to access a particular indexed parameter in such scenarios.

---

If the parameter to be accessed, is not a Profiler known parameter, then the attribute 'OtherParameter' is used in combination with the 'Parameter' attribute. The 'OtherParameter' attribute value should be the name of the unknown parameter and the value of the 'Parameter' attribute should be 'Other'.

**Example**

```
<!--Get MAddr parameter of first Via Header -->
```

```
<SipParameter Header="Via" Index="1" Parameter="MAddr"/>
```

```
<!--Get Transport port parameter of Contact header-->
```

```
<SipParameter Header="Contact" Parameter="Transport"/>
```

```
<!--Get Transport parameter from SIP Request line-->
```

```
<SipParameter header="RequestLine" parameter="Transport"/>
```

```
<!--Get unknown pvt parameter Dialogic-Param from first Via header -->
```

```
<SipParameter Header="Via" Index="1" Parameter="Other" OtherParameter="Dialogic-Param"/>
```

### 10.2.4.3 <SipRequestLine>

Parent element: <And>, <Or>, <Not>, <SipRule>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <Greater>, <Less>, <Sum>, <Difference>, <ParseIPAddress>, <Concatenate>, <List>, <Insert>, <Delete>, <Assign>

Child elements: None

**Attributes**

Name	Required	Type	Description	Value Set	Default
Field	O	String	Predefined SIP Request Line field name		

**Usage Guidelines:** This element is used to access SIP Request Line value/fields.

---

**Note:**

SIP Request Line only exists in SIP requests. SIP responses contains SIP Status Line.

---

The attribute 'Field' is used to access a specific field inside the SIP Request Line. There is a predefined set of fields for the SIP Request Line.

**Example**

```
<!--Check if SIP message is a SIP Request, if it is, 'Exists' will return True-->
```

```
<Exists><SipRequestLine/></Exists>
```

```
<!--Check if a SIP request is not a REGISTER request-->
```

```
<NotEqual>
```

```
<SipRequestLine Field="Method"/>
```

```
<String Value="Register"/>
```

```
</NotEqual>
```

### 10.2.4.4 <SipStatusLine>

Parent element: <And>, <Or>, <Not>, <SipRule>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <Greater>, <Less>, <Sum>, <Difference>, <ParseIPAddress>, <Concatenate>, <List>, <Insert>, <Delete>, <Assign>

Child elements: None

#### Attributes:

Name	Required	Type	Description	Value Set	Default
Field	O	String	Predefined SIP Status Line field name		

**Usage Guidelines:** This element is used to access SIP Status Line value/fields.

---

#### Note:

SIP Status Line only exists in SIP responses, SIP requests contains SIP Request Line.

---

The attribute 'Field' is used to access a specific field inside the SIP Status Line. There is a predefined set of fields for SIP Status Line.

#### Example

```
<!--Check if SIP message is a SIP Response, if it is, 'Exists' will return True-->
```

```
<Exists><SipStatusLine/></Exists>
```

```
<!--Get reason phrase from SIP response message-->
```

```
<SipStatusLine Field="Reason"/>
```

### 10.2.4.5 <SipBody>

Parent element: <And>, <Or>, <Not>, <SipRule>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <List>, <Delete>

Child elements: None

**Attributes:** None

**Usage Guidelines:** This element is used to access the SIP Body in a SIP message. Using this element a user may check for the existence of a body inside a SIP message as well as delete a SIP message body using <Delete>.

---

#### Note:

The Profiler does not support accessing individual elements inside a SIP Body object.

---

**Example**

```

<!--Check if Body exists in a SIP message or not-->

<Exists><SipBody/></Exists>

<!--Delete SIP Body from SIP message -->

<Delete>

<SipBody/>

</Delete>

```

## 10.2.5 Other Data Elements

### 10.2.5.1 <Configuration>

Parent element: <And>, <Or>, <Not>, <SipRule>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <Greater>, <Less>, <Sum>, <Difference>, <ParseIPAddress>, <Concatenate>, <List>

Child elements: None

**Attributes**

Name	Required	Type	Description	Value Set	Default
Field	M	String	Predefined configuration field name		

**Usage Guidelines**

This element is used to access configuration fields/values that are configured to the SCS.

The attribute 'Field' is used to access a specific field from configuration tables. There is a predefined set of fields for the <Configuration> element.

**Example**

```

<Less>

<SipHeader Header="Expires"/>

<Configuration Field='ToBeDefined'/>

</Less>

```

In the example above, <Less> returns "True" if the Expires header value is less than the configured value for the ToBeDefined field, otherwise <Less> returns "False".

### 10.2.5.2 <IP>

Parent element: <And>, <Or>, <Not>, <SipRule>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <Greater>, <Less>, <Sum>, <Difference>, <ParseIPAddress>, <Concatenate>, <List>, <Assign>

Child elements: None

#### Attributes

Name	Required	Type	Description	Value Set	Default
Field	M	String	Predefined IP field name		

**Usage Guidelines:** This element is used to access IP layer fields/values of a SIP message.

The attribute 'Field' is used to access a specific field in the IP layer. There is a predefined set of fields for the <IP> element.

#### Example

```
<NotEqual>
```

```
<SipParameter Header="Contact" Parameter="Transport"/>
```

```
<IP Field="InInterfaceProtocol"/>
```

```
</NotEqual>
```

In the example above, <NotEqual> returns "True" if the Contact header's transport parameter value is not equal to the incoming interface transport protocol, otherwise <NotEqual> returns "False".

### 10.2.5.3 <LocalVariable>

Parent element: <And>, <Or>, <Not>, <SipRule>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <Greater>, <Less>, <Sum>, <Difference>, <ParseIPAddress>, <Concatenate>, <List>, <Assign>

Child elements: None

#### Attributes

Name	Required	Type	Description	Value Set	Default
Index	O	Number	Index of variable	1 - 10	
Name	O	String	Unique Name of variable		

**Usage Guidelines:** This element is used to store and access data to the Profiler local variables.

The attributes 'Index' and 'Name' are used to uniquely identify the respective local variable.

The 'Index' attribute is mandatory when <LocalVariable> is used for storing data. The 'Index' attribute is optional when <LocalVariable> is used for accessing previously stored variable data.

The 'Name' attribute is always optional, regardless of whether data is being stored in or retrieved from a local variable. A local variable can be named at the time when data is stored in it. Once named, a local variable can be accessed either by its 'Index' or by its 'Name'.

While retrieving data from a variable either the 'Name' or 'Index' must be present to identify the variable.

#### Example

```
<Assign>
```

```
<LocalVariable Name="TempString" Index="1"/>
```

```
<String Value="Hello world" />
```

```
</Assign>
```

In the example above, a constant string "Hello world" is stored in a local variable at index "1". In addition, the local variable is given a name "TempString". Data stored in this local variable can then be retrieved by using either of following statements.

*<!--Retrieve by index-->*

```
<Equal>
```

```
<LocalVariable Index="1"/>
```

```
<String Value="Hello world" />
```

```
</Equal>
```

or

*<!--Retrieve by name-->*

```
<Equal>
```

```
<LocalVariable Name="TempString"/>
```

```
<String Value="Hello world" />
```

```
</Equal>
```

### 10.2.5.4 <TransactionVariable>

Parent element: <And>, <Or>, <Not>, <SipRule>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <Greater>, <Less>, <Sum>, <Difference>, <ParseIPAddress>, <Concatenate>, <List>, <Assign>

Child elements: None

#### Attributes

Name	Required	Type	Description	Value Set	Default
Index	O	Number	Index of variable	1 - 5	
Name	O	String	Unique Name of variable		

**Usage Guidelines:** This element is used to access and store data to Profiler transaction variables.

The attributes 'Index' and 'Name' are used to uniquely identify the respective transaction variable.

The 'Index' attribute is mandatory when <TransactionVariable> is used for storing data. The 'Index' attribute is optional when <TransactionVariable> is used for accessing previously stored variable data.

The 'Name' attribute is always optional, regardless of whether data is being stored into or retrieved from a transaction variable. A transaction variable can be named at the time when data is stored in it. Once named, a transaction variable can be accessed either by its 'Index' or by its 'Name'.

While retrieving data from a variable either the 'Name' or 'Index' must be present to identify the variable.

#### Example

```

<Assign>

<TransactionVariable Name="TempNumber" Index="4"/>

<Number Value="20" />

</Assign>

```

In the example above, a constant number "20" is stored in a transaction variable at index "4". In addition, the transaction variable is given a name "TempNumber". Data stored in this transaction variable can then be retrieved by using either of following statements.

```

<!--Retrieve by index-->

<Equal>

<TransactionVariable Index="4"/>

<Number Value="20" />

</Equal>

```

or

```

<!--Retrieve by name-->

<Equal>

<TransactionVariable Name="TempNumber"/>

<Number Value="20" />

</Equal>

```

### 10.2.5.5 <SessionVariable>

Parent element: <And>, <Or>, <Not>, <SipRule>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <Greater>, <Less>, <Sum>, <Difference>, <ParseIPAddress>, <Concatenate>, <List>, <Assign>

Child elements: None

#### Attributes

Name	Required	Type	Description	Value Set	Default
Index	O	Number	Index of variable	1 - 5	
Name	O	String	Unique Name of variable		

**Usage Guidelines:** This element is used to access and store data to Profiler session variables.

The attributes 'Index' and 'Name' are used to uniquely identify the respective session variable.

The 'Index' attribute is mandatory when <SessionVariable> is used for storing data. The 'Index' attribute is optional when <SessionVariable> is used for accessing previously stored variable data.

The 'Name' attribute is always optional, regardless of whether data is being stored into or retrieved from a session variable. A session variable can be named at the time when data is stored in it. Once named, a session variable can be accessed either by its

'Index' or by its 'Name'.

While retrieving data from a variable either the 'Name' or 'Index' must be present to identify the variable.

#### Example

```
<Assign>
```

```
<SessionVariable Name="FirstVia" Index="2"/>
```

```
<SipHeader Header="Via" Index="1" />
```

```
</Assign>
```

In example above, the first Via header is stored in a session variable at index "2". In addition, the session variable is given a name "FirstVia". Data stored in this session variable can then be retrieved by using either of following statements.

```
<!--Retrieve by index-->
```

```
<Insert>
```

```
<SipHeader Header="Via" Location="Bottom" />
```

```
<SessionVariable Index="2" />
```

```
</Insert>
```

or

```
<!--Retrieve by name-->
```

```
<Insert>
```

```
<SipHeader Header="Via" Location="Bottom" />
```

```
<SessionVariable Name="FirstVia" />
```

```
</Insert>
```

## 10.2.6 Action Elements

### 10.2.6.1 <Assign>

**Parent element:** <Action>

**1st Child element:** <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>

**2nd Child elements:** <Number>, <Sum>, <Difference>, <Product>, <String>, <Concatenate>, <ReplaceString>, <RemoveString>, <SubString>, <Boolean>, <And>, <Or>, <Contains>, <BeginsWith>, <EndsWith>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <Not>, <Greater>, <GreaterOrEqual>, <Less>, <LessOrEqual>, <ParseIPAddress>, <MatchPattern>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

**Returns:** None

**Attributes:** None

**Usage Guidelines:** Use this element to perform an 'Assign' operation to some data - e.g., this element can be used to temporarily store some data value inside a variable or it can be used to change the data value referenced by a data element.

