GENESYS
AN ALCATEL-LUCENT COMPANY

VoiceGenie 7.2.2

SIP Proxy

User's Guide

About Genesys

Genesys Telecommunications Laboratories, Inc., a subsidiary of Alcatel-Lucent, is 100% focused on software for call centers. Genesys recognizes that better interactions drive better business and build company reputations. Customer service solutions from Genesys deliver on this promise for Global 2000 enterprises, government organizations, and telecommunications service providers across 80 countries, directing more than 100 million customer interactions every day. Sophisticated routing and reporting across voice, e-mail, and Web channels ensure that customers are quickly connected to the best available resource—the first time. Genesys offers solutions for customer service, help desks, order desks, collections, outbound telesales and service, and workforce management. Visit www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Genesys, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 or +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-127-645-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp

Prior to contacting technical support, please refer to the Genesys Technical Support Guide for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the Genesys 7 Licensing Guide.

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com
Document Version: 10-2009

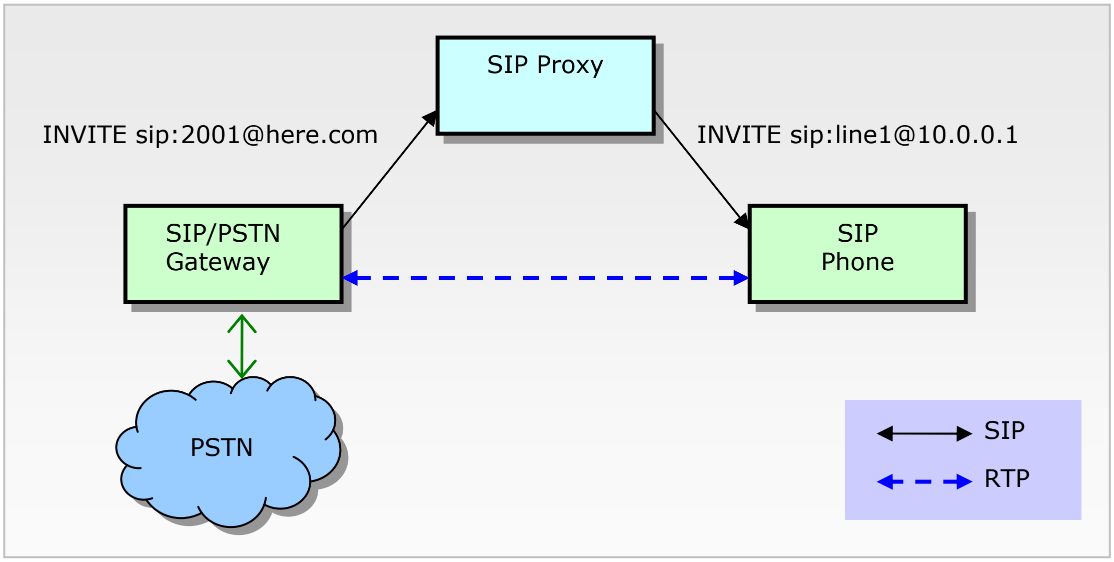
Table of Contents

Chapter

1 Introduction

Article I. 1.1 Overview

SIP proxies provide a variety of services in VoIP networks that are based on the use of SIP, such as authorization and access control, validation and security, call routing, accounting, user location, and others. A variety of SIP proxies are available, some of which are commercial (e.g. Cisco) and some of which are open source (e.g. Vovida, iptel.org). Fundamentally, any SIP proxy delivers services by controlling how requests and responses are routed between a SIP client (UAC) and a SIP server (UAS), generally by deciding on how requests get routed, as well as by manipulating headers in the request or response. A simple example of this is shown in the figure below:



The VoiceGenie SIP proxy provides a very specific set of services that are not available from general-purpose SIP proxies that are widely available. These services are specific to providing media-centric SIP services, such as

VoiceXML dialogs, and conferencing capabilities. The VoiceGenie SIP Proxy is also designed for scalability and has redundant architecture to protect against server failures.

1.2 Features and Capabilities

The VoiceGenie SIP proxy focuses on the following capabilities:

- **SIP Service mapping** capabilities, where incoming calls are mapped to particular services with support for transformations on service parameters. For instance, an `INVITE` request processed by the proxy with the Request-URI set to `sip:1234@test.com` could be remapped to `sip:ccxml@box1.server.com;page=http://webserver.com/a/test.ccxml`.
- **Active monitoring** of SIP resources, ensuring that requests for the service will be routed around any unavailable SIP resources.
- **Load balancing** of requests/calls to a particular SIP service across the SIP resources that provide that service, such as distributing a set of incoming calls to a collection of VoiceGenie media platforms.
- **High availability,** in which multiple clustered VoiceGenie SIP proxies handle the failure of individual proxies within the cluster, maintaining availability of SIP services managed by the cluster.
- **Conference management,** in which conferences are allocated to available conferencing resources, with all calls to access a conference being routed to the appropriate resource. This alleviates conference users from having to be aware of which conferences exist on what resources.

High availability is very closely related to the way that the proxy is deployed and is discussed below in Deployment Architecture. The other capabilities listed above are described in greater detail in the following sections.

1.3 Terminology

The follow table list essential terminologies used throughout this document.

Terminology	Description
SIP Service	A proxy manages access to SIP Services via SIP requests, which are delivered using one or more SIP Resources managed by the proxy. Examples: • VoiceXML Attendant Service • Conference Server • CCXML Interpreter • Media Server • SIP/PSTN gateway

SIP Resource	An instance of server that provides one or more SIP Services. A SIP Resource is addressed by a specific hostname and port of the server that accepts SIP requests. It may have a number of other properties such as an assigned SIP resource type.
	Example:
	`MediaPlatform1 {`
	`Host = mediaplatform.voicegenie.com:5060`
	`ip address = 10.0.0.123SIP Resource Type = VG-MediaPlatform-7 }`

SIP Resource Type	A collection of properties that describe specific types of resources, and provide the information necessary for the SIP proxy to use that resource. For instance, a `VG-MediaPlatfrom-7` resource type could be defined as one that supports 300 ports of VoiceXML or Conferencing sessions. All SIP resources have a single SIP Resource Types. Example Properties for defining a SIP Resource Type: • Registration Method • Monitoring Method • Capability • Number of ports • Maximum conference size
Capability	Defines the type of capability that a SIP Resource can provide. A SIP Resource can provide one or more capabilities. Examples: • Conferencing • VoiceXML • CCXML

Terminology	Description
Registration Method	Each SIP Resource must be registered with the SIP Proxy in order to properly forward requests to the resource. A resource can register to the SIP via SIP `REGISTER` method, or if the component is part of the VoiceGenie OA&M framework, the component can register through the CMP interface. Examples: • SIP REGISTER • CMP • no registration methods (assumed registered)
Monitoring Method	SIP Proxy is required to monitor the availability of the resources in order to forward requests to active resources. SIP Proxy can periodically send SIP `OPTIONS` message to a resource to check the livelihood of the server, or use the VoiceGenie OA&M framework to monitor the resource. SIP Proxy only forwards to resources that are active. Examples: • SIP OPTIONS • CMP • no monitoring methods (assumed active)
SIP Transport	SIP Proxy uses different types of transport to receive and deliver the SIP messages. Each transport protocol has different properties and provides different advantages. Examples: • UDP (unicast) • UDP (multicast) • TCP • SCTP (not supported in this release)

Chapter

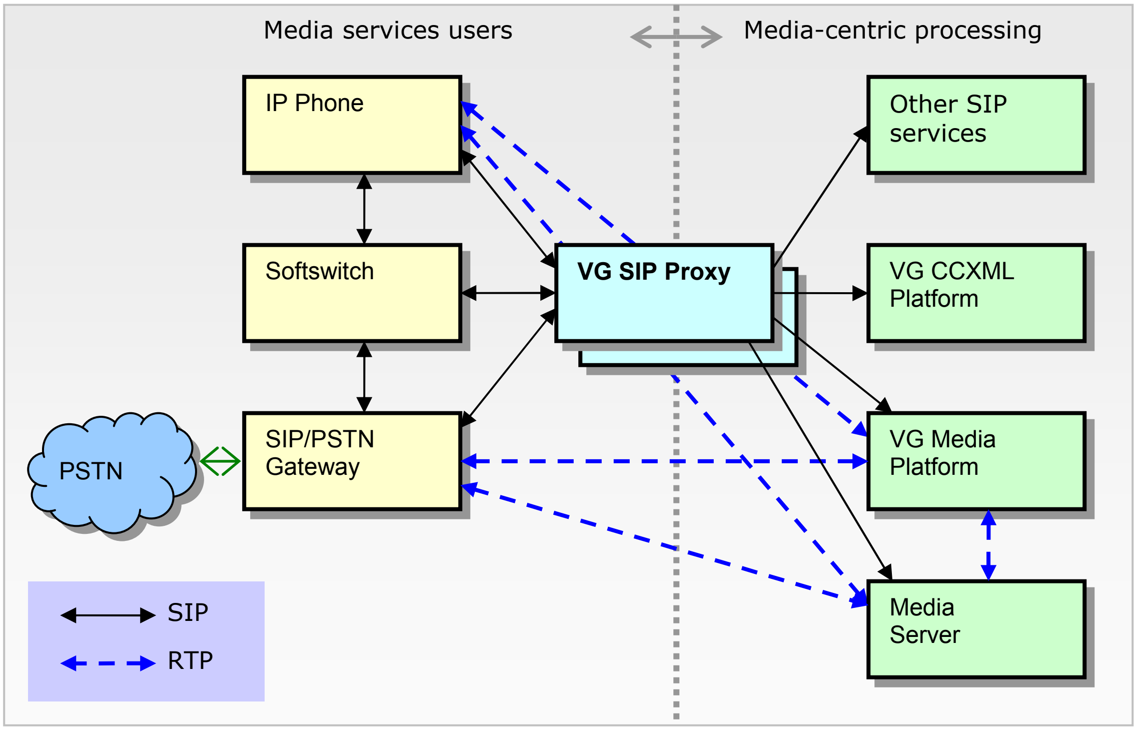
2 Deployment Architecture

There are a number of options with respect to how the VoiceGenie SIP proxy can be deployed. These options are explained by the following sections:

- The 2.1 Logical Architecture of a deployment using the VoiceGenie SIP proxy is first discussed, showing how the proxy interacts with other elements in the network;
- The 2.2 Network Configurations section shows more concretely how the VoiceGenie SIP proxy is connected to the network or networks that exist in a typical deployment.