This element has exactly two child elements. The data type of both child elements must be the same for this operation to be successful. For example, you can assign a Boolean value only to a Boolean data.

The first child element can only be a data element which is storable - i.e., you are able to store data into it. The second child element can be either an operator element or a data element which returns same data type as first child element.

---

**Note:**

Variables can store any type of data.

---

**Example**

```
<Assign>
```

```
<TransactionVariable index="1" name="Status"/>
```

```
<String value="verified_1"/>
```

```
</Assign>
```

The example above stores a string "verified\_1" inside a transaction variable at index 1, named "Status". This transaction variable can then later be accessed using either by index or by its name.

## 10.2.6.2 <Delete>

**Parent element:** <Action>

**Child elements:** <SipHeader>, <SipParameter>, <SipBody>

**Attributes:** None

**Usage Guidelines:** This element is used to delete a SIP header, a SIP parameter or the SIP body from the SIP message.

This element has exactly one child element (<SipHeader>, <SipParameter> or <SipBody>)

**Example**

```
<!--To delete a body from SIP message -->
```

```
<Delete><SipBody/></Delete>
```

```
<!--To delete first Via header from SIP message-->
```

```
<Delete><SipHeader Header="Via" Index="1"/></Delete>
```

## 10.2.6.3 <DropSipMessage>

**Parent element:** <Action>

**Child elements:** None

**Attributes:** None

**Usage Guidelines:** This element is used to drop a SIP message.



**Example**

```

<!--To Drop a SIP Message from Profiler, if no branch in Via header -->

<ProfilerRule Id="Drop" Comment="Request with no via header">

<SipRule>

<NotExists><SipParameter Header="Via" Parameter="Branch"/></NotExists>

</SipRule>

<Action>

<DropSipMessage/>

</Action>

</ProfilerRule>

```

**10.2.6.4 <Execute>****Parent element:** <Action>**Child elements:** None**Attributes**

Name	Required	Type	Description	Value Set	Default
Rule	M	String	Name of the new Profiler to be executed		

**Usage Guidelines:** This element is used to execute another Profiler document from within the currently executing Profiler.

The attribute 'Rule' contains the name of the new Profiler document to be executed.

**Note:**

The new Profiler to be executed should already be loaded on the BorderNet SBC.

**Example**

```

<!--To execute a new Profiler document from currently executing Profiler -->

<Execute Rule="ChildProfiler.xml"/>

```

**10.2.6.5 <Insert>****Parent element:** <Action>**1st Child element:** <SipHeader>

**2nd Child element:** <And>, <Or>, <Equal>, <NotEqual>, <Exists>, <NotExists>, <Not>, <Greater>, <Less>, <ParseIPAddress>, <Boolean>, <Number>, <String>, <Concatenate>, <Configuration>, <IP>, <SessionVariable>, <TransactionVariable>, <LocalVariable>, <SipHeader>, <SipParameter>, <SipRequestLine>, <SipStatusLine>, <SipBody>

**Attributes:** None

**Usage Guidelines:** This element is used to insert a new SIP header inside the SIP message.

This element has exactly two child elements; the first child element is <SipHeader>. The second child element is any data element which can be used to initialize the newly inserted header.

---

**Note:**

To insert multiple headers, use multiple <insert> tags.

---

**Example**

```
<!--Insert a new Expired header in SIP message with value = 3600 -->
```

```
<Insert>
```

```
<SipHeader Header="Expires"/>
```

```
<Number Value="3600"/>
```

```
</Insert>
```

## 10.2.6.6 <RejectSipMessage>

**Parent element:** <Action>

**Child elements:** None

**Attributes:**

Name	Required	Type	Description	Value Set	Default
StatusCode	M	Number	SIP Negative Status code		
Warning	M	String	Reason of rejection		

**Usage Guidelines:** This element is used to reject a SIP message with a negative response.

The attribute 'StatusCode' contains the negative response code with which the request will be rejected.

The attribute 'Warning' contains a reason for the rejection. This reason is added as a Warning header inside the negative response.

**Example**

```
<!--To Reject a SIP Message from Profiler, if no branch in Via header -->
```

```
<ProfilerRule Id="Reject" Comment="Request with no via header">
```

```
<SipRule>
```

```
<NotExists><SipParameter Header="Via" Parameter="Branch"/></NotExists>
```

```
</SipRule>
```

```
<Action>
```

```
<RejectSipMessage StatusCode='400' Warning='Invalid Via header'/>
```

```
</Action>
```

```
</ProfilerRule>
```

### 10.2.6.7 <Return>

**Parent element:** <Action>

**Child elements:** <None>

**Attributes:** None

**Usage Guidelines:** This element is used to stop the Profiler execution after a certain condition is met.

#### Example

```
<!--Stop profiler execution if the SIP message is not a SIP Request -->
```

```
<ProfilerRule Id='opt1' Comment='Optimization rule'>
```

```
<SipRule>
```

```
<NotExists><SipRequestLine/></NotExists>
```

```
</SipRule>
```

```
<Action>
```

```
<Return/>
```

```
</Action>
```

```
</ProfilerRule>
```

## 10.3 Examples

### 10.3.1 Retrieving and Modifying SIP Header/Parameter Values

#### 10.3.1.1 Retrieve a Header Field

```
<ProfilerRule Id="uniqueid_1" Comment="URI scheme in Contact header is not sip or tel">
```

```
<SipRule>
```

```
<And>
```

```
<Exists><SipHeader Header="Contact"/></Exists>
```

```
<NotEqual>
```

```
<SipHeader Header="Contact" Field="Address_Scheme"/>
```

```
<List>
```

```
<String Value="Sip"/>
```

```
<String Value="Tel"/>

</List>

</NotEqual>

</And>

</SipRule>

<Action>

<RejectSipMessage StatusCode='416' Warning='Unsupported URI Scheme'/>

</Action>

</ProfilerRule>
```

In the example above, the 'SipRule' checks for two conditions

1. Contact Header exists

And

1. Contact Header's address scheme is neither 'Sip' nor 'Tel'

If both conditions are true, this Profiler rule will reject the SIP message with SIP status code 416 and a warning stating "Unsupported URI Scheme".

### 10.3.1.2 Modify a Header Field

```
<ProfilerRule Id="uniqueid_3" Comment="Change User part of To Header Address">

<SipRule>

<Equal>

<SipHeader Header="To" Field="Address_User"/>

<String Value="testa"/>

</Equal>

</SipRule>

<Action>

<Assign>

<SipHeader Header="To" Field="Address_User"/>

<String Value="testb"/>

</Assign>

</Action>

</ProfilerRule>
```

In the example above, the 'SipRule' checks if the User field of the To Header's address is equal to "testa", then re-assigns it to "testb".

### 10.3.2 Retrieve a Header Parameter

```
<ProfilerRule Id="uniqueid_3" Comment="Contact header transport is not valid ">

<SipRule>

<NotEqual>

<SipParameter Header="Contact" Parameter="Transport"/>

<IP Field="InInterfaceProtocol"/>

</NotEqual>

</SipRule>

<Action>

<RejectSipMessage StatusCode='406' Warning='Contact header transport invalid'/>

</Action>

</ProfilerRule>
```

In the example above, the 'SipRule' checks if the Contact Header's transport parameter value is not equal to the transport configured on the incoming SCS interface. If it is not equal then it rejects the SIP message with status code 416.

### 10.3.3 Adding New SIP Headers

#### Insert a New Known SIP Header

```
<ProfilerRule id='uniqueid_10' comment=' Add Max-Forwards if missing from Sip Requests '>

<SipRule>

<NotExists><SipHeader header="Max-Forwards"/></NotExists><!--if SIP header Max-Forwards does not exist -->

</SipRule>

<Action>

<Insert><!--Insert SIP header Max-Forwards with a value 70 -->

<SipHeader header="Max-Forwards"/>

<String value="70"/>

</Insert>

</Action>

</ProfilerRule>
```

```
</Profiler>
```

In the second Profiler rule, the 'SipRule' checks if the SIP request contains Max-Forwards header or not. If not, then 'NotExists' inside the 'SipRule' will return "true" and 'Action' is executed. The action is to insert a Max-Forwards header in a SIP request and initialize its value to "70".

## 10.3.4 Insert a New Unknown SIP Header

```
<ProfilerRule Id="uniqueid_11" Comment="Insert a proprietary header">
```

```
<SipRule>
```

```
<Equal>
```

```
<SipRequestLine Field="Address_User"/>
```

```
<String Value="7777"/>
```

```
</Equal>
```

```
</SipRule>
```

```
<Action>
```

```
<Insert>
```

```
<SipHeader Header="Other" OtherHeader="MyHeader"/>
```

```
<String Value="7777"/>
```

```
</Insert>
```

```
</Action>
```

```
</ProfilerRule>
```

In the second Profiler rule, the 'SipRule' checks if user part of SIP message request line is equal to "7777" then insert a new unknown header 'MyHeader' with value "7777" in to the message.

## 10.3.5 Deleting SIP Headers

### 10.3.5.1 Delete a SIP Header

```
<ProfilerRule Id="uniqueid_20" Comment="Delete first Route header">
```

```
<SipRule>
```

```
<Equal>
```

```
<SipRequestLine Field="Address_User"/>
```

```
<String Value="7777"/>
```

```
</Equal>
```

```
</SipRule>

<Action>

<Delete><SipHeader Header="Route" Index="1"/></Delete>

</Action>

</ProfilerRule>
```

In example above, the 'SipRule' checks if the user part of the SIP message request line is equal to "7777", then deletes the first 'Route' header present inside the SIP message.

### 10.3.5.2 Delete All SIP Headers of One Type

```
<ProfilerRule Id="uniqueid_21" Comment="Delete first Route header">

<SipRule>

<Exists><SipHeader Header="Route"/></Exists>

</SipRule>

<Action>

<Delete><SipHeader Header="Route" Index="All"/></Delete>

</Action>

</ProfilerRule>
```

In the example above, the 'SipRule' checks if there is any Route header present in the SIP message, then deletes all the Route headers from the message.

### 10.3.6 Adding a New SIP Header Parameter

#### Insert a New Parameter

```
<ProfilerRule Id="uniqueid_30" Comment="Insert a parameter dummy in top most Record-Route header">

<SipRule>

<Exists><SipRequestLine/></Exists>

</SipRule>

<Action>

<Assign>

<SipParameter Header="Record-Route" Location="Top" Parameter="Other" OtherParameter="Dummy"/>

<String Value="12345"/>

</Assign>

</Action>
```

```
</ProfilerRule>
```

In the example above, the 'SipRule' checks if the SIP message received is a SIP Request, then it adds a new unknown parameter 'Dummy' to the top most 'Record-Route' header.

## 10.3.7 Deleting SIP Header Parameter

### Delete a SIP Header Parameter Example

```
<ProfilerRule Id="uniqueid_31" Comment="Delete maddr parameter from topmost Via header">
```

```
<SipRule>
```

```
<Exists><SipParameter Header="Via" Index="1" Parameter="Maddr"/></Exists>
```

```
</SipRule>
```

```
<Action>
```

```
<Delete>
```

```
<SipParameter Header="Via" Index="1" Parameter="Maddr"/>
```

```
</Delete>
```

```
</Action>
```

```
</ProfilerRule>
```

In the example above, the 'SipRule' checks if the top-most 'Via' of the SIP message contains the 'maddr' parameter then deletes this parameter.

## 10.3.8 Retrieving and Storing Data in From/To Variables

### 10.3.8.1 Store a String in a Transaction Variable

```
<Profiler>
```

```
<ProfilerRule id="uniqueid_40" comment="Store a String in Transaction Variable if SipRequest is INVITE with CSeq value as 1">
```

```
<SipRule>
```

```
<And>
```

```
<Equal>
```

```
<SipRequestLine field="Method"/>
```

```
<String value="INVITE"/>
```

```
</Equal>
```

```
<Equal>
```

```
<SipHeader header="CSeq" field="Step"/>
```



```
<Number value="1"/>
</Equal>
</And>
</SipRule>
<Action>
<Assign>
<TransactionVariable index="1" name="Status"/>
<String value="verified_1"/>
</Assign>
</Action>
</ProfilerRule>
</Profiler>
```

In this Profiler rule, 'SipRule' evaluates the following two conditions and checks whether both conditions return true:

1. SIP request is an INVITE request

And

1. SIP request contains a CSeq header, whose Step value is "1"

If both conditions return "true" then the 'Action' element is executed. The 'Action' element stores a string "verified\_1" inside a transaction variable whose index is "1" and whose name is "Status".

### 10.3.8.2 Retrieve a String from a Session Variable

```
<Profiler>
<ProfilerRule id="uniqueid_41" comment="Store a String in Session Variable if SipRequest is INVITE">
<SipRule>
<And>
<Exists><SipRequestLine/></Exists>
<Equal>
<SipRequestLine field="Method"/>
<String value="INVITE"/>
</Equal>
</And>
</SipRule>
```

```

<Action>

<Assign>

<SessionVariable index="1" name="Status"/>

<String value="verified_1"/>

</Assign>

</Action>

</ProfilerRule>

<ProfilerRule id="uniqueid_42" comment="Insert a new proprietary header in response to INVITE">

<SipRule>

<And>

<Exists><SipStatusLine/></Exists>

<Equal>

<SipHeader Header="CSeq" field="Method"/>

<String value="INVITE"/>

</Equal>

</And>

</SipRule>

<Action>

<Insert>

<SipHeader Header="Other" OtherHeader="MyHeader"/>

<SessionVariable name="Status"/>

</Insert>

</Action>

</ProfilerRule>

</Profiler>

```

If the SIP message in the Profiler rule 'uniqueid\_41' is an INVITE then a session variable is created with index "1" and named 'status'. A String value 'verified\_1' is assigned to this variable.

If the SIP message in the Profiler rule 'uniqueid\_42' is a response to an INVITE then an unknown header 'MyHeader' is inserted into the SIP message with the value stored in the Session Variable named 'Status'.

## 10.3.9 Retrieving Data from Configuration Tables

```

<Profiler>

```

```
<ProfilerRule Id="uniqueid_43" Comment="Expires header is less than provisioned MIN-REGISTRATION-PERIOD">
  <SipRule>
    <And>
      <Exists><SipHeader Header="Expires"/></Exists>
      <Less>
        <SipHeader Header="Expires"/>
        <Configuration Field="ToBeDefined"/>
      </Less>
    </And>
  </SipRule>
  <Action>
    <RejectSipMessage StatusCode="423" Warning="Registration Interval Too Brief"/>
  </Action>
</ProfilerRule>
</Profiler>
```

In the example above, the Profiler rule checks if the Expires Header exists in the SIP message and if the value of this header is less than the configured minimum value for a registration period. If the value is less than the minimum value, the SIP request is rejected with status code 423.

## 10.3.10 Retrieving Data from IP Layer Fields

```
<Profiler>
  <ProfilerRule Id="1005" Comment="Invalid Record-Route Header">
    <SipRule>
      <And>
        <Exists><SipHeader Header="Record-Route" Index="1"/></Exists>
        <Or>
          <NotEqual>
            <SipHeader Header="Record-Route" Field="Address_Scheme"/>
            <String Value="Sip"/>
          </NotEqual>
        </Or>
      </And>
    </SipRule>
  </ProfilerRule>
</Profiler>
```

```
<SipParameter Header="Record-Route" Parameter="Transport"/>
<IP Field="InInterfaceProtocol"/>
</NotEqual>
<Not>
<ParseIPAddress Type="IPv4">
<SipHeader Header="Record-Route" Field="Address_Host"/>
</ParseIPAddress>
</Not>
</Or>
</And>
</SipRule>
<Action>
<RejectSipMessage StatusCode="406" Warning="Invalid Record-Route Header"/>
</Action>
</ProfilerRule>
</Profiler>
```

In the example above, the Profiler rule checks for all of the following conditions

1. A Record-Route Header exists in the SIP message

And

1. Either
  - o This Record-Route header's address scheme is not SIP OR
  - o This Record-Route header's transport parameter value is not equal to the transport parameter of the SCS's incoming interface
  - o This Record-Route header's address host field is not IPv4

If these conditions are met, then the Profiler rule rejects the SIP message with status code 406.

# 11. Session Detail Records

The BorderNet SBC contains a **Session Control Service (SCS)** component that takes "snapshots" of call sessions and writes these sessions to a file. This information is recorded in SDRs that can be sent to an external SDR destination to be used for billing or other purposes.

SDR writes a record of the session when one of the following events occurs:

- When a session terminates
- If a session is established but not complete, SDR writes a record when:
  - Media parameters are updated, such as a change of Codec or transferring the call from Audio to Fax.
  - There is a re-attempt, such as call forwarding or call redirection.
  - A termination attempt is unsuccessful, such as in rerouting or discovery.

Each record is a snapshot of the current ingress leg and the current egress leg.

The SDR Writer writes the data to a file. The following example shows a single SDR record within an SDR file:

```
1,bn4trail1,30886295,2010-05-28+00:05:36,2,1,1,1,2,2,,,, 244994-5160@172.29.8.58,244994,245014,"icid-
value=4b63434e4c27e7500000000000000004;orig-ioi=pcscf.Veraz.com;term-ioi=""172.29.8.58:5065"";
gcid=4b63434e4c27e7500000000000000004;hcid=4b63434e00000004",2010-05-28+00:05:36,2010-05-28+00:05:36,200,,,172.29.8.58:5060:UDP,172.29.14.104:5060:UDP,172.29.14.104:5061:UDP,172.29.8.58:5065:UDP,sip:31111@172.29.14.104:5060;user=phone,
sip:31111@172.29.8.58:5065;user=phone, sip:11111@172.29.8.58;user=phone, sip:11111@172.29.14.104:5061;user=phone,
sip:31111@172.29.14.104:5060;user=phone, sip:31111@172.29.14.104:5060;user=phone,,,, sip:11111@172.29.8.58:5060,
sip:11111_4c27e75000000003@172.29.14.104:5061, sip:172.29.8.58:5065,
sip:nabc_4c27e75000000005@172.29.14.104:5060,,,,,,,,,,,,,,,,,,,,,,1,,
```

By default, a new file is created every 10 seconds. This file contains the data for sessions that had a parameter change or were completed in that 10 second interval. The File Transport Service collects the files and sends the SDR data via FTPs or SCPs to the configured SDR destination.

In the event of an IBCF system service reset, platform reset, or software upgrade of a standalone BorderNet SBC, call data for active sessions will not be written to SDRs at the time of call completion. Care should be taken to perform these actions when there are no sessions active in the system.

For a high availability (HA) system, session data is mirrored on the standby platform to protect the integrity of the active session data necessary to complete SDR records. If the BorderNet SBC is deployed in an HA configuration, SDR files are generated from the Active platform. In the event of a failover, the new Active platform will generate the files.

The SDR file names are generated using the following format:

```
sdr.bn4k.[SystemName].[HostName].[date].[time].[fileSequence].[sequenceIndicator].v[SDRVersion].csv[.gz]
```

The [SystemName] is the name the user entered in the System Configuration screen. The following example shows the name of a compressed SDR file:

```
sdr.bn4k.NewYork2.bnsysA.20120217.121314.002233.0.v1.0.csv.gz
```

The BorderNet SBC provides 190 format fields in the SDR files, as listed in the below table.

---

**Note:**

When the SDR files are exported to a .csv file, the header row is printed in Excel.

---

Enum String	Field Disc. (Group - Name)	Format	Events
-------------	-------------------------------	--------	--------

Enum String	Field Disc. (Group - Name)	Format	Events	
SDRVersion	General - SDR Version	Char[4]	Interim, Stop	Current v
SDRSeqId	General - Record Sequence Identifier			Sequer
LocalTimeZone	General - BorderNet SBC Local Time Zone	Char[20]	Interim, Stop	
SoftwareVersion	General - BorderNet SBC Software Version that was running when generating these records	Char[20]	Interim, Stop	
TerminalName	General - Terminal Name	Char[20]	Interim, Stop	
TerminalId	General - Terminal Type	Char[2]	Interim, Stop	can be
AccountStatusType	General - Accounting status type	Enum(Start(1), Stop(2), Interim(3))	Interim, Stop	1-Star
AccountEventReason	General - Accounting Event Reason	Enum( Release, ReAttempt, MediaUpdate)	Interim, Stop	
SwitchOverTimeStmp	General - Timestamp when the session switchover occurred	String	Interim, Stop	Empty wh
AccountingTimestamp	General - Accounting event timestamp	String	Stop, Interim	UTC in UN
SDRSessionNumber	General - SDR Session Number	UInt64	Stop, Interim	Unique Borda
SDRRecordNumber	General - SDR Record within Sessions	UInt16	Stop, Interim	Unique index with Final Rec
ServiceType	General - Interface Type	Enum (lbcf)	Stop, Interim	
RoutingType	General - Routing Type	Enum(DstMsgRouting(1), lHopRouting(2),P2PRouting(3), PolicyRouting(4))	Stop, Interim	Messa BorderN Policy-ba:
AccountingSessionDuration	General - Accounting session time (Duration)	UInt32	Stop	From