2.1 Logical Architecture

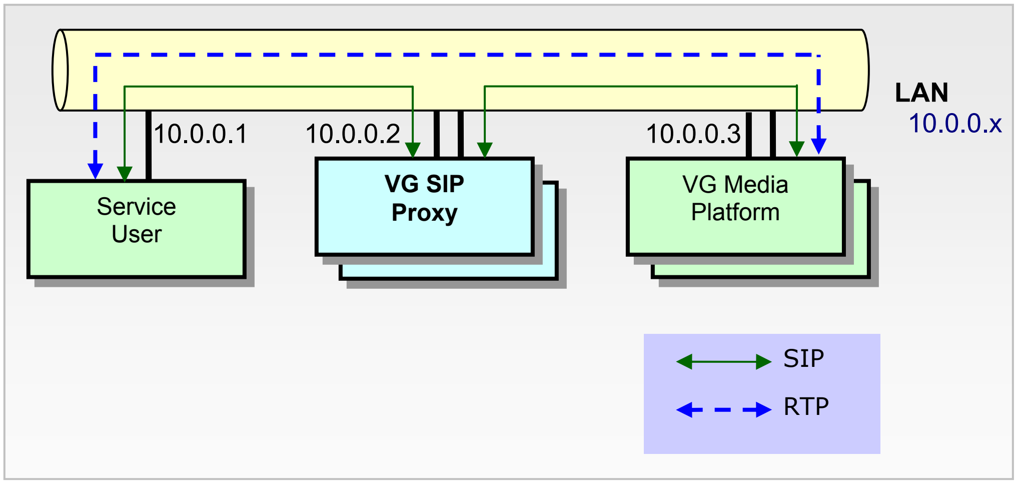
From a logical perspective, the purpose of a VoiceGenie SIP proxy is to act as an interface to a collection of media processing resources, such as the VoiceGenie Media Platform, CCXML platform, audio conferencing or other resources. SIP devices and applications can then make use of media-centric services through the proxy, without having to know the actual location of those resources or how to manage various routing decisions. The services provided by the VoiceGenie SIP proxy are used not only by clients such as media gateways or softswitches, but may also be used by internal media resources to co-ordinate interactions with one another. For instance, the VoiceGenie CCXML platform offers the ability to manage a VoiceXML dialog that actually executes on a VoiceGenie media platform; the CCXML platform may make use of proxy capabilities to locate an appropriate VoiceXML platform. The following diagram illustrates this logical architecture:



Although the above diagram shows a number of elements, both users of media services managed by the proxy as well as the underlying media-centric services that the proxy routes to, the actual configuration used in a deployment is typically much simpler. For instance, in a contact centre environment providing automated self-service or call routing, the deployment might consist solely of a number of SIP/PSTN gateways to handle incoming calls, a cluster of VoiceGenie media platforms that provide the actual treatment of calls through touch-tone or speech applications, and a redundant pair of VoiceGenie SIP proxies that provide load balancing across the available VoiceGenie platforms. In more complex next-generation network architectures, the VoiceGenie SIP proxy essentially acts as a *logical media server,* aggregating the capabilities of heterogeneous array of processing resources and acting as a single interface point for media services users.

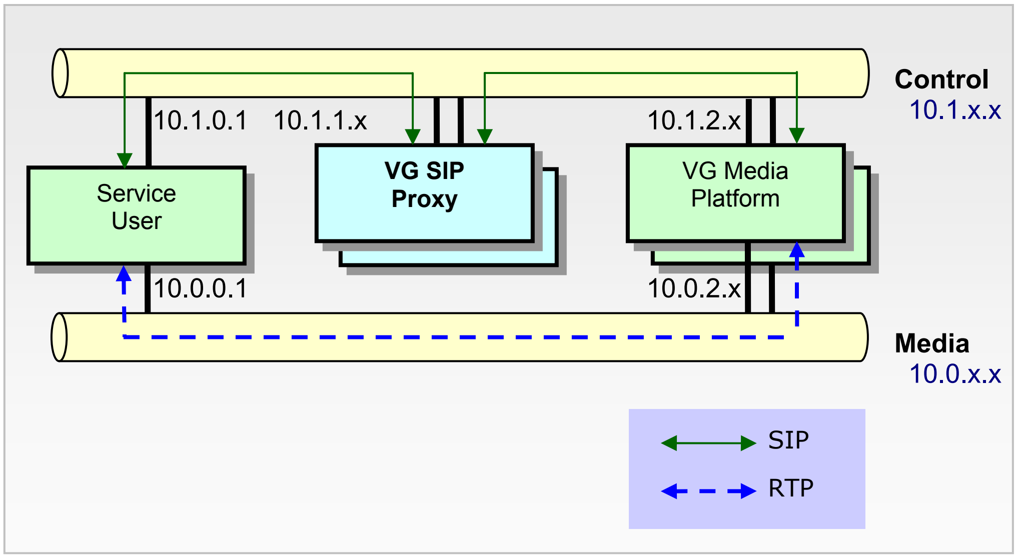
2.2 Network Configurations

The VoiceGenie SIP proxy can be deployed in two basic network configurations, which vary with respect to the degree of isolation between SIP control and RTP media, and in the separation of "external" SIP interfaces – which face the users of the SIP proxy – from "internal" SIP interfaces – which face the SIP resources being managed. The simplest of these is the *integrated network* configuration, in which all SIP and RTP traffic between elements make use of a single physical network:



The single integrated network shown is a *logical* network; the physical realization of this may employ the use of redundant network connections with Ethernet bonding, or other similar approaches to ensure that there is no single physical point of failure.

In some environments, it is desirable to separate control and media, typically to ensure priority and quality of service of the control channel, but also sometimes to ensure the privacy of data exchanged on the control channel. This *separated network* configuration is shown below:



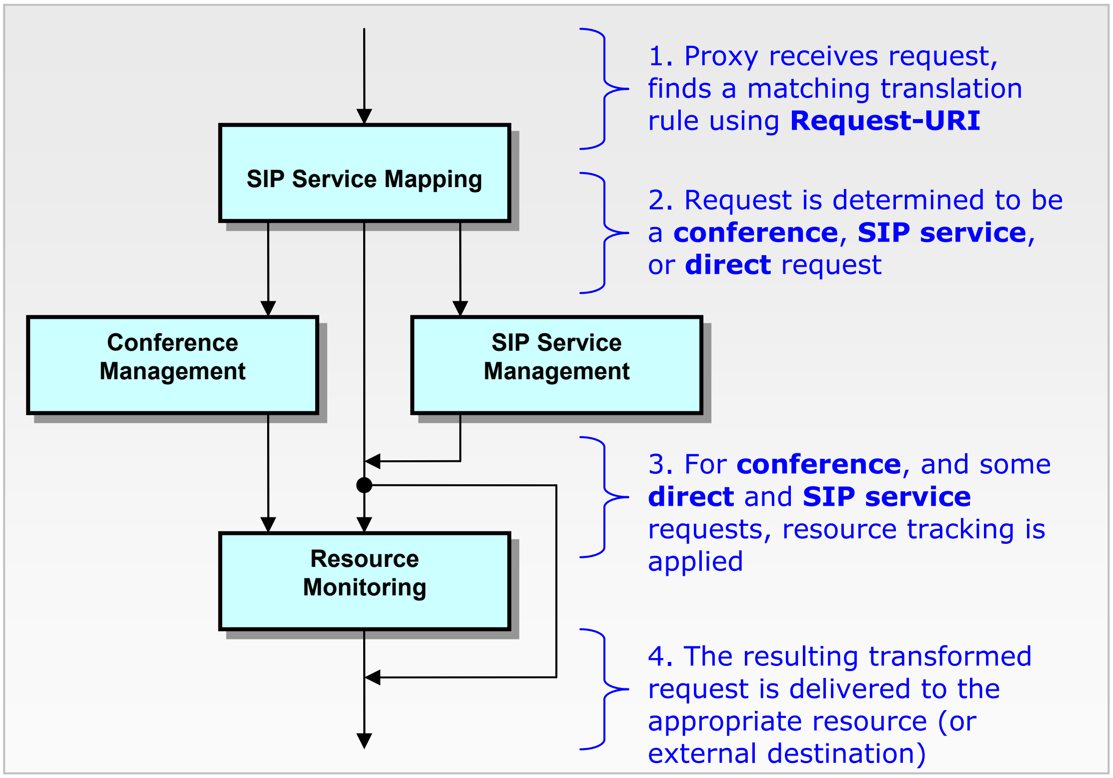
As mentioned above, each of the control/media networks may themselves involve multiple redundant physical connections. However, it is also possible that the two networks (control and media) may be logically separated, but might share the same physical network infrastructure. As an example, a single physical switch might provide routing for two separate VLANs, with the service user and VoiceGenie media platform each being configured to support two IP addresses (each on a different VLAN) on a single physical interface.

Chapter

3 Features

3.1 Request Processing

The first step involved when a VoiceGenie SIP proxy handles a request for service (which currently is for `INVITE` requests only) is to perform service translation and matching to transform the incoming request and match that request to a service offered by the proxy. In addition to the translation services that the proxy offers, it may also provide SIP service, conference, and resource management. The following diagram illustrates the process of handling a request:


SIP Service Mapping
Conference Management
SIP Service Management
Resource Monitoring
1. Proxy receives request, finds a matching translation rule using **Request-URI**
2. Request is determined to be a **conference**, **SIP service**, or **direct** request
3. For **conference**, and some **direct** and **SIP service** requests, resource tracking is applied
4. The resulting transformed request is delivered to the appropriate resource (or external destination)

- *SIP Service Mapping* provides rule-based matching of a received request, and allows for general-purpose transformations of the Request-URI and any parameters of the Request-URI. The example given above was the use of this service to convert a URI such as `sip:dialog.vxml.http%3A//a/test.vxml@test.com` (from the Rosenberg IETF draft) to `sip:service1234@box1.server.com;voicexml=http://a/test.vxml` (consistent with RFC 4240). The rules, which are supplied by the proxy user, make use of regular expression matching and manipulation to allow a combination of input parameters to be mapped to a combination of output parameters. In addition, the rules allow control over whether the request will be further handled by conference service management, the more general SIP service management, or whether the request is directly routed to a particular target URI.
- *Conference Management* is described separately below, and thus will not be covered here.
- *SIP Service Management* allows for requests to be routed to higher level SIP services. As part of defining these SIP services, the administrator defines the resources that can be used to fulfill these services. This allows for the identification of a collection of resources that could potentially fulfill a given request. The capabilities described in the sections below on resource monitoring and load balancing are then used to filter and

prioritize the resources that are actually used as the destination for the request.