Enum String	Field Disc. (Group - Name)	Format	Events	
SDRSessionStatus	General - SDR Session Status	Enum (Completed,InProgress,UnSuccessful)	Stop, Interim	Indicates an Answer When the In an Ingress leg
LRBT file name	The local ring back tone file name, played towards the originator peer	String		If the tone play
Spare				
Spare				
Spare				
Spare				
Spare				
Spare				
Spare				
Spare				
Spare				
IngressAudioMediaEncryption	Media	Char[30]		
IngressVideoMediaEncryption	Media	Char[30]		
InSpare3				
InSpare4				
InSpare5				
InSpare6				
InSpare7				
InSpare8				
InSpare9				
InRFactor				
EgressAudioMediaEncryption	Media	Char[30]		
EgressVideoMediaEncryption	Media	Char[30]		
EgSpare3				
EgSpare4				
EgSpare5				
EgSpare6				
EgSpare7				
EgSpare8				

Enum String	Field Disc. (Group - Name)	Format	Events	
EgSpare9				
EgRFactor				
GenSpare1				
GenSpare2				
GenSpare3				
GenSpare4				
GenSpare5				
GenSpare6				
GenSpare7				
GenSpare8				
GenSpare9				
GenSpare10				
PayloadTypeIW	SIP Dialog - Payload Type Inter-working	Boolean (1/0)	Interim, Stop	Indicates signific
IngressOriginatingTgld	General - Ingress Origination Tgrp Id	Char[20]	Interim, Stop	Ir
IngressOriginatingTrunkContext	General - Ingress Origination Trunk Context	Char[20]	Interim, Stop	Indic
EgressOriginatingTgld	General - Egress Originating Tgrp Id	Char[20]	Interim, Stop	Indicates
EgressOriginatingTrunkContext	General - Egress Originating Trunk Context	Char[20]	Interim, Stop	Indicates
IngressDestinationTgld	General - Ingress Destination Tgrp Id	Char[20]	Interim, Stop	Indicates
IngressDestinationTrunkContext	General - Ingress Destination Trunk Context	Char[20]	Interim, Stop	Indicate
EgressDestinationTgld	General - Egress Destination Tgrp Id	Char[20]	Interim, Stop	Indicate:
EgressDestinationTrunkContext	General - Egress Destination Trunk Context	Char[20]	Interim, Stop	Indicates i
Egress3xxDestinationTgld	General - Egress Destination Tgrp Id in 3xx response	Char[20]	Interim, Stop	Indica



Enum String	Field Disc. (Group - Name)	Format	Events	
Egress3xxDestinationTrunkContext	General - Egress Destination Trunk Context received in 3xx response	Char[20]	Interim, Stop	Indicates
EmergencyCall	General - Emergency Call	Boolean (1/0)	Interim, Stop	
IngressSigProtocol	Signaling - Ingress Signaling Protocol	Enum (SIP, H323)	Stop, Interim	
IngressQ850CauseCodeValue	Signaling - The Q.850 Cause Code Value sent or received at the ingress side of the session	Uint16	Stop	This is th This can't the re:
IngressSigRemoteAddress	Signaling - Incoming remote address	Char[20]	Stop, Interim	
IngressSigLocalAddress	Signaling - Incoming local address	Char[20]	Stop, Interim	
IngressSigReqLine	Signaling - Incoming request URI	Char[150]	Stop, Interim	Can be cl be prior
IngressSigFromHeader	Signaling - Incoming From header	Char[150]	Stop, Interim	Can be c
IngressSigToHeader	Signaling - Incoming To header	Char[150]	Stop, Interim	Can be c
IngressSigAsserted	Signaling - Incoming P-Asserted-Id	Char[150]	Stop, Interim	Car
IngressSigPreferred	Signaling - Incoming p-preferred-identity	Char[150]	Stop, Interim	Car
IngressSigSourceContact	Signaling - Incoming source contact	Char[150]	Stop, Interim	Can be c
IngressSigLocalContact	Signaling - Contact of the Local Point	Char[150]	Stop, Interim	Can be c
EgressSigProtocol	Signaling - Egress Signaling Protocol	Enum (SIP, H323)	Stop, Interim	BorderN
EgressQ850CauseCodeValue	Signaling - The Q.850 Cause Code Value sent or received at the egress side of the session	Uint16	Stop	This is th This can't the re:

Enum String	Field Disc. (Group - Name)	Format	Events	
OutSigLocalAddr	Signaling - Outgoing local address	Char[20]	Stop, Interim	
OutSigDstAddr	Signaling - Outgoing destination address	Char[20]	Stop, Interim	
OutSigReqLine	Signaling - Outgoing request URI	Char[150]	Stop, Interim	Can be ch
OutSigFrom	Signaling - Outgoing From header	Char[150]	Stop, Interim	Can be ch
OutSigTo	Signaling - Outgoing To header	Char[150]	Stop, Interim	Can be ch
OutSigAsserted	Signaling - Outgoing P- Asserted-Id	Char[150]	Stop, Interim	Can
OutSigPreferred	Signaling - Outgoing p- preferred- identity	Char[150]	Stop, Interim	Can
OutSigLocalContct	Signaling - Outgoing local contact	Char[150]	Stop, Interim	Can be c
OutSigDstContct	Signaling - Outgoing Called contact	Char[150]	Stop, Interim	Can be c
IngressPeer	SIP Dialog - Ingress Peer Identifier	Uint32	Stop, Interim	BorderNe
IngressInterface	SIP Dialog Ingress Interface Identifier	Uint32	Stop, Interim	Br
IngressParamProfile	SIP Dialog - Ingress Parameter Profile	Uint32	Stop, Interim	Br
IngressServiceProfile	SIP Dialog	Uint32	Stop, Interim	BorderN
IngressSecurityProfile	SIP Dialog	Uint32	Stop, Interim	BorderN
IngressMediaProfile	SIP Dialog	Uint32	Stop, Interim	BorderN
IngressTLSProfile	SIP Dialog	Uint32	Stop, Interim	BorderN
IngressAdvPolicy	SIP Dialog	Uint32	Stop, Interim	BorderN
IngressIncomingSipMsgProfiler	SIP Dialog	String	Stop, Interim	BorderNe

Enum String	Field Disc. (Group - Name)	Format	Events	
IngressOutgoingSipMsgProfiler	SIP Dialog	String	Stop, Interim	BorderNe
IngressSipCallId	SIP Dialog - SIP Call-ID	Char[150]	Stop, Interim	
IngressSipFromTag	SIP Dialog - SIP From Tag	Char[150]	Stop, Interim	
IngressSipToTag	SIP Dialog - SIP To Tag	Char[150]	Stop, Interim	
IngressTimeStampINVITE	SIP Dialog - Incoming INVITE Timestamp	UInt16	Stop, Interim	
IngressTimeStamp18x	Sip Dialog - Outgoing 18x Timestamp	UInt16	Stop, Interim	
IngressAlertingSent	SIP Dialog - Outgoing 180 Ringing Timestamp	String	Stop, Interim	
IngressAnswerSent	SIP Dialog - Outgoing 200 OK Answer Timestamp	String	Stop, Interim	
IngressReleaseTimeStamp	SIP Dialog - Time Stamp when Release happened	String	Stop, Interim	This is
IngressReleaseCompleteTimeStamp	SIP Dialog - Time Stamp when Release Complete happened	String	Stop, Interim	This is Note: If SC
IngressReleaseSent	SIP Dialog - Release Message Type	ReleaseType_None = 0, ReleaseType_3xx, ReleaseType_4xx, ReleaseType_5xx, ReleaseType_6xx, ReleaseType_CANCEL, ReleaseType_BYE, ReleaseType_200OKBYE, ReleaseType_487, ReleaseType_ACK, ReleaseType_Internal Error	Stop	
IngressReleaseReceived	SIP Dialog - Release Message Type	ReleaseType_None = 0, ReleaseType_3xx, ReleaseType_4xx, ReleaseType_5xx, ReleaseType_6xx, ReleaseType_CANCEL, ReleaseType_BYE, ReleaseType_200OKBYE, ReleaseType_487, ReleaseType_ACK, ReleaseType_Internal Error	Stop	
IngressReleaseCodeValue	SIP Dialog - Release Response Value	UInt16	Stop	Capturing

Enum String	Field Disc. (Group - Name)	Format	Events	
IngressInternalCauseCode	SIP Dialog - Internal Release Code Generated by the System	Uint16	Stop	This is tl session when t
EgressPeer	SIP Dialog - Egress Peer Identifier	Uint32	Stop, Interim	BorderN
EgressInterface	SIP Dialog - Egress Interface Identifier	Uint32	Stop, Interim	
EgressParamProfile	SIP Dialog - Egress Parameter Profile	Uint32	Stop, Interim	
EgressServiceProfile	SIP Dialog	Uint32	Stop, Interim	BorderN
EgressSecurityProfile	SIP Dialog	Uint32	Stop, Interim	BorderN
EgressMediaProfile	SIP Dialog	Uint32	Stop, Interim	BorderN
EgressTLSProfile	SIP Dialog	Uint32	Stop, Interim	BorderN
EgressAdvPolicy	SIP Dialog	Uint32	Stop, Interim	BorderN expose
EgressIncSipMsgProfiler	SIP Dialog	String	Stop, Interim	BorderNe
EgressOutSipMsgProfiler	SIP Dialog	String	Stop, Interim	BorderNe
EgressSipCallId	SIP Dialog - SIP Call-ID	Char[150]	Stop, Interim	
EgressSipFromTag	SIP Dialog - SIP From Tag	Char[150]	Stop, Interim	
EgressSipToTag	SIP Dialog - SIP To Tag	Char[150]	Stop, Interim	
EgressTimeStmpInvite	SIP Dialog - Outgoing INVITE Timestamp	Uint16	Stop, Interim	I
EgressTimeStmp18xRcvd	Sip Dialog - Incoming18x Timestamp	Uint16	Stop, Interim	
EgressAlertingReceived	SIP Dialog - Incoming 180 Ringing Timestamp	String	Stop, Interim	
EgressAnswerReceived	SIP Dialog - Incoming 200 OK Answer Timestamp	String	Stop, Interim	