- *SIP Resource Monitoring* is an optional service that is typically applied for resources managed by the proxy, but skipped for external resources such as media gateways which may be shared by other users. Resource tracking allows the proxy to keep track of the level of utilization of a particular resource, based on the requests that are active against that resource. Far from a simple tracking table, the resource tracking module provides synchronization across a cluster of proxies, so that each proxy is capable of making a globally optimal decision with respect to resource routing.

The above steps in the request handling process do not necessarily result in the selection of a single resource or destination for the request; rather, a prioritized list is generated of resources that could handle the needs of a given request. This is described in greater detail in the load balancing section below.

3.2 SIP Resource Monitoring

In order to make optimal decisions, the VoiceGenie SIP proxy needs to have up-to-date information concerning the status of resources under the management of the proxy. The proxy uses a variety of mechanisms to monitor resources:

- **SIP Registration.** The proxy allows resources to use it as a SIP registrar – however, this is not required. Resources that require registration but do not hold an active registration are considered to be unavailable.
- **Periodic OPTIONS Requests.** The proxy can monitor SIP resources by sending a periodic `OPTIONS` request to validate that the SIP resource is responding correctly. This allows the proxy to be aware within a relatively short period (configurable with a default of two minutes) if a resource fails.
- **OA&M Integration.** When the proxy is used with VoiceGenie products that are integrated into the VoiceGenie common OA&M infrastructure, the proxy gets real-time status information about the resource, supported by a frequent heartbeat between the OA&M infrastructure and the resource.

When the proxy determines the set of resources that are capable of servicing a particular request, it will restrict that set to the resources that are currently detected as being available. This ensures that requests are only routed to resources that are available.

In addition to using monitoring information to make optimal routing decisions, the VoiceGenie SIP proxy also feeds this information into VoiceGenie's underlying OA&M infrastructure, allowing the System Management Console (SMC) to display status information about the resources in the cluster as perceived by the VoiceGenie SIP proxy. This provides the operator with a single view over all resources being used in conjunction with the proxy.

Note: If the OA&M Integration method is used to monitor the status of the SIP resources, and there is only a primary CMP Server for a cluster environment of SIP Proxies, the CMP Server is a potential single point of failure. If the CMP Server goes offline, the status of SIP resources that were monitored by the CMP Server will be unavailable to the SIP Proxy. The SIP Proxy will therefore will not route calls to these SIP resources whose status is unavailable. The SIP Proxy will still be able to monitor the status of SIP resources which are not monitored by the CMP Server. To prevent this single point of failure, the SIP Proxy can be configured to use SIP OPTIONS to monitor the SIP resources instead of the OA&M Integration method in the case of a cluster environment with only a primary CMP Server with no backup CMP Server.

3.3 Load Balancing

Once the VoiceGenie SIP proxy has determined the resources that are capable of servicing a particular request, and validates that those resources are available (based on actively monitoring resources as described above), it load balances requests across those resources. Two kinds of load balancing are supported by the proxy:

- *Round Robin* load balancing, which rotates requests amongst available resources without using additional information about the resource;
- *Least Utilized* routing, in which the proxy makes use of information about the services currently being provided using each resource in order to select the resource that has the most available capacity.

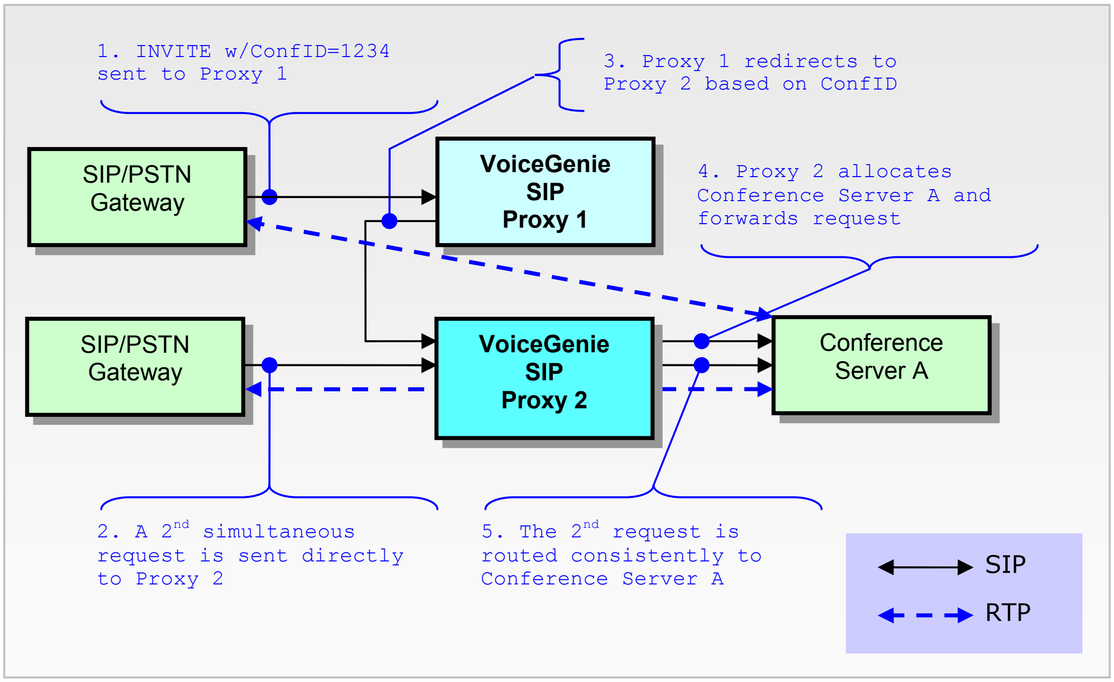
The mechanism used for load balancing serves as a means for prioritizing across available resources. Thus, if the resource that the proxy believes is the least utilized turns out not to be available to service the request, the proxy will not fail the request and return an error – it will try the next least utilized resource in the set it identified was capable of servicing the request. From a more technical perspective, the VoiceGenie SIP proxy thus acts as a sequentially forking proxy.

The proxy also supports customization of the timeouts used when sending requests, so that it is possible to failover to an available second resource quickly in the event that the initial resource selections are not available.

3.4 Conference Management

A key capability of the proxy that is relative orthogonal to other capabilities is the management of conferencing/mixing resources. Whereas general SIP services generally have the property that independent instances (separate calls) do not interact with one another, conferences have the property that multiple callers to the same conference need to be routed to the same physical resource for mixing. In addition, even if a request is received simultaneously on two or more proxy instances in a clustered environment, it is necessary that the decision be consistent across the entire cluster. That is to say, if proxy # 1 routed conference 1234 to conferencing resource *X,* then proxy # 3 must also route any request for conference 1234 to conferencing resource *X.*

The VoiceGenie SIP proxy provides consistent *conference routing* through two internal capabilities. The first is *transparent request redirection,* where one specific available proxy is always guaranteed to handle requests to a particular conference. Thus, if proxy # 2 is designated as the handler for conference 1234, then all requests related to conference 1234 will be handled by proxy # 2. Through supporting low-level message redirection or UDP multicast (described below), the proxy is able to achieve this very efficiently. UDP multicast in particular adds very little processing overhead, since an extra network transmission is not necessary – the proxy chosen to handle the incoming SIP request is always the right one. The second capability is *conference state mirroring,* in which conferences that are set up successfully are broadcasted to all proxies in the cluster. In this way, if one particular proxy in the cluster was to fail, the surviving members of the cluster could route conference requests in a manner that was consistent with the failed proxy. This is shown in the diagram below:



In addition to managing the routing of conferences, the proxy also manages *conference resource allocation.* Conference resources are typically rated to handle a particular number of simultaneous conference participants; they are also typically further subject to limitations on the maximum size of any individual conference being hosted. The VoiceGenie SIP proxy makes use of information about the currently available resource level, coupled with minimum and maximum conference size information that is derived from the conference request, to allocate conferences to the optimal resource with the highest chance of being able to serve the request successfully. The ability to specify a minimum conference size allows the proxy to reserve a certain amount of capacity on a conference resource, such that even if other calls arrive in the interim, it is guaranteed that sufficient capacity will be available to deliver the required minimum number of participants on existing conferences using that resource.

Conference management requires the following 3 request-URI parameters for all SIP requests to conferences:

1. `confinstid` – a cluster-wide unique identifier that identifies the conference.
2. `confmaxsize` – the maximum size of the conference. If there are more participants joining the conference than the max size, SIP Proxy will reject the callers above the maximum conference size.
3. `confreserve` – the number of ports reserved for this conference resource when the first caller starts a conference. Port count for the conference resource will be incremented by the amount specified by this parameter when the first caller starts a conference.

Note that if any of the three parameters are specified, all three parameters have to be provided by the caller. `confmaxsize` parameter has to be always greater than or equal to `confreserve` parameter. SIP Service Mapping can use translation rules to provide the URI parameters. Here is an example:

Caller sends:

```
sip:conf=1234@10.0.0.1;conftype=threewaycall
```

SIP Service Mapping translates the URI to:

```
sip:conf=1234@10.0.0.1;sipservice=conf;confinstid=1234;
confmaxsize=3;confreserve=3
```

The represents a 3-way call where the caller reserves 3 ports on the conference resource. SIP Proxy only allows a maximum of 3 concurrent participants to this conference. The 4th caller will be rejected by the SIP Proxy.

Chapter

4 SIP Proxy Redundancy

SIP Proxy redundancy can be achieved by deploying 2 SIP Proxies. SIP Proxy redundancy is supported on two deployment configurations:

1. Multicast setup
2. Unicast setup

For both setups, SIP dialog failover is supported but not on a transaction level. If SIP proxy fails during a transaction, the affected call may fail.

4.1 Multicast Setup

In multicast SIP Proxy redundancy setup, both proxies receive incoming messages from the same multicast address. The proxies internally exchange information to decide which proxy to handle the call. In the case of a SIP Proxy failure, the remaining one will detect the problem and start processing all SIP requests that were suppose to be handled by the failed proxy. Under multicast setup, all external device should send incoming calls to the multicast address. Please refer to 5.2.4 Multicast setting for setup instructions.

4.2 Unicast Setup

In unicast SIP Proxy redundancy setup, a DNS server is required. External devices should send incoming call to a proxy by first lookup the name for the proxies. Please refer to 5.2.5 Unicast setting for setup instructions.