Enum String	Field Disc. (Group - Name)	Format	Events	
EgressReleaseTimeStamp	SIP Dialog - Time Stamp when Release happened	String	Stop, Interim	This is
EgressReleaseCompleteTimeStamp	SIP Dialog - Time Stamp when Release Complete happened	String	Stop, Interim	This is  Note: If SC
EgressReleaseSent	SIP Dialog - Release Type	ReleaseType_None = 0, ReleaseType_3xx, ReleaseType_4xx, ReleaseType_5xx, ReleaseType_6xx, ReleaseType_CANCEL, ReleaseType_BYE, ReleaseType_200OKBYE, ReleaseType_487, ReleaseType_ACK, ReleaseType_Internal Error	Stop, Interim	
EgressReleaseRcvd	SIP Dialog - Release Type	ReleaseType_None = 0, ReleaseType_3xx, ReleaseType_4xx, ReleaseType_5xx, ReleaseType_6xx, ReleaseType_CANCEL, ReleaseType_BYE, ReleaseType_200OKBYE, ReleaseType_487, ReleaseType_ACK, ReleaseType_Internal Error	Stop, Interim	
EgressReleaseCodeValue	SIP Dialog - Release Cause Value	UInt16	Stop, Interim	Capturing
EgressTimeStmpResp	Sip Dialog - First Response Timestamp	UInt16	Stop, Interim	
EgressResponseCode	Sip Dialog - INVITE Response Code	UInt16	Stop, Interim	
EgressResponseWarning	Sip Dialog - INVITE Response Warning	Char[150]	Stop, Interim	
EgressInternalCauseCodeValue				
CallingPartyUser	SIP Dialog - The Calling Party User Id associated with the entire session	String	Stop, Interim	BorderNe which the Sip-F
CalledPartyUser	SIP Dialog - The Called Party User Id associated with the entire session	String	Stop, Interim	BorderNe which the

Enum String	Field Disc. (Group - Name)	Format	Events	
OrigCallingPartyUser	SIP Dialog - The Original Calling Party User Id that arrived with the session	String	Stop, Interim	This is
OrigCalledPartyUser	SIP Dialog - The Original Called Party User Id associated with the entire session	String	Stop, Interim	
GenericParameterSipProf	SIP Dialog - This is the generic parameter set by the SIP Profiler to be used in advanced policy	String	Stop, Interim	
PolicySipMsgProfiler (Optional)	SIP Dialog	String	Stop, Interim	Border treatm
PChargingVector	SIP Dialog - P-Charging-Vector	Char[150]	Stop, Interim	Example: ioi=home the file a ioi=""172.
PChargingFuncAddr	SIP Dialog - P-Charging-Function-Addresses	Char[150]	Stop, Interim	Example: ecf=192.1
MediaInterception	SIP Dialog - If BorderNet SBC intercepts media per configuration	Enum (0,1,2,3)	Stop, Interim	
MediaOfferSentTimeStamp	Sip Dialog - SDP Offer Advertised by BorderNet SBC	String	Stop, Interim	When SD intercept contain medi importan pi As Bor
MediaAnswerSentTimeStamp (Optional)	Sip Dialog - SDP Answer Advertised by BorderNet SBC	String	Stop, Interim	When S being inte cor Since th importan As Bor
ReleaseType	Sip Dialog - Release Type	1 = Normal 2 = Cancel 3 = Dead Call 4 = Internal Release	Stop	This is wh releas Negative I Call - Su Int

Enum String	Field Disc. (Group - Name)	Format	Events	
InAudioSentOutCodecListProfileId	Media - On the ingress side, this is the list of codec profile id that the system sent out as either offer/answer	Char[100]	Stop, Interim	String of " be part of
InAudioSentOutCodecListType	Media - On the ingress side, this is the list of codec types (Media Subtype) that the system sent out as either offer/answer	Char[200]	Stop, Interim	String of the actua
InAudioRcvdCodecListProfileId	Media - On the ingress side, this is the list of codec profile id that the system received as either offer/answer	Char[100]	Stop, Interim	String of " be part of
InAudioRcvdCodecListType	Media - On the ingress side, this is the list of codec types (Media Subtype) that the system received as either offer/answer	Char[200]	Stop, Interim	String of the actua
InAudioPType	Media - Incoming voice payload type	Char[20]	Stop, Interim	Media Su chosen
InAudioSrcAddr (Optional)	Media - Incoming voice udp/rtp source address	Char[20]	Stop, Interim	
InAudioLocalAddr	Media - Incoming voice udp/rtp destination address	Char[20]	Stop, Interim	
InAudioSts	Media - Incoming voice statistics	Char[100]	Stop, Interim	String o
InImageSentOutCodecListProfileId	Media - On the ingress side, this is the list of codec profile id that the system sent out as either offer/answer	Char[100]	Stop, Interim	String of " be part of

Enum String	Field Disc. (Group - Name)	Format	Events	
InImageSentOutCodecListType	Media - On the ingress side, this is the list of codec types (Media Subtype) that the system sent out as either offer/answer	Char[200]	Stop, Interim	String of
InImageRcvdCodecListProfileId	Media - On the ingress side, this is the list of codec profile id that the system received as either offer/answer	Char[100]	Stop, Interim	String of " be part of
InImageRcvdCodecListType	Media - On the ingress side, this is the list of codec types (Media Subtype) that the system received as either offer/answer	Char[200]	Stop, Interim	String of
InImagePType	Media - Incoming image payload type	Char[20]	Stop, Interim	Media Su ingress sic
InImageSrcAddr	Media - Incoming image source address	Char[20]	Stop, Interim	
InImageLocalAddr	Media - Incoming image local address	Char[20]	Stop, Interim	
InImageSts	Media - Incoming image statistics	Char[100]	Stop, Interim	String o
InVideoSentOutCodecListProfileId	Media - On the ingress side, this is the list of codec profile id that the system sent out as either offer/answer	Char[100]	Stop, Interim	String of " be part of
InVideoSentOutCodecListType	Media - On the ingress side, this is the list of codec types (Media Subtype) that the system sent out as either offer/answer	Char[200]	Stop, Interim	String , display i



Enum String	Field Disc. (Group - Name)	Format	Events	
InVideoRcvdCodecListProfileId	Media - On the ingress side, this is the list of codec profile id that the system received as either offer/answer	Char[100]	Stop, Interim	String of " be part of
InVideoRcvdCodecListType	Media - On the ingress side, this is the list of codec types (Media Subtype) that the system received as either offer/answer	Char[200]	Stop, Interim	String of " display i
InVideoPType (Optional)	Media - Incoming video payload type	Char[20]	Stop, Interim	Media S chosen i
InVideoSrcAddr	Media - Incoming video source address	Char[20]	Stop, Interim	
InVideoLocalAddr	Media - Incoming video local address	Char[20]	Stop, Interim	
InVideoSts	Media - Incoming image statistics	Char[100]	Stop, Interim	String of
OutAudioSentOutCodecListProfileId	Media - On the egress side, this is the list of codec profile id that the system sent out as either offer/answer	Char[100]	Stop, Interim	String of " be part of
OutAudioSentOutCodecListType	Media - On the egress side, this is the list of codec types (Media Subtype) that the system sent out as either offer/answer	Char[200]	Stop, Interim	String of the actua
OutAudioRcvdCodecListProfileId	Media - On the egress side, this is the list of codec profile id that the system received as either offer/answer	Char[100]	Stop, Interim	String of " be part of

Enum String	Field Disc. (Group - Name)	Format	Events	
OutAudioRcvdCodecListType	Media - On the egress side, this is the list of codec types (Media Subtype) that the system received as either offer/answer	Char[200]	Stop, Interim	String of the actua
OutVoicePType	Media - Outgoing voice payload type	Char[20]	Stop, Interim	Media St chosen on
OutAudioLocalAddr	Media - Outgoing voice source address	Char[20]	Stop, Interim	
OutAudioDstAddr	Media - Outgoing voice source address	Char[20]	Stop, Interim	
OutAudioSts	Media - Outgoing voice statistics	Char[100]	Stop, Interim	String o
OutImageSentOutCodecListProfileId	Media - On the ingress side, this is the list of codec profile id that the system sent out as either offer/answer	Char[100]	Stop, Interim	String of " be part of
OutImageSentOutCodecListType	Media - On the ingress side, this is the list of codec types (Media Subtype) that the system sent out as either offer/answer	Char[200]	Stop, Interim	String of
OutImageRcvdCodecListProfileId	Media - On the ingress side, this is the list of codec profile id that the system received as either offer/answer	Char[100]	Stop, Interim	String of " be part of
OutImageRcvdCodecListType	Media - On the ingress side, this is the list of codec types (Media Subtype) that the system received as either offer/answer	Char[200]	Stop, Interim	String of
OutImagePType	Media - Outgoing image payload type	Char[20]	Stop, Interim	Media St egress sid

Enum String	Field Disc. (Group - Name)	Format	Events	
OutImageLocalAddr	Media - Outgoing image local address	Char[20]	Stop, Interim	
OutImageDstAddr	Media - Outgoing image destination address	Char[20]	Stop, Interim	
OutImageSts	Media - Outgoing image statistics	Char[100]	Stop, Interim	String o
OutVideoSentOutCodecListProfileId	Media - On the ingress side, this is the list of codec profile id that the system sent out as either offer/answer	Char[100]	Stop, Interim	String of " be part of
OutVideoSentOutCodecListType	Media - On the ingress side, this is the list of codec types (Media Subtype) that the system sent out as either offer/answer	Char[200]	Stop, Interim	String i display i
OutVideoRcvdCodecListProfileId	Media - On the ingress side, this is the list of codec profile id that the system received as either offer/answer	Char[100]	Stop, Interim	String of " be part of
OutVideoRcvdCodecListType	Media - On the ingress side, this is the list of codec types (Media Subtype) that the system received as either offer/answer	Char[200]	Stop, Interim	String i display i
OutVideoPType	Media - Outgoing video payload type	Char[20]	Stop, Interim	Media Su chosen on
OutVideoLocalAddr	Media - Outgoing video source address	Char[20]	Stop, Interim	
OutVideoDstAddr	Media - Outgoing video source address	Char[20]	Stop, Interim	
OutVideoSts	Media - Outgoing video statistics	Char[100]	Stop, Interim	String o

Enum String	Field Disc. (Group - Name)	Format	Events	
DiameterSessionId	Diameter - DIAMETER Session ID	UInt32	Strt, Int, Stp	
ServerIPAddress	Diameter - Server IP address	Char[20]	Strt, Int, Stp	
ServerPort	Diameter - Server port			
IngressSecurityType (Optional)	Signaling - Ingress Call Leg Security Type	1.Unsecured 2. IP Sec. 3. TLS	Interim, Stop	Ir
IngressSecurityProtocol (O)	Signaling - Ingress Call Leg Security Protocol	1. AH 2. ESP 3. TLS	Interim, Stop	Ind
EgressSecurityType (Optional)	Signaling - Egress Call Leg Security Type	1.Unsecured 2. IP Sec. 3. TLS		II
EgressSecurityProtocol (Optional)	Signaling - Egress Call Leg Security Protocol	1. AH 2. ESP 3. TLS	Interim, Stop	Ind
SIPICall	Signaling			
IS_TRANSCODED_ CALL (Optional)	SIP Dialog - Indication on whether transcoding was performed by Border-Net on the call	UInt16	Start, Interim, Stop	The value

Table 15: SDR Fields

## Appendix A Port Information

The BorderNet SBC uses standard ports as follows:

- 21: FTP
- 22: SSH
- 23: Telnet
- 80: HTTP (for the sole purpose of redirecting to https)
- 443: HTTPS

The BorderNet SBC uses non-standard ports as follows:

- 2010: Tracing using RPCAP control session
- The data port that will be used for sending RPCAP packets is requested from the OS dynamically and sent to the client in a response to an Open or Start response message.