4.3 Registrar

SIP Proxy redundancy setup is used for both unicast and multicast setup. Multicast setup should be used for Registrar transport when deploying with 2 proxies. In this case, a single REGISTER message will be received and processed by both proxies. For unicast setup, `REGISTER` message must be sent to each individual proxy.

Chapter

5 Installation

Like all other VoiceGenie 7.2 components, the SIP Proxy should be installed/deployed/configured using VoiceGenie OA&M framework, which must be installed before any other VoiceGenie software can be deployed.

This section covers the installation process for the SIP Proxy module, assuming that the operating system, all third party software and the VoiceGenie OA&M Framework software are installed; if they are not installed please refer to the document: *VoiceGenie 7.2 Installation Guide*

Required files:

Component	Filename
SIP Proxy	ccp-proxy-7.2.0-x.tar.gz
CCP Redundancy Manager	ccp-rm-7.2.0-x.tar.gz
Linux Bonding Driver (Linux only)	bonding-7.0.0-3.tar.gz

5.1 Installation Steps

1. Product Manager

   Open a web browser and go to the System Management Console homepage;

   Login and click on `Configuration` tab;

   On the left hand column click on `Product Manager;`

   Click `Browse` and select one of the filenames listed above and click `OK;`

   Click `Upload` to upload the component to Product Manager;

   Repeat until all files are uploaded.

2. Config Profile Manager

   In System Management Console, go to `Configuration` tab;

On the left hand column click on `Config Profile Manager`;

At the `Product` drop-down list, select `SIP Proxy - 7.0.0`;

Enter a name for this profile;

Click `Create`, and you will notice the profile name will show up at the table below;

Repeat the above steps for the Call Control Platform Redundancy Manager product.

3. Deployment Manager

   In System Management Console, go to `Configuration` tab;

   On the left hand column click on `Deployment Manager`;

   Click on the radio button that corresponds to `SIP Proxy 7.0.0`;

   Select the configuration profile from the drop-down list that corresponds to this row;

   Select the servers where SIP Proxy will be deployed;

   Click `Deploy` button to start deploying;

   Check Deployment History to make sure the product is deployed successfully;

   Repeat the above steps for the Call Control Platform Redundancy Manager product.

5.2 Configuring the SIP Proxy

5.2.1 General

Before starting SIP Proxy, there are 6 configuration tables that need to be completed in order for SIP Proxy to properly forward incoming SIP requests:

1. SIP Service Table
2. SIP Services Mapping
3. SIP Resource Types Table
4. SIP Resources Table
5. SIP Proxy Configuration
6. Redundancy Manager Cluster Mapping

Please refer to Provisioning the SIP Proxy for instructions on provisioning the first 4 tables.

Please refer to the *VoiceGenie 7.2 SIP Proxy System Reference Guide* for the full lists of configuration parameters for SIP Proxy Configuration and Redundancy Manager Cluster Mapping.

SIP Proxy requires a license for operation. Please refer to the *VoiceGenie 7.2 Installation Guide* for obtaining and installing the license.

Note about enabling TCP Support:

The `sip.transport.X` parameters configures the SIP stack's transport settings. By default `sip.transport.0` has the value `transport0 udp:any:5060`.

To enable TCP transport:

1. Enable `sip.transport.1` and assign its value as `transport1 tcp:any:5060`.
2. Enable `sip.route.default.udp` and assign its value as `0`.
3. Enable `sip.route.default.tcp` and assign its value as `1`.

Note about Call Manager (under Media Platform) configuration:

Following changes are required for Call Manager in order to enable conferencing capability and interoperate with the VoiceGenie CCXML Platform:

Required for Conferencing capability

- `sessmgr.modules` = `VXML Remdial Conference`

  (hold down the `Control` key, and click each of the three values in the menu, to select them together)

- `sessmgr.appmodules` = `VXML:VXML Remdial:RemoteDial Conference:Conference`

  (hold down the `Control` key, and click each of the three values in the menu, to select them together)

- `sessmgr.conference.conference` = `Conference`

  (type `Conference` in text box, and click `Add` button)

Required for Dialog Server functionality

- `sip.transfermethods` = `REFER`

  (select `REFER` from menu)

- `sip.defaultblindxfer` = `REFER`

  (select `REFER` from drop-down menu)

- `sip.referxferhold` = `0`

  (please enable this parameter by clicking the checkbox beside the parameter, and select `0` from drop-down menu)

5.2.2 All-in-one Environment

After deploying the SIP Proxy, CCXML Platform, and Media Platform, please modify the following configuration parameters through SMC, to set up an all-in-one installation. Changes documented in this section apply only if Media Platform or CCXML Platform is installed with proper license.

In SMC, go to the `Configuration` tab. For each component below, click on the name in the left-hand menu, select the radio button for the configuration name, and click `Edit`, to edit the parameters.

SIP Proxy (listed under CCXML Platform)

1. `proxy.sip.transport.0` – change to `transport0 udp:any:5060`
2. `proxy.sip.proxy.respaddr` – add `<IP address of this machine>`

CCXML Interpreter (listed under CCXML Platform)

1. `sip.transport.0` – change to `transport0 udp:any:5068`
2. `ccpccxml.sip.proxy` – change to `<IP address of this machine>:<SIP Proxy port>`

Note: Note about `ccpccxml.sip.proxy` parameter:

CCXML Platform can refer to SIP Proxy with loop-back address only if `127.0.0.1` is listed as one of the responsible domains in SIP proxy's `proxy.sip.proxy.respaddr` configuration parameter.

Call Manager (listed under Media Platform)

`sip.transport.0` – enable and change to `transport0 udp:any:5066`

5.2.3 2 network IPs with bonding driver setting

The following are the instructions to configure 2 network IPs with bonding driver, and necessary SIP Proxy configuration changes to make this architecture work.

5.2.3.1 Network configuration changes

Bonding configuration files are located under `/etc/sysconfig/network-scripts`.

To configure 2 bondings, there should be two files such as `ifcfg-bond0`, `ifcfg-bond1`.

The first file `ifcfg-bond0`:

```
DEVICE=bond0
BOOTPROTO=none
IPADDR="10.0.0.174"
NETMASK="255.255.255.0"
BROADCAST="10.0.0.255"
ONBOOT="yes"
USERCTL=no
```

The second file `ifcfg-bond1:`

```
DEVICE=bond1
BOOTPROTO=none
IPADDR="10.0.1.174"
NETMASK="255.255.255.0"
BROADCAST="10.0.1.255"
ONBOOT="yes"
USERCTL=no
```

Network configuration files:

This totally depends on how many cards are installed. The example here is four. The file names are `ifcfg-eth[0-3]`. If `eth0` and `eth1` are bonding to `bond0`, the file content should be:

```
DEVICE=eth0
USERCTL=no
MASTER=bond0
SLAVE=yes
BOOTPROTO=none
ONBOOT="yes"
```

Device name will be different for `eth1`. If different bonding driver the NICs are bonding to the `MASTER` will be different:

```
DEVICE=eth2
. . .
MASTER=bond1
. . .
```

Change `/etc/modules.conf:`

Below is an example:

```
alias eth0 tg3
alias eth1 tg3
alias eth2 e1000
alias eth3 e1000
alias scsi_hostadapter mptbase
alias scsi_hostadapter1 mptscsih
alias usb-controller usb-ohci
alias bond0 bonding
options bond0 -o bond0 miimon=100 mode=1 multicast=1
alias bond1 bonding
options bond1 -o bond1 miimon=100 mode=1 multicast=1
```

5.2.3.2 CCP Proxy configuration change.

Second IP should be set in the `respaddr` parameter:

```
proxy.sip.proxy.respaddr = 10.0.0.174,10.0.1.174
```

1 more transport is added for second IP and 2 more are added if TCP is configured as well:

```
proxy.sip.transport.0 = transport0 udp:10.0.0.174:5061
proxy.sip.transport.1 = transport1 udp:10.0.1.174:5061
proxy.sip.transport.2 = transport2 tcp:10.0.0.174:5062
proxy.sip.transport.3 = transport3 tcp:10.0.1.174:5062
```

Related route destinations also need to be added:

```
proxy.sip.route.dests = 0 1 2 3
proxy.sip.route.dest.0 = 10.0.0.0 255.255.255.0 0 0
proxy.sip.route.dest.1 = 10.0.1.0 255.255.255.0 1 0
proxy.sip.route.dest.2 = 10.0.0.0 255.255.255.0 2 0
proxy.sip.route.dest.3 = 10.0.1.0 255.255.255.0 3 0
```

5.2.4 Multicast setting

The following SIP Proxy configurations must be changed for multicasting:

Multicast IP must be added in the `respaddr` parameter:

```
proxy.sip.proxy.respaddr = 10.0.0.99,10.0.0.137,225.0.0.10
```

`uactype` must be set to `multicast`:

```
proxy.sip.proxy.uactype = multicast
```

Multicast address and its device must be appended to the existing transport configuration:

```
proxy.sip.transport0 = transport0 udp:any:5060 mcast=225.0.0.10
mcast-if=eth0
```

Or if device is bonding to `bond0`:

```
proxy.sip.transport0 = transport0 udp:any:5060 mcast=225.0.0.10
mcast-if=bond0
```

The same changes apply to `registrar.sip.transport0` parameter except that `5062` is used instead of `5060` for port number.

Devices accessing SIP Proxy should point to `225.0.0.10`.

Warning! All the above mentioned configuration parameter values must be identical for the SIP Proxies within the same cluster.

5.2.5 Unicast setting

The following SIP Proxy configurations must be changed for unicasting:

Unicast IP must be added to the `respaddr` parameter. In the case of deploying 2 SIP Proxies for redundancy, both SIP Proxy IPs must be added:

```
proxy.sip.proxy.respaddr = 10.0.0.99,10.0.0.137
```

`uactype` must be set to `unicast`:

```
proxy.sip.proxy.uactype = unicast
```

Unicast transport should be configured:

```
proxy.sip.transport0 = transport0 udp:any:5060
```

When deploying 2 sip proxies with unicast setup for redundancy, the DNS server must be specified:

```
proxy.sip.proxy.dnsname = vgproxy.mynetwork.com:5061
```

Please note that the port is optional.

Warning! All the above mentioned configuration parameter values must be identical for the SIP Proxies within the same cluster.

5.2.6 SIP Registrar Setting

The following SIP Proxy configurations control the SIP registrar behavior:

A registrar transport must be specified. In a cluster of 2 SIP Proxies deployment, the user has a choice to setup this transport as multicast. In this case, both proxies will receive the same `REGISTER` message and both will reply with an identical response. If unicast is used, the `REGISTER` message must be send to the unicast IP of both SIP Proxies.

```
registrar.sip.transport.0 = transport0 udp:any:5060
```

Specify a list of acceptable domains for this registrar:

```
registrar.sip.registrar.domain = 10.0.0.190;
something.yournetwork.com
```

SIP Proxy will inspect the host portion of the Request-URI of the incoming `REGISTER` message. It will be accepted if it matches one of the domain in the list. If the list is empty, it will accept anything in the host portion.

Chapter

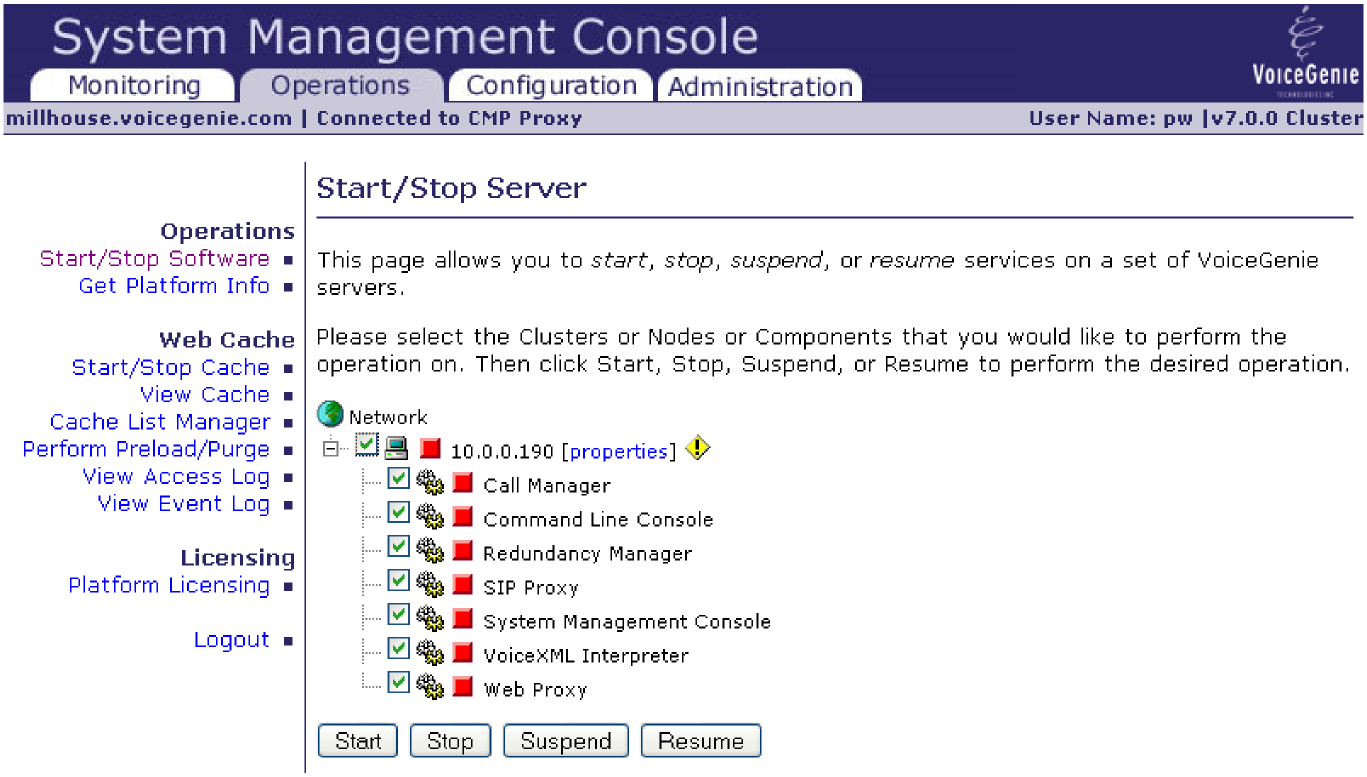
6 Operations, Administration, Maintenance

6.1 Starting and Stopping

Like all other VoiceGenie 7.2 components, the System Management Console offers a dedicated page to start or stop the SIP Proxy. Click on the `Operations` tab and click on `Start/Stop Software` on the left hand column. Click on the server(s) that you want to start/stop and then click the `Start/Stop` button.

Note that the “Suspending” of a SIP Proxy is not supported.

The following is a screenshot of the start/stop page:



Start/Stop Server

This page allows you to *start*, *stop*, *suspend*, or *resume* services on a set of VoiceGenie servers.

Please select the Clusters or Nodes or Components that you would like to perform the operation on. Then click Start, Stop, Suspend, or Resume to perform the desired operation.

```
Network
⊟ ☑ 🟥 10.0.0.190 [properties]
    ☑ 🟥 Call Manager
    ☑ 🟥 Command Line Console
    ☑ 🟥 Redundancy Manager
    ☑ 🟥 SIP Proxy
    ☑ 🟥 System Management Console
    ☑ 🟥 VoiceXML Interpreter
    ☑ 🟥 Web Proxy
```

6.2 Health Status

Health status summary can be made available through the command line console (CLC) or the SMC. Individual health status items can be retrieved through SNMP traps.

In CLC, type: `health ccpproxy`

In SMC, go to `Monitoring` tab and click on the `SIP Proxy` item.

Here is a sample output:

```
Health for SIP Proxy (ccpproxy) on 10.0.0.212
Started: 2005-01-26/17:40:38.913
Proxy State: Active
Requests forwarded: 0
Conferences created: 0
Requests forwarded in past 5 minutes: 0
Peak requests forwarded: 0
Bad Messages Received: 0
SIP Transports: UDP:10.0.0.212:5061
 Resource(10.0.0.212:5060) Ports:0/100 Registered:No Status:Up
 Resource(10.0.0.99:5060) Ports:0/100 Registered:No Status:Down
```

The following are explanations of each parameter in the health status summary:

- `Started` – The time SIP Proxy started

- `Proxy State` – The current status of the proxy. Possible states are `Initialized`, `Unavailable`, `Activating`, `Active`, `Standby`, `Soft Shutdown`, or `Unknown`.
- `Requests forwarded` – The total number of SIP requests forwarded
- `Conferences created` – The total number of conferences created
- `Requests forwarded in part 5 minutes` – The total number of SIP requests forwarded in the past 5 minutes
- `Peak requests forwarded` – The peak number of SIP requests forwarded in any 5 minute period
- `Bad messages received` – The number of bad SIP messages received and discarded
- `SIP Transports` – A list of the proxy SIP transports, including transport type (`UDP` or `TCP`), IP address, and the port number on which it listens.
- `SIP Resources` – A list of the SIP resources on each line and their operational states. It includes the host and port, current ports used/total ports available, registered state, and operational status of each resource.

The health status parameters are available through SNMP get. For a complete list of parameters and description, see Appendix A. For details about how to retrieve the information, refer to the *OA&M Framework User’s Guide.*

Note: When the bonding driver is used, the `ifconfig` command should not be used to monitor the status of the network. Instead the following command should be used:

```
cat /proc/net/bonding/bond0
```

A response similar to the example below will be displayed. In the example below, link `eth0` is up.

```
Ethernet Channel Bonding Driver: v2.6.0 (January 14, 2004)

Bonding Mode: fault-tolerance (active-backup)

Primary Slave: None

Currently Active Slave: eth0

MII Status: up

MII Polling Interval (ms): 100

Up Delay (ms): 0
```

```
Down Delay (ms): 0



Slave Interface: eth0

MII Status: up

Link Failure Count: 0

Permanent HW addr: 00:07:e9:23:db:4f



Slave Interface: eth1

MII Status: down

Link Failure Count: 0

Permanent HW addr: 00:07:e9:23:db:4e
```

6.3 Logging

SIP Proxy uses the VoiceGenie OA&M Framework to generate logging information. All logs with levels of `Critical` (`LOG_0`), `Error` (`LOG_1`), or `Warning` (`LOG_2`) are sent upstream and to the log file. The default log file is stored in `/usr/local/ccp-proxy/log/CMP.log.ccpproxy`.

Log levels for `Notice` (`LOG_3`) and `Information` (`LOG_4`) are stored in the log file.

`Trace` logs (`LOG_5`) are disabled by default. Enabling them traces all SIP messages that are sent and received by the SIP Proxy. `Trace` is not recommended for a deployment environment, because the flood of trace files can quickly degrade system performance.

To enable `trace`, go to SIP Proxy Configuration and select `false` for `cmp.trace_flag`. Click `Update` to submit the configuration change.

6.4 Cluster Management

Details for how to create and administer Clusters can be found in the following document:

VoiceGenie 7.2 OA&M Framework – SMC User’s Guide

Chapter

7 Provisioning the SIP Proxy

One can provision the SIP Proxy to properly forward incoming SIP requests through changing the following 4 configuration tables:

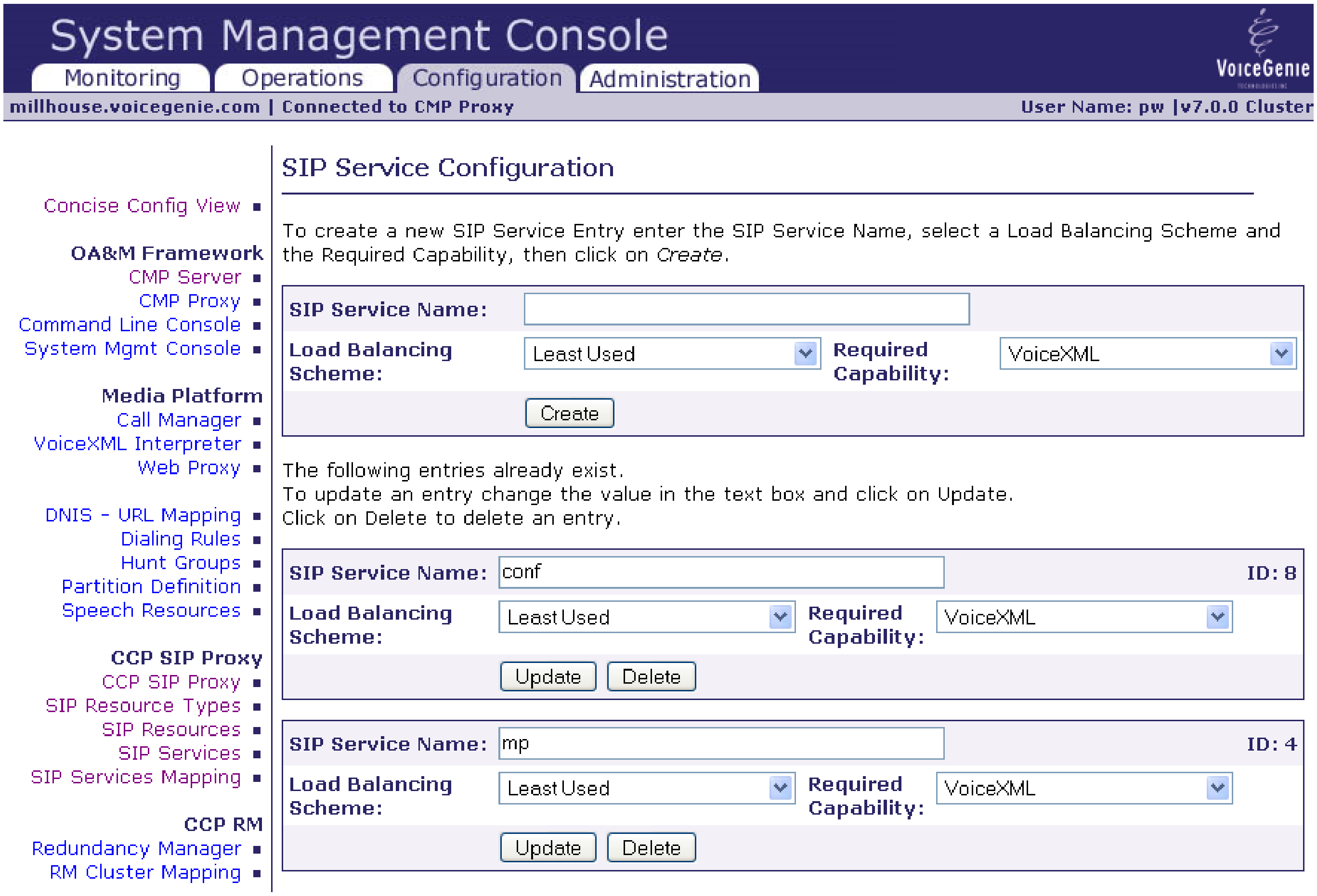
1. **SIP Service Table** defines all available SIP services within the cluster. Load balancing scheme can be defined for each SIP Service.
2. **SIP Services Mapping Table** defines a set of rules that maps incoming requests to a SIP Service. This table can also translate SIP Request URI based on regular expression rules.
3. **SIP Resource Types Table** defines a template for SIP resources that share common attributes.
4. **SIP Resources Table** defines the list of SIP resources. Each SIP Resource provides one or more SIP Service and each SIP Resource belongs to only one SIP Resource Type.

Note that these provisioning pages do not require targeting; clicking on the `Create` or the `Update` button will apply the changes automatically to all configured SIP Proxy in the cluster.

The following subsections describe the details of setting up the tables.

7.1 SIP Service Table

The SIP Service Table defines all the SIP Services in the cluster. The table can be accessed through SMC in the `Configuration` tab > `SIP Services`. The following is a screen shot of the table:



SIP Service Configuration

To create a new SIP Service Entry enter the SIP Service Name, select a Load Balancing Scheme and the Required Capability, then click on *Create*.

SIP Service Name:			
Load Balancing Scheme:	Least Used	**Required Capability:**	VoiceXML

The following entries already exist.
To update an entry change the value in the text box and click on Update.
Click on Delete to delete an entry.

SIP Service Name:	conf			**ID: 8**
Load Balancing Scheme:	Least Used	**Required Capability:**	VoiceXML	

SIP Service Name:	mp			**ID: 4**
Load Balancing Scheme:	Least Used	**Required Capability:**	VoiceXML	

Each Provisioning entry represents a row in the table. Each entry contains the following items:

- `SIP Service Name` is a string (no space) that identifies the SIP Service.
- `Load Balancing Scheme` defines the scheme of which the SIP Resource is selected. There are two schemes available:

  `Least Used` – the SIP Resource with the least number of active dialogs will be selected.

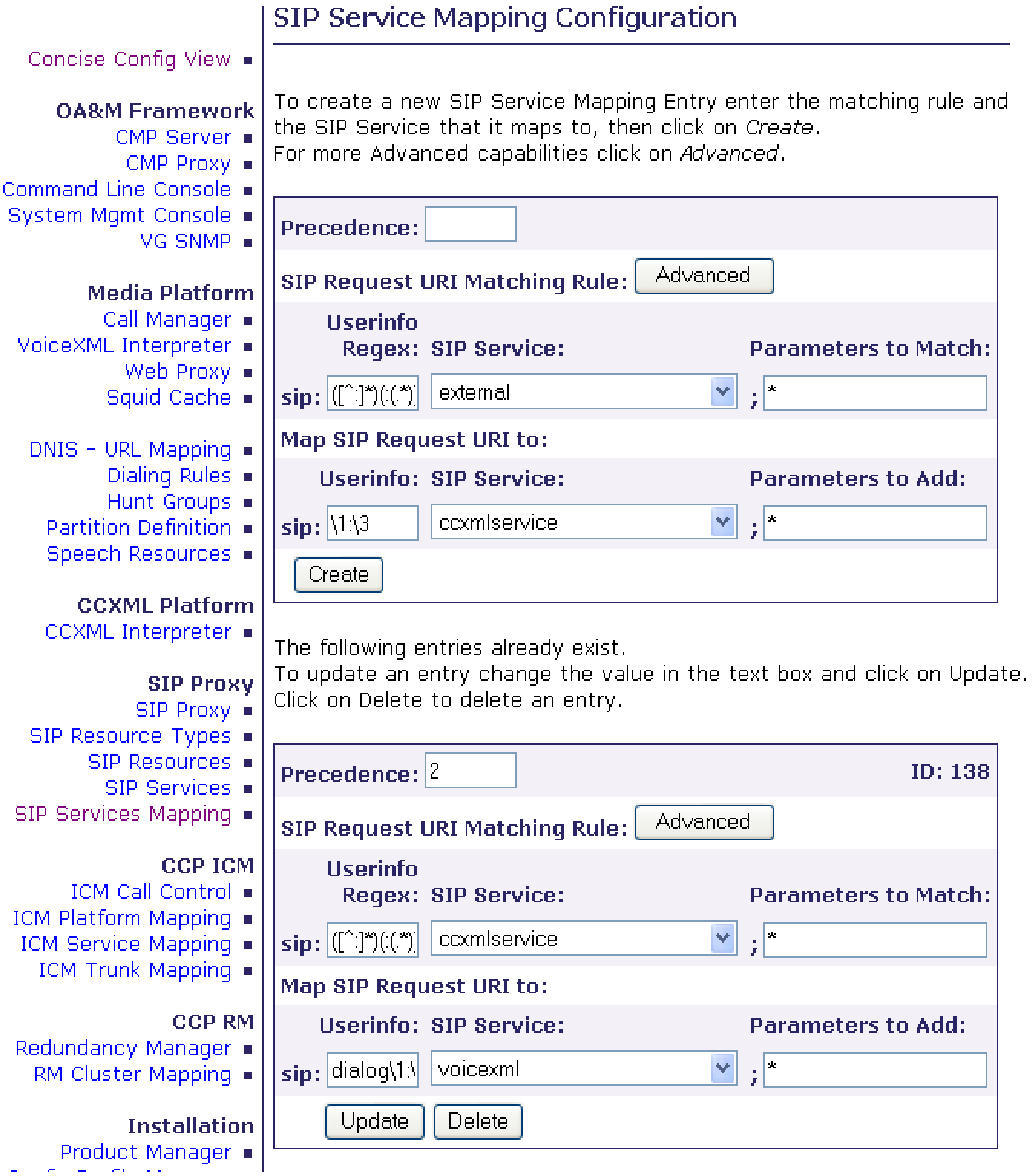
  `Round-Robin` – selects a SIP Resource in a round-robin fashion regardless of the load of the resource
- `Required Capability` – there are three types of capabilities available: `VoiceXML`, `Conference` and `CCXML`.

7.2 SIP Services Mapping Table

7.2.1 General

The SIP Service Mapping Table lists rules for mapping each incoming SIP request to a SIP Service to which it is forwarded. The rules are sorted by precedence and each SIP request is handled in that order until it is matched to a rule.

The table is accessible through SMC by clicking the `Configuration` tab > `SIP Service Mapping`.



SIP Service Mapping Configuration

To create a new SIP Service Mapping Entry enter the matching rule and the SIP Service that it maps to, then click on *Create*.
For more Advanced capabilities click on *Advanced*.

Precedence:

SIP Request URI Matching Rule:

Userinfo Regex: SIP Service: Parameters to Match:

sip: ([^:]*)(:(.*) external ; *

Map SIP Request URI to:

Userinfo: SIP Service: Parameters to Add:

sip: \1:\3 ccxmlservice ; *

The following entries already exist.
To update an entry change the value in the text box and click on Update.
Click on Delete to delete an entry.

Precedence: 2 ID: 138

SIP Request URI Matching Rule:

Userinfo Regex: SIP Service: Parameters to Match:

sip: ([^:]*)(:(.*) ccxmlservice ; *

Map SIP Request URI to:

Userinfo: SIP Service: Parameters to Add:

sip: dialog\1:\ voicexml ; *

Each provisioning entry is a rule in the SIP Service Mapping table. There are six fields in each rule:

`Precedence` defines the rule’s position relative to the other rules in the table. Rules are considered in ascending order. Precedence values need not be consecutive, but they must be unique. If there is more than one entry that contains the same precedence value, an alarm is generated and order is not guaranteed.

`Matching rules` contain three rules; all of them must be met in order for the entry to be considered a match. The use of matching rules requires the user to have some basic knowledge of regular expressions syntax and usage.

1. `Userinfo Regex` defines a regular expression. If the regular expression matches the `userinfo` of the SIP Request-URI, then the request is considered a match. Note that the `userinfo` of a SIP Request-URI is only the `user:password` portion.