All of the above ports can be changed in the XMLs. These ports are not open by default, with one exception: 80/443 is open to same subnet only. The rest of the listed above ports must be both enabled and have an ACL created.

## Appendix B Configuration Limits

The following configuration limits apply to the BorderNet SBC:

Entity	Limit
VLANs	1024
Maximum IP Addresses (for Signaling and Media)	2048
Maximum IP Addresses per VLAN	254
SIP Interfaces	512
H.323 Interfaces	512
Peers (these are configured peers)	2048
Media Profiles	2048
Security Profiles	2048
Service Profiles	2048
Simultaneous Dashboard Access Clients	15
Software Load Versions	5

Table 16: Configuration Limits

#### Appendix C SIP Headers and Parameters

SIP Headers			
Allow	Min-SE	P-DCS-Trace-Party-ID	RequestLine
Allow-Events	Other	P-Media-Authorization	Retry-After
Authentication-Info	P-Access-Network-Info	P-Preferred-Identity	Route
Authorization	P-Answer-State	P-Profile-Key	Security-Client
CSeq	P-Asserted-Identity	P-User-Database	Security-Server
Call-ID	P-Associated-Uri	P-Visited-Network-ID	Security-Verify
Contact	P-Called-Party-ID	Path	Service-Route
Content-Length	P-Charging-Function-Addresses	Proxy-Authorization	Session-Expires
Content-Type	P-Charging-Vector	Rack	StatusLine
Date	P-DCS-Billing-Info	Record-Route	Subscription-State
Event	P-DCS-LAES	Referred-By	Supported
Expires	P-DCS-OSPS	Refer-To	To
From	P-DCS-Redirect	Replaces	Via
Max-Forwards	Feature-Caps		

Table 17: SIP Headers

#### Appendix D SIP Parameters

SIP Header	Field	Type	Field	Type
SipRequestLine	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean

SIP Header	Field	Type	Field	Type
	Address-Host	String	Address-Transport	Enum
	Address-PhoneNum	String	Address-Uri	String
	Address-Port	Number	Address-User	String
	Address-Protocol	Enum	Method	Enum
SipStatusLine	Code	Number	Reason	String
Allow	Method	Enum		
Allow-Events	Event-Package	String	Event-Template	String
Authentication-Info	Cnonce	String	Nc	Number
	Nextnonce	String	Qop	Enum
	Rspauth	String		
Authorization	AKAv	Number	Opaque	String
	Algorithm	Enum	Qop	Enum
	Auts	String	Realm	String
	Cnonce	String	Response	String
	Integrity-Protected	Enum	Scheme	Enum
	Nc	Number	Uri	String
	Nonce	String	Username	String
Contact	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
	Address-Protocol	Enum	Star	Boolean
Content-Type	M-SubType	Enum	M-Type	Enum
CSeq	Method	Enum	Step	Number
Date	Day	Number	Seconds	Number
	Hour	Number	WkDay	Enum
	Minute	Number	Year	Number
	Month	Enum		
Event	Event-Package	String	Event-Template	String
Expires	Delta-Seconds	Number		
Feature-Caps	+g.3gpp.trf	String	+g.3gpp.loopback	String
From	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobaNum	String	Address-Transport	Enum

SIP Header	Field	Type	Field	Type
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
	Address-Protocol	Enum		
MaxForwards	Value	Number		
Min-SE	Delta-Seconds	Number		
P-Access-Network-Info	Access-Type	Enum		
P-Answer-State	Answer-Type	Enum		
P-Asserted-Identity	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
	Address-Protocol	Enum	Display-Name	String
P-Associated-URI	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
	Address-Protocol	Enum	Display-Name	String
Path	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
P-Called-Party-ID	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
	Address-Protocol	Enum	Display-Name	String
P-Charging-Vector	Icid-Value	String		

SIP Header	Field	Type	Field	Type
P-DCS-Billing-Info	Billing-Correlation-ID	String	FEID-Host	String
	FEID	String		
P-DCS-LAES	Sig-Host	String	Sig-Port	Number
P-DCS-OSPS	Tag	Enum		
P-DCS-Redirect	Called-ID	SipAddress		
P-DCS-Trace-Party-ID	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
	Address-Protocol	Enum	Display-Name	String
P-Media-Authorization	Token	String		
P-Preferred-Identity	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
	Address-Protocol	Enum	Display-Name	String
P-Profile-Key	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
	Address-Protocol	Enum		
P-User-Database	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
	Address-Protocol	Enum		
P-Visited-Network-ID	VNetwork-Spec	String		
RAck	CSeq-Num	Number	Response-Num	Number



SIP Header	Field	Type	Field	Type
	Method	Enum		
Reason	Protocol	Enum		
Record-Route	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
Referred-By	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
	Address-Protocol	Enum		
Refer-To	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
	Address-Protocol	Enum		
Replaces	CallId	String		
Retry-After	Comment	String	Delta-Seconds	Number
Route	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
	Address-Protocol	Enum		
RSeq	Response-Num	Number		
Security-Client	Mechanism	Enum		
Security-Server	Mechanism	Enum		
Security-Verify	Mechanism	Enum		
Service-Route	Address	SipAddress	Address-Scheme	Enum

SIP Header	Field	Type	Field	Type
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
Session-Expires	Delta-Seconds	Number		
Subscription-State	Substate	Enum		
To	Address	SipAddress	Address-Scheme	Enum
	Address-Global	Boolean	Address-Secure	Boolean
	Address-GlobalNum	String	Address-Transport	Enum
	Address-Host	String	Address-Uri	String
	Address-PhoneNum	String	Address-User	String
	Address-Port	Number	Address-Username	String
	Address-Protocol	Enum		
Via	Host	String	Transport	Enum
	Port	Number		

Table 18: SIP Header Fields

SIP Header	Parameter	Type	Parameter	Type
SipRequestLine	Cic	String	Npdi	Boolean
	CicContext	String	Orig	Boolean
	Comp	Enum	Other	String
	Context	String	PostDial	String
	Cpc	Enum	Rn	String
	Extension	String	RnContext	String
	Headers	String	TokenizedBy	String
	IsdnSubAddr	String	Transport	Enum
	Lr	Boolean	Ttl	Number
	MAddr	String	User	Enum
	Method	Enum		
SipStatusLine	SipStatusLine	None		
Authorization	Other	String		
Contact	Action	Enum	Pub-Gruu	String
	Actor	String	Q	String
	Application	String	Reg-Id	Number
	Audio	String	Rn	String

SIP Header	Parameter	Type	Parameter	Type
	Automata	String	RnContext	String
	Cic	String	Schemes	String
	CicContext	String	Sip-Instance	String
	Class	String	Temp-Gruu	String
	Comp	Enum	Text	String
	Context	String	TokenizedBy	String
	Control	String	Transport	Enum
	Cpc	Enum	Ttl	Number
	Data	String	Type	String
	Description	String	User	Enum
	Duplex	String	Video	String
	Events	String	Lr	Boolean
	Expires	Number	MAddr	String
	Extension	String	Method	Enum
	Extensions	String	Methods	String
	Headers	String	Mobility	String
	IsdnSubAddr	String	Npdi	Boolean
	IsFocus	String	Orig	Boolean
	Language	String	Other	String
	PostDial	String	Priority	String
Content-Type	Base	String	Start	String
	Boundary	String	Version	String
	Other	String		
Event	Id	String	Other	String
From	Cic	String	MAddr	String
	CicContext	String	Method	Enum
	Comp	Enum	Npdi	Boolean
	Context	String	Orig	Boolean
	Cpc	Enum	Other	String
	Extension	String	PostDial	String
	Headers	String	Rn	String
	IsdnSubAddr	String	RnContext	String
	Lr	Boolean	Tag	String
	TokenizedBy	String	Ttl	Number
Transport	Enum	User	Enum	
Min-SE	Other	String		

SIP Header	Parameter	Type	Parameter	Type
P-Access-Network-Info	Cgi-3Gpp	String	Network-Provided	Boolean
	Ci-3Gpp2	String	Other	String
	Dsl-Location	String	Utran-Cell-Id-3Gpp	String
	I-Wlan-Node-Id	String		
P-Answer-State	Other	String		
P-Asserted-Identity	Cic	String	Method	Enum
	CicContext	String	Npdi	Boolean
	Comp	Enum	Orig	Boolean
	Context	String	Other	String
	Cpc	Enum	PostDial	String
	Extension	String	Rn	String
	Headers	String	RnContext	String
	IsdnSubAddr	String	TokenizedBy	String
	Lr	Boolean	Transport	Enum
	MAddr	String	Ttl	Number
	User	Enum		
P-Associated-ID	Cic	String	Lr	Boolean
	CicContext	String	MAddr	String
	Comp	Enum	Method	Enum
	Context	String	Npdi	Boolean
	Cpc	Enum	Orig	Boolean
	Extension	String	Other	String
	Headers	String	PostDial	String
	IsdnSubAddr	String	Rn	String
	Transport	Enum	RnContext	String
	Ttl	Number	TokenizedBy	String
	User	Enum		
Path	Cic	String	Npdi	Boolean
	CicContext	String	Orig	Boolean
	Comp	Enum	Other	String
	Context	String	PostDial	String
	Cpc	Enum	Rn	String
	Extension	String	RnContext	String
	Headers	String	TokenizedBy	String
	IsdnSubAddr	String	Transport	Enum
	Lr	Boolean	Ttl	Number

SIP Header	Parameter	Type	Parameter	Type
	MAddr	String	User	Enum
	Method	Enum		
P-Called-Party-ID	Cic	String	Npdi	Boolean
	CicContext	String	Orig	Boolean
	Comp	Enum	Other	String
	Context	String	PostDial	String
	Cpc	Enum	Rn	String
	Extension	String	RnContext	String
	Headers	String	TokenizedBy	String
	IsdnSubAddr	String	Transport	Enum
	Lr	Boolean	Ttl	Number
	MAddr	String	User	Enum
	Method	Enum		
P-Charging-Function-Addresses	Ccf-List	String	Other	String
	Ecf-List	String		
P-Charging-Vector	Bcid	String	Other	String
	Bras	String	Packetcable-Multimedia	Boolean
	Ggsn	String	Pdg	Boolean
	Ggsn-Auth-Token	String	Term-loi	String
	Icid-Generated-At	String	Xdsl-Auth-Token	String
	Orig-loi	String		
P-DCS-Billing-Info	Called	SipAddress	Other	String
	Calling	SipAddress	RKS-Group-ID	String
	Charge	SipAddress	Routing	SipAddress
	LocRoute	SipAddress		
P-DCS-LAES	Content-Host	String	Key	String
	Content-Port	Number	Other	String
P-DCS-Redirect	Count	Number	Redirector-Uri	SipAddress
	Other	String		
P-DCS-Trace-Party-ID	Cic	String	Method	Enum
	CicContext	String	Npdi	Boolean
	Comp	Enum	Orig	Boolean
	Context	String	PostDial	String
	Cpc	Enum	Rn	String
	Extension	String	RnContext	String
	Headers	String	TokenizedBy	String