Note: The `userinfo` regex must be in escaped format. This allows proper parsing of the SIP URI into the user and password fields, and is consistent with RFC 3261. The escape rules for the `userinfo` regex are similar to the rules that are defined in section 25.1 of RFC 3261. The user and password fields, however, follow different rules concerning which characters are allowed. The following is an excerpt from Section 25.1 of RFC 3261 that describes which characters are allowed in the `userinfo` portion of a SIP Request-URI.

```
userinfo         =  ( user / telephone-subscriber ) [ ":" password ]
"@"

user             =  1*( unreserved / escaped / user-unreserved )

user-unreserved  =  "&" / "=" / "+" / "$" / "," / ";" / "?" / "/"

password         =  *( unreserved / escaped /

                    "&" / "=" / "+" / "$" / "," )

unreserved  =  alphanum / mark

mark        =  "-" / "_" / "." / "!" / "~" / "*" / "'"

               / "(" / ")"

escaped     =  "%" HEXDIG HEXDIG

alphanum  =  ALPHA / DIGIT
```

The BNF for `telephone-subscriber` can be found in RFC 2806. Note, however, that any characters allowed there that are not allowed in the `userinfo` portion of the SIP URI must be escaped. An example of the use of escaped characters is provided in Example 4 in this section.

2. `SIP Service` defines the SIP service that sends the initial Request-URI. A Request-URI that does not come from a registered SIP Service is marked as external.
3. `Parameters to Match` defines a list of parameters that must be included in the SIP Request URI in order to consider the request a match. Multiple parameters can be defined in the field; it is space delimited. Use `*` to match any parameters.

Service Mapping defines three fields that translate the Request URI in order to determine where to forward the SIP message. Regular expression replacement tokens can be used in the fields. Matched groups can be expressed in tokens such as `\1, \2,` or `\3,` while matched parameters in `Parameters to Match` can be replaced with `[parameter]` tokens. A total of up to none `backreference` tokens can be used in either the `Userinfo` field or the `Parameters to Add` field. The `backreference` tokens all refer to the matched groups from the `Userinfo Regex` in the incoming Request-URI.

1. `Userinfo` defines the portion of the Request URI that will be replaced. Regex replacement tokens can be used.

Note: If a password is specified in the translated Request-URI, the separator `:` must be explicitly expressed in the translated Request-URI. The default value of `\1:\3` for the `Userinfo` portion separated the user and password portions from the incoming Request-URI, and should suffice in the majority of cases.

Note: When `backreference` tokens such as `\1` or `\3` are used, the first colon that is used is the delimiter between user and password, for example, "`\1:\3:test`". The other "`:`" is part of the `password` field. If parameter variables that contain "`:`" are used, these characters are considered to be part of the `user` or `password` fields and are therefore escaped in the resulting SIP message. For example, in the parameter variable `[param]`="`abc:def`", the translated `user-info` rule is "`test[param]`". The remaining `user-info` portion is "`testabc%3Adef`”.

2. `SIP Service` selects one SIP Service that is defined in the SIP Service Table.
3. `Parameters to Add` defines URI parameters. In addition to allowing regex replacement tokens, the wildcard `*` can be used to include all matched parameters in `Parameters to Match.`

SIP Mapping examples

Example 1: Retrieving the `user` portion of the Request-URI and passing it as a new parameter in the translated `Request-URI:`

```
Initial Request-URI: sip:alice123@atlanta.com;paramname1=paramvalue1
Userinfo Regex: ([^:]*)(:(.*))?
SIP Service: external
Parameters to Match: *

Map SIP Request-URI to:
Userinfo: \1:\3
SIP Service: ccxmlservice
Parameters to Add: * name=\1

Final Request-URI: sip:alice123@ccxmlservice;
paramname1=paramvalue1;name=alice123
```

In this example, the `backreference` token `\1` corresponds to the sub-expression that is enclosed by the first-opened parenthesis, `([^:]*)`. The matched string for `\1` is here `alice123`.

Likewise, `\2` corresponds to the subexpression matched by `(:(.*))`, and `\3` corresponds to `(.*)`, the sub-expression enclosed by the third opening parenthesis.

Here, both `\2` and `\3` do not match any string because there is no `password` field in the initial Request-URI. Also note that since there is no `password` field, the colon `:` from `\1:\3` is omitted from the final Request-URI.

In the above example, we assume that `10.0.0.190` is the SIP resource providing the service `ccxmlservice`.

The parameter `sipservice=ccxmlservice` is automatically added by the SIP Proxy in the translated Request-URI.

Example 2: Retrieving the old password from the initial Request-URI and using it in the translated Request-URI with a new user:

```
Initial Request-URI:
sip:alice:pwd@atlanta.com;paramname1=paramvalue1
Userinfo Regex: alice(:(.*))?
SIP Service: external
Parameters to Match: *

Map SIP Request-URI to:
Userinfo: bob:\2
SIP Service: ccxmlservice
```

```
Parameters to Add: *

Final Request-URI:
sip:bob:pwd@10.0.0.190;sipservice=ccxmlservice;paramname1=paramvalue
1
```

Note: The matched string for `\1` is `:pwd`. The matched string for `\2` is `pwd`.

In the above example, we assume that `10.0.0.190` is the SIP resource providing the service `ccxmlservice`.

The parameter `sipservice=ccxmlservice` is automatically added by the SIP Proxy in the translated Request-URI.

Example 3: Replacing a password with a new password that is extracted from the parameters list of the initial Request-URI:

```
Initial Request-URI: sip:alice:pwd@atlanta.com;newpassword=p%40:wd
Userinfo Regex: ([^:]*)(:(.*))?
SIP Service: external
Parameters to Match: newpassword

Map SIP Request-URI to:
Userinfo: \1:[newpassword]
SIP Service: ccxmlservice
Parameters to Add: *

Final Request-URI:
sip:alice:p%40%3Awd@10.0.0.190;sipservice=ccxmlservice;newpassword=p
%40:wd
```

In the preceding example, the value of the parameter name `newpassword` contains the colon character `:`, along with `%40`, which is the escaped character for `@`. The translated `userinfo` rule substitutes the original value in the `password` field with the value of the parameter `newpassword`. The password in the translated Request-URI is `p%40%3Awd`, where both the `@` and `:` characters are escaped according to the escape rules that were previously described.

The matched string for `\1` is “`alice`”

The matched string for `\2` is “`:pwd`”

The matched string for `\3` is “`pwd`”

In the above example, we assume that 10.0.0.190 is the SIP resource providing the service ccxmlservice.

The parameter “`sipservice=ccxmlservice`” is automatically added by the SIP Proxy in the translated Request-URI.

Example 4: Matching an escaped userinfo character in the initial Request-URI.

```
Initial Request-URI: sip:%40lice123@atlanta.com
Userinfo Regex: (%40lice[^:]*)(:(.*))?
SIP Service: external
Parameters to Match: *

Map SIP Request-URI to:
Userinfo: \1:\3
SIP Service: ccxmlservice
Parameters to Add: *

Final Request-URI: sip:%40lice123@10.0.0.190;sipservice=ccxmlservice
```

Note that the userinfo portion of the initial Request-URI “`%40lice123`” corresponds to a user field of “`@lice`” in unescaped form. The userinfo regex has been written such that the “`@`” is escaped according to the escape rules described previously and in RFC 3261.

The matched string for `\1` is “`%40lice123`”

`\2` and `\3` do not match any strings in this example.

Also note that since there is no password field, the colon “`:`” from "`\1:\3`” is omitted from the final Request-URI.

In the above example, we assume that 10.0.0.190 is the SIP resource providing the service ccxmlservice.

The parameter “`sipservice=ccxmlservice`” is automatically added by the SIP Proxy in the translated Request-URI.

7.2.2 Advanced Configuration

Due to the flexibility of the SIP Service Mapping Table, complex matching rules can be created to match anything that regular expression can offer in the `userinfo` portion of the initial Request-URI. The `Advanced` button allows the user to go into advanced mode that shows the raw form of the rules.

This is the text format of the mapping rule:

```
[order]
[resource name]
[Request URI]
[URI parameters]
[translated userinfo]
[translated hostport]
[translated URI parameters]
```

- `[order]` is the order number of the entry;
- `[resource name]` is the SIP Service name where the SIP request must originate from;
- `[Request URI]` is the regular expression that must match the *userinfo portion* of the SIP request URI. The `sip:` part of the field is ignored during the pattern matching process;
- `[URI parameters]` is a space delimited string of URI parameters that must exist in the SIP Request URI;
- `[translated userinfo]` is the userinfo portion of the Request URI to be replaced. Regex replacement tokens can be used;
- `[translated hostport]` can be a SIP service name or an actual hostname;

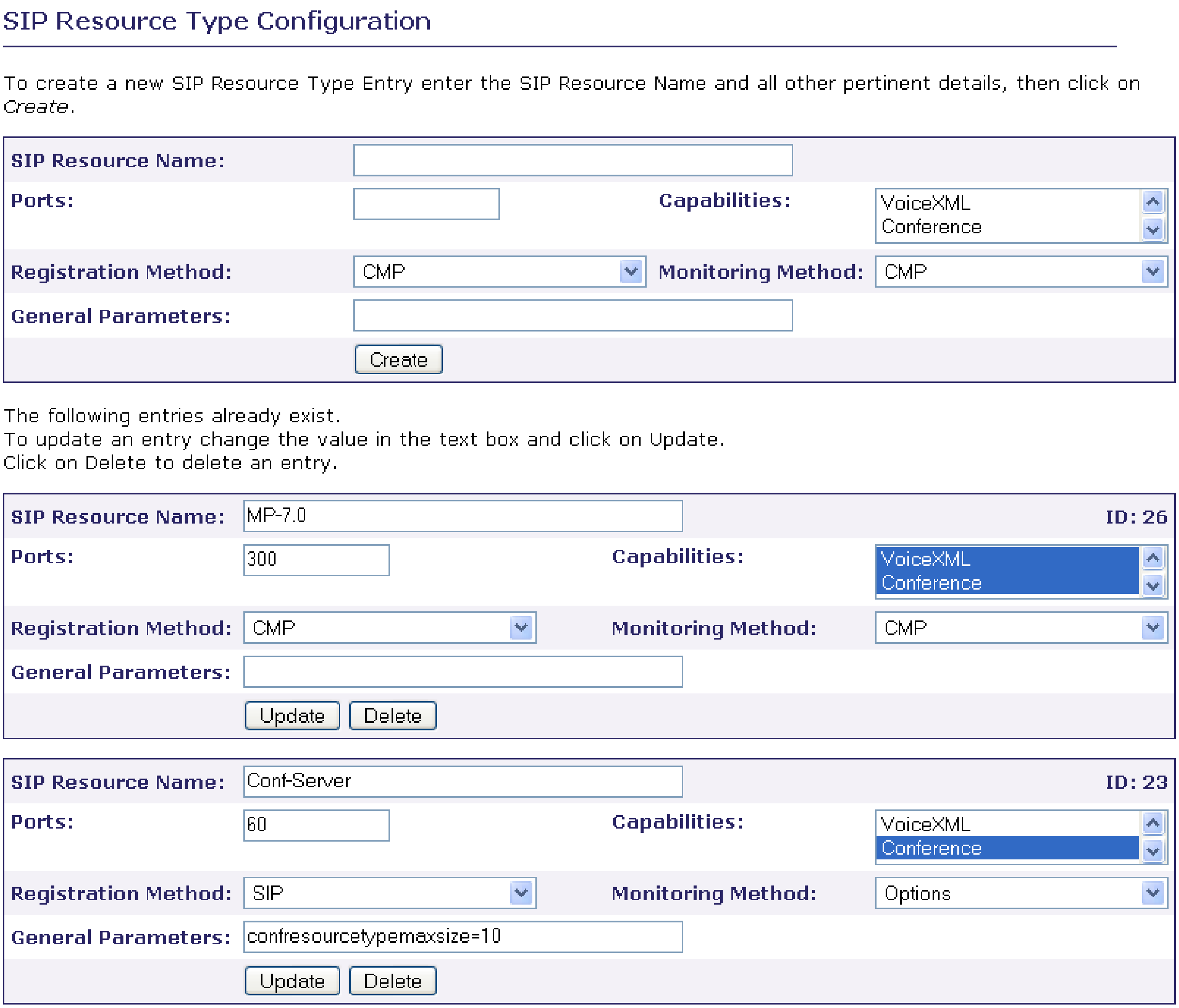
- `[translated URI parameters]` is a space delimited string of URI parameters that will be in the replaced Request URI. Regex replacement tokens can be used.