SIP Header	Parameter	Type	Parameter	Type
	IsdnSubAddr	String	Transport	Enum
	Lr	Boolean	Ttl	Number
	MAddr	String	User	Enum
P-Preferred-Identity	Cic	String	Npdi	Boolean
	CicContext	String	Orig	Boolean
	Comp	Enum	Other	String
	Context	String	PostDial	String
	Cpc	Enum	Rn	String
	Extension	String	RnContext	String
	Headers	String	TokenizedBy	String
	IsdnSubAddr	String	Transport	Enum
	Lr	Boolean	Ttl	Number
	MAddr	String	User	Enum
	Method	Enum		
P-Profile-Key	Cic	String	Npdi	Boolean
	CicContext	String	Orig	Boolean
	Comp	Enum	PostDial	String
	Context	String	Rn	String
	Cpc	Enum	RnContext	String
	Extension	String	TokenizedBy	String
	Headers	String	Transport	Enum
	IsdnSubAddr	String	Ttl	Number
	Lr	Boolean	User	Enum
	MAddr	String	Method	Enum
	P-User-Database	Cic	String	Orig
CicContext		String	Other	String
Comp		Enum	PostDial	String
Context		String	Rn	String
Cpc		Enum	RnContext	String
Extension		String	TokenizedBy	String
Headers		String	Transport	Enum
IsdnSubAddr		String	Ttl	Number
Lr		Boolean	User	Enum
MAddr		String	Npdi	Boolean
Method		Enum		
P-Visited-Network-ID	Other	String		

SIP Header	Parameter	Type	Parameter	Type
Record-Route	Cic	String	Npdi	Boolean
	CicContext	String	Orig	Boolean
	Comp	Enum	Other	String
	Context	String	PostDial	String
	Cpc	Enum	Rn	String
	Extension	String	RnContext	String
	Headers	String	TokenizedBy	String
	IsdnSubAddr	String	Transport	Enum
	Lr	Boolean	Ttl	Number
	MAddr	String	User	Enum
	Method	Enum		
Referred-By	Cic	String	Method	Enum
	CicContext	String	Npdi	Boolean
	Cid	String	Orig	Boolean
	Comp	Enum	Other	String
	Context	String	PostDial	String
	Cpc	Enum	Rn	String
	Extension	String	RnContext	String
	Headers	String	TokenizedBy	String
	IsdnSubAddr	String	Transport	Enum
	Lr	Boolean	Ttl	Number
	Refer-To	Cic	String	Npdi
CicContext		String	Orig	Boolean
Comp		Enum	Other	String
Context		String	PostDial	String
Cpc		Enum	Rn	String
Extension		String	RnContext	String
Headers		String	TokenizedBy	String
IsdnSubAddr		String	Transport	Enum
Lr		Boolean	Ttl	Number
MAddr		String	User	Enum
Method		Enum		
Replaces	EarlyFlag	Boolean	Other	String
	FromTag	String	ToTag	String
Retry-After	Duration	Number	Other	String
Route	Cic	String	Npdi	Boolean

SIP Header	Parameter	Type	Parameter	Type
	CicContext	String	Orig	Boolean
	Comp	Enum	Other	String
	Context	String	PostDial	String
	Cpc	Enum	Rn	String
	Extension	String	RnContext	String
	Headers	String	TokenizedBy	String
	IsdnSubAddr	String	Transport	Enum
	Lr	Boolean	Ttl	Number
	MAddr	String	User	Enum
	Method	Enum		
Security-Client	Alg	Enum	Port-C	Number
	D-Alg	Enum	Port-S	Number
	D-Qop	Enum	Prot	Enum
	D-Ver	String	Q	String
	EAlg	Enum	Spi-C	Number
	Mod	Enum	Spi-S	Number
	Other	String		
Security-Server	Alg	Enum	Port-C	Number
	D-Alg	Enum	Port-S	Number
	D-Qop	Enum	Prot	Enum
	D-Ver	String	Q	String
	EAlg	Enum	Spi-C	Number
	Mod	Enum	Spi-S	Number
	Other	String		
Security-Verify	Alg	Enum	Port-C	Number
	D-Alg	Enum	Port-S	Number
	D-Qop	Enum	Prot	Enum
	D-Ver	String	Q	String
	EAlg	Enum	Spi-C	Number
	Mod	Enum	Spi-S	Number
	Other	String		
Service-Route	Cic	String	Orig	Boolean
	CicContext	String	Other	String
	Comp	Enum	PostDial	String
	Context	String	Rn	String
	Cpc	Enum	RnContext	String



SIP Header	Parameter	Type	Parameter	Type
	Extension	String	TokenizedBy	String
	Headers	String	Transport	Enum
	IsdnSubAddr	String	Ttl	Number
	Lr	Boolean	User	Enum
	MAddr	String	Npdi	Boolean
Session-Expires	Method	Enum		
	Other	String	Refresher	Enum
Subscription-State	Expires	Number	Reason	String
	Other	String	Retry-After	Number
To	Cic	String	Npdi	Boolean
	CicContext	String	Orig	Boolean
	Comp	Enum	Other	String
	Context	String	PostDial	String
	Cpc	Enum	Rn	String
	Extension	String	RnContext	String
	Headers	String	Tag	String
	IsdnSubAddr	String	TokenizedBy	String
	Lr	Boolean	Transport	Enum
	MAddr	String	Ttl	Number
	Method	Enum	User	Enum
Via	Other	String	Branch	String
	Received	String	Comp	Enum
	Rport	Number	Hidden	Boolean
	Ttl	Number	MAddr	String

Table 19: SIP Parameters

## Appendix E IP Fields

Field	Type	Field	Type
InInterfaceId	Number	InInterfaceDomain	String
InInterfaceName	String	InInterfaceSIPconnect	Boolean
InInterfaceType	Enum	InInterfaceSIPconnectType	Enum
InInterfaceIP	String	InInterfaceIMS	Boolean
InInterfaceNetworkType	Enum	InInterfaceCIDBase	String
InInterfacePType	Enum	InInterfaceNetwork	String
InInterfacePort	Number	InInterfaceSubTraffic	Boolean
InInterfaceProtocol	Enum	InInterfaceAllowOnlyAssocPeers	Boolean

Field	Type	Field	Type
InInterfaceAccType	Enum	OutInterfaceType	Enum
InInterfaceOperOd	String	OutInterfaceIP	String
InInterfaceSigTos	Number	OutInterfaceNetworkType	Enum
InInterfaceTgrpContext	String	OutInterfaceIPType	Enum
InInterfaceTrustLevel	Enum	OutInterfacePort	Number
InInterfaceTimeZone	String	OutInterfaceProtocol	Enum
OutInterfaceId	Number	OutInterfaceDomain	String
OutInterfaceName	String	OutInterfaceSIPconnect	Boolean
OutInterfaceSIPconnectType	Enum	OutInterfaceSigTos	Number
OutInterfaceIMS	Boolean	OutInterfaceTgrpContext	String
OutInterfaceCIDBase	String	OutInterfaceTrustLevel	Enum
OutInterfaceNetwork	String	OutInterfaceTimeZone	String
OutInterfaceSubTraffic	Boolean	CurInterfaceId	Number
OutInterfaceAllowOnlyAssocPeers	Boolean	CurInterfaceName	String
OutInterfaceAccType	Enum	CurInterfaceType	Enum
OutInterfaceOperOd	String	CurInterfaceIP	String
CurInterfaceNetworkType	Enum	CurInterfaceCIDBase	String
CurInterfaceIPType	Enum	CurInterfaceNetwork	String
CurInterfacePort	Number	CurInterfaceSubTraffic	Boolean
CurInterfaceProtocol	Enum	CurInterfaceAllowOnlyAssocPeers	Boolean
CurInterfaceSIPconnect	Boolean	CurInterfaceAccType	Enum
CurInterfaceSIPconnectType	Enum	CurInterfaceOperOd	String
CurInterfaceDomain	String	CurInterfaceSigTos	Number
CurInterfaceIMS	Boolean	CurInterfaceTgrpContext	String
CurInterfaceTrustLevel	Enum	InPeerSubTraffic	Boolean
CurInterfaceTimeZone	String	InPeerOperId	String
InPeerId	Number	InPeerTrustLevel	Enum
InPeerName	String	InPeerHostType	Enum
InPeerClassId	String	InPeerHost	String
InPeerType	Enum	InPeerPort	Number
InPeerNetworkType	Enum	InPeerProtocol	Enum
InPeerIms	Boolean	InPeerTgrpId	String
InPeerTimeZone	String	OutPeerOperId	String
OutPeerId	Number	OutPeerTrustLevel	Enum
OutPeerName	String	OutPeerHostType	Enum
OutPeerClassId	String	OutPeerHost	String

Field	Type	Field	Type
OutPeerType	Enum	OutPeerPort	Number
OutPeerNetworkType	Enum	OutPeerProtocol	Enum
OutPeerIms	Boolean	OutPeerTgrpld	String
OutPeerSubTraffic	Boolean	OutPeerTimeZone	String
CurPeerId	Number	CurPeerTrustLevel	Enum
CurPeerName	String	CurPeerHostType	Enum
CurPeerClassId	String	CurPeerHost	String
CurPeerType	Enum	CurPeerPort	Number
CurPeerNetworkType	Enum	CurPeerProtocol	Enum
CurPeerIms	Boolean	CurPeerTgrpld	String
CurPeerSubTraffic	Boolean	CurPeerTimeZone	String
CurPeerOperId	String	InLegParamNetworkType	Enum
InLegParamIMS	Boolean	InLegMaxSE	Number
InLegParamSubTraffic	Boolean	InLegSessionTimer	Number
InLegT1Timer	Number	InLegReqRelRspinINV	Boolean
InLegT2Timer	Number	InLegInitiateRelRsp	Boolean
InLegTimerC	Number	InLegForceFastStart	Boolean
InLegMaxRetransmissions	Number	InLegTgrpFormat	Enum
InLegSupportedMethods	String	InLegInsertTgrpInfo	Boolean
InLegMinSE	Number	InLegMinMF	Number
OutLegParamNetworkType	Enum	OutLegMinSE	Number
OutLegParamIMS	Boolean	OutLegMaxSE	Number
OutLegParamSubTraffic	Boolean	OutLegSessionTimer	Number
OutLegT1Timer	Number	OutLegReqRelRspinINV	Boolean
OutLegT2Timer	Number	OutLegInitiateRelRsp	Boolean
OutLegTimerC	Number	OutLegForceFastStart	Boolean
OutLegMaxRetransmissions	Number	OutLegTgrpFormat	Enum
OutLegSupportedMethods	String	OutLegInsertTgrpInfo	Boolean
OutLegMinMF	Number	CurLegSupportedMethods	String
CurLegParamNetworkType	Enum	CurLegMinSE	Number
CurLegParamIMS	Boolean	CurLegMaxSE	Number
CurLegParamSubTraffic	Boolean	CurLegSessionTimer	Number
CurLegT1Timer	Number	CurLegReqRelRspinINV	Boolean
CurLegT2Timer	Number	CurLegInitiateRelRsp	Boolean
CurLegTimerC	Number	CurLegForceFastStart	Boolean
CurLegMaxRetransmissions	Number	CurLegTgrpFormat	Enum

Field	Type	Field	Type
CurLegInsertTgrpInfo	Boolean	InLegAudioCodecRef	String
CurLegMinMF	Number	InLegVideoCodecPref	String
InLegMediaNetworkType	Enum	InLegFaxCodecPref	String
InLegMediaIMS	Boolean	InLegServiceNetworkType	Enum
InLegMediaSubTraffic	Boolean	InLegServiceIMS	Boolean
InLegInterceptMedia	Boolean	InLegServiceSubTraffic	Boolean
InLegMediaTos	Number	InLegMsgRouting	Boolean
InLegMediaLatching	Enum	InLegMaxRouting	Number
InLegTgrpMapping	Boolean	InLegDscCheckBodyTransp	Boolean
InLegRedirectMode	Enum	InLegDscCheckMediaTransp	Boolean
InLegPrivacy	Boolean	InLegDscCheckFuncTransp	Boolean
InLegDscCheckTopTransp	Boolean	InLegStaticTopTransp	Boolean
InLegDscCheckDialogTransp	Boolean	InLegStaticDialogTransp	Boolean
InLegDscCheckIdentityTransp	Boolean	InLegStaticIdentityTransp	Boolean
InLegDscCheckAcctTransp	Boolean	InLegStaticAcctTransp	Boolean
InLegDscCheckHeaderTransp	Boolean	InLegStaticHeaderTransp	Boolean
InLegStaticBodyTransp	Boolean	InLegMaxRegInterval	Number
InLegStaticMediaTransp	Boolean	InLegRegExp	Number
InLegStaticFuncTransp	Boolean	InLegSubscrRegEv	Boolean
InLegMediaInactivityMonitor	Boolean	InLegSubscrPer	Number
InLegMediaInactMonPeriod	Number	InLegRelRegExp	Boolean
InLegStrictOfferAnswerMode	Boolean	InLegFeNatTravMode	Enum
InLegMaxAllowRef	Number	InLegFeNatTravInterval	Number
InLegMinRegInterval	Number	OutLegMediaNetworkType	Enum
OutLegMediaIMS	Boolean	OutLegServiceNetworkType	Enum
OutLegMediaSubTraffic	Boolean	OutLegServiceIMS	Boolean
OutLegInterceptMedia	Boolean	OutLegServiceSubTraffic	Boolean
OutLegMediaTos	Number	OutLegMsgRouting	Boolean
OutLegMediaLatching	Enum	OutLegMaxRouting	Number
OutLegAudioCodecRef	String	OutLegTgrpMapping	Boolean
OutLegVideoCodecPref	String	OutLegRedirectMode	Enum
OutLegFaxCodecPref	String	OutLegPrivacy	Boolean
OutLegDscCheckTopTransp	Boolean	OutLegStaticTopTransp	Boolean
OutLegDscCheckDialogTransp	Boolean	OutLegStaticDialogTransp	Boolean
OutLegDscCheckIdentityTransp	Boolean	OutLegStaticIdentityTransp	Boolean
OutLegDscCheckAcctTransp	Boolean	OutLegStaticAcctTransp	Boolean

Field	Type	Field	Type
OutLegDscCheckHeaderTransp	Boolean	OutLegStaticHeaderTransp	Boolean
OutLegDscCheckBodyTransp	Boolean	OutLegStaticBodyTransp	Boolean
OutLegDscCheckMediaTransp	Boolean	OutLegStaticMediaTransp	Boolean
OutLegDscCheckFuncTransp	Boolean	OutLegStaticFuncTransp	Boolean
OutLegMediaInactivityMonitor	Boolean	OutLegSubscrPer	Number
OutLegMediaInactMonPeriod	Number	OutLegRelRegExp	Boolean
OutLegStrictOfferAnswerMode	Boolean	OutLegFeNatTravMode	Enum
OutLegMaxAllowRef	Number	OutLegFeNatTravInterval	Number
OutLegMinRegInterval	Number	CurLegMediaNetworkType	Enum
OutLegMaxRegInterval	Number	CurLegMediaIMS	Boolean
OutLegRegExp	Number	CurLegMediaSubTraffic	Boolean
OutLegSubscrRegEv	Boolean	CurLegInterceptMedia	Boolean
CurLegMediaTos	Number	CurLegMsgRouting	Boolean
CurLegMediaLatching	Enum	CurLegMaxRouting	Number
CurLegAudioCodecRef	String	CurLegTgrpMapping	Boolean
CurLegVideoCodecPref	String	CurLegRedirectMode	Enum
CurLegFaxCodecPref	String	CurLegPrivacy	Boolean
CurLegServiceNetworkType	Enum	CurLegDscCheckTopTransp	Boolean
CurLegServiceIMS	Boolean	CurLegDscCheckDialogTransp	Boolean
CurLegServiceSubTraffic	Boolean	CurLegDscCheckIdentityTransp	Boolean
CurLegDscCheckAcctTransp	Boolean	CurLegStaticAcctTransp	Boolean
CurLegDscCheckHeaderTransp	Boolean	CurLegStaticHeaderTransp	Boolean
CurLegDscCheckBodyTransp	Boolean	CurLegStaticBodyTransp	Boolean
CurLegDscCheckMediaTransp	Boolean	CurLegStaticMediaTransp	Boolean
CurLegDscCheckFuncTransp	Boolean	CurLegStaticFuncTransp	Boolean
CurLegStaticTopTransp	Boolean	CurLegMediaInactivityMonitor	Boolean
CurLegStaticDialogTransp	Boolean	CurLegMediaInactMonPeriod	Number
CurLegStaticIdentityTransp	Boolean	CurLegStrictOfferAnswerMode	Boolean
CurLegMaxAllowRef	Number	CurLegFeNatTravInterval	Number
CurLegMinRegInterval	Number	Direction	Enum
CurLegMaxRegInterval	Number	Incoming	Boolean
CurLegRegExp	Number	Outgoing	Boolean
CurLegSubscrRegEv	Boolean	PEGenericParameter	String
CurLegSubscrPer	Number	BandwidthUsed	Number
CurLegRelRegExp	Boolean	CurLegFeNatTravMode	Enum

Table 20: IP Fields

Appendix F SDP Parameters

Line	Field	Type	Line	Field	Type
v	Version	String	s	SessionName	String
o	UserName	String	i	Information	String
	Id	String	u	URI	String
	Version	String	e	Address	String
	NetworkType	Enum		Text	String
	AddressType	Enum	p	PhoneNumber	String
	Address	String		Text	String
	FQDN	Boolean	b	BandwidthType	String
NetworkType	Enum	BandwidthValue		Number	
c	AddressType	Enum	z	ZoneAdjustment	Number
	Address	String		ZoneOffset	String
	TTL	Number	k	KeyMethod	Enum
	AddressCount	Number		KeyData	String
	FQDN	Boolean			
t	StartTime	Number			
	StopTime	Number			
	RepeatInterval	Number			
	RepeatDuration	Number			
	RepeatOffset	Number			

Table 21: SDP Session Lines and Fields

**Note:**

The r-line is incorporated into the t-line. There is no direct reference to an r-line.

Attribute	Field	Type	Field	Type
cat	Category	String		
charset	CharacterSet	String		
crypto	Tag	Number	KeyInfo	String
	Suite	String	SessionParameter	String
	KeyMethod	String		
fingerprint	Fingerprint	String		
group	Semantics	Enum	IdTag	String
key_mgmt	Protocol	String	Data	String
keywds	Keyword	String		
lang	Language	String		
setup	Role	String		

Attribute	Field	Type	Field	Type
tool	Value	String		
type	Value	String		
Other	Value	String		
inactive	These four attributes do not have fields.			
recvonly				
sendonly				
sendrecv				

Table 22: SDP Session Attributes and Fields

Line	Field	Type	Field	Type
m	Port	Number	Format	String
	PortCount	Number	Payload	Number
	Protocol	Enum	Codec	String
i	Information	String		
c	NetworkType	Enum	TTL	Number
	AddressType	Enum	AddressCount	Number
	Address	String	FQDN	Boolean
b	BandwidthType	String	BandwidthValue	Number
k	KeyMethod	Enum	KeyData	String

Table 23: SDP Media Lines and Fields

Attribute	Field	Type	Field	Type
accept_types	MediaType	String		
accept_wrapped_types	MediaType	String		
charset	CharacterSet	String		
conf	PreconditionType	Enum	PreconditionDirection	Enum
	PreconditionStatus	Enum		
confid	Conferenceld	String		
connection	ConnectionValue	String		
crypto	Tag	Number	KeyInfo	String
	Suite	String	SessionParameter	String
	KeyMethod	String		
curr	PreconditionType	Enum	PreconditionDirection	Enum
	PreconditionStatus	Enum		
des	PreconditionType	Enum	PreconditionStatus	Enum
	PreconditionStrength	Enum	PreconditionDirection	Enum

Attribute	Field	Type	Field	Type
fingerprint	Fingerprint	String		
floorctrl	FloorRole	String		
floorid	FloorId	String		
fntp	Format	Number	FormatParameter	String
framerate	FrameRate	String		
key_mgmt	Protocol	String	Data	String
lang	Language	String		
max_size	MaxMessageSize	String		
maxptime	MaxPacketTime	String		
mid	IdTag	String		
orient	Orientation	String		
path	PathURI	String		
ptime	PacketTime	String		
quality	Quality	String		
rtcp	Port	Number	Address	Number
	NetworkType	Enum	FQDN	Boolean
	AddressType	Enum		
rtpmap	Format	Number	EncodingName	String
	Payload	Number	ClockRate	Number
	Channel	Number	EncodingParameter	String
setup	Role	String		
userid	UserId	String		
Other	Value	String		

Table 24: SDP Media Attribute and Fields

[1] Useful in remote SP-Network cases when BN has to send via unsecure network

[2] For the sake of accommodating large messages