Note: Please note that the `sip`: part of the `[Request URI]` field is automatically added by the SMC if the “simplified” view of the SIP Service Mapping configuration is used to create the new rule.

The user can modify the `sip:` string of the `[Request URI]` field in the `Advanced Configuration` page and the regex in the `Advanced Configuration` view will be used. However, if the user then modifies the same mapping rule from the simplified Configuration, the `sip:` string will be added back to the `[Request URI]` field and any previous changes made from the `Advanced Configuration` page will be lost.

7.3 SIP Resource Types Table

The SIP Resource Types table defines common attributes for each SIP Resource Type. The following is a screenshot of the table:



SIP Resource Type Configuration

To create a new SIP Resource Type Entry enter the SIP Resource Name and all other pertinent details, then click on *Create*.

SIP Resource Name:			
Ports:		**Capabilities:**	VoiceXML Conference
Registration Method:	CMP	**Monitoring Method:**	CMP
General Parameters:			
	Create		

The following entries already exist.
To update an entry change the value in the text box and click on Update.
Click on Delete to delete an entry.

SIP Resource Name:	MP-7.0		**ID: 26**
Ports:	300	**Capabilities:**	VoiceXML Conference
Registration Method:	CMP	**Monitoring Method:**	CMP
General Parameters:			
	Update Delete		

SIP Resource Name:	Conf-Server		**ID: 23**
Ports:	60	**Capabilities:**	VoiceXML Conference
Registration Method:	SIP	**Monitoring Method:**	Options
General Parameters:	confresourcetypemaxsize=10		
	Update Delete		

There are 6 fields for each SIP Resource Type:

1. `SIP Resource Name` – the name of the SIP Resource Type
2. `Ports` – the number of ports available for the SIP Resource belonging to this SIP Resource Type
3. `Capabilities` – defines the type of capabilities that this SIP Resource Type can provide.
4. `Registration Method` – defines how the SIP Resource registers with SIP Proxy. Registering provides up to date information to the SIP Proxy about how to forward SIP messages to the resource. There are 3 methods supported:

`SIP` – The resource will send SIP `REGISTER` message periodically. SIP Proxy expects an expiry period of 60 seconds and will modify the Contact header with an expiry parameter of 60.

`CMP` – VoiceGenie products that are integrated into the VoiceGenie common OA&M infrastructure will select this value in order for the proxy to get real-time status information about the resource.

`None` – If the non-VoiceGenie resource type does not support SIP `REGISTER`, select this parameter.

5. `Monitoring Method` – SIP Proxy monitors the resource with one of the following 3 methods:

`OPTIONS` – The proxy can monitor SIP resources by sending a periodic `OPTIONS` request to validate that the SIP resource is responding correctly. This allows the proxy to be aware within a relatively short period (configurable with a default of two minutes) if a resource fails.

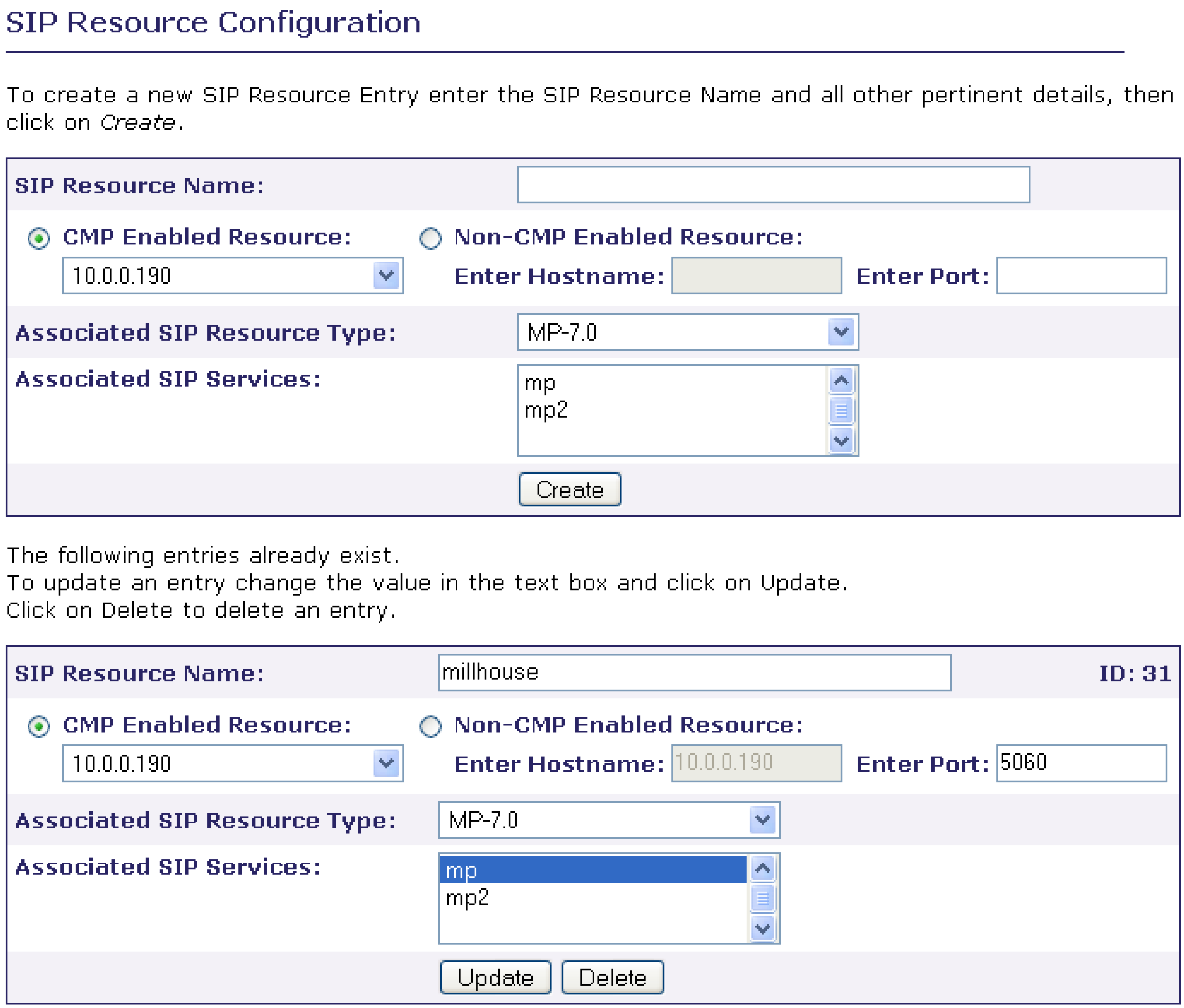
`CMP` – SIP Proxy receives a periodic heartbeat from VoiceGenie products via the the VoiceGenie common OA&M infrastructure.

`None` – the resource is incapable of handling SIP `OPTIONS` message and SIP Proxy assumes the resource is always available.

6. `General Parameters` – any other attributes that belongs to the resource type will go into this text field. The field accepts a comma-delimited string of attribute-value pairs. An example of the parameter is `confresourcetypemaxsize=10`. Please refer to Appendix C of *SIP Proxy System Reference Guide* for a complete list of SIP Resource Type parameters.

7.4 SIP Resources Table

The SIP Resource Table lists all the SIP Resources available in the cluster. The following is a screenshot of the table:



SIP Resource Configuration

To create a new SIP Resource Entry enter the SIP Resource Name and all other pertinent details, then click on *Create*.

SIP Resource Name:

● CMP Enabled Resource: ○ Non-CMP Enabled Resource:

10.0.0.190 Enter Hostname: Enter Port:

Associated SIP Resource Type: MP-7.0

Associated SIP Services: mp
mp2

The following entries already exist.
To update an entry change the value in the text box and click on Update.
Click on Delete to delete an entry.

SIP Resource Name: millhouse ID: 31

● CMP Enabled Resource: ○ Non-CMP Enabled Resource:

10.0.0.190 Enter Hostname: 10.0.0.190 Enter Port: 5060

Associated SIP Resource Type: MP-7.0

Associated SIP Services: mp (selected), mp2

There are 4 fields in a SIP Resource entry:

1. `CMP Enabled Resource` – click the radio button to select a VoiceGenie resource. When this radio button is selected, `Enter Hostname` field will be grayed out. Use the drop-down list to select the host of the resource. `Enter Port` field is the SIP port defaults to `5060`.
2. `Non-CMP Enabled Resource` – click the radio button to select a non-VoiceGenie resource. The drop-down list will be grayed out. Enter the hostname of the SIP resource in `Enter Hostname` field. Enter the SIP port as well; the port defaults to `5060`.
3. `Associated SIP Resource Type` – this is a drop-down list that contains the names of all SIP Resource Types. Select one type to make this resource as part of this SIP Resource Type.

4. `Associated SIP Services` – this is a multi-select field that shows all SIP Services available. `Ctrl`-click on all SIP services that you wish the SIP Resource to be part of.