Avaya Solution & Interoperability Test Lab

Application Notes for Avaya Voice Portal 5.1, Avaya Aura® Session Manager 5.2, and Acme Packet Net-Net 3800 Integration with Skype Connect 2.0 –Issue 1.0

Abstract

These Application Notes describe the steps to enable calls between Avaya Voice Portal 5.1 and the PSTN through an Avaya Aura® SIP trunk solution with Skype Connect 2.0.

Skype Connect allows PSTN users and Skype registered users to place telephone calls into Avaya Voice Portal 5.1 and interact with customer self-service applications. Avaya Voice Portal also enables these callers to be transferred to Avaya Contact Centers to speak with contact center representatives. Avaya Voice Portal is a Web services-based, speech enabled interactive voice response system.

Testing was conducted at the Avaya Solution & Interoperability Test Lab utilizing a traditional Internet T1 ISP circuit for accessing the Skype Connect 2.0 service directly over the Internet.

Table of Contents

1. Introduction

These Application Notes describe the steps to configure Avaya Voice Portal 5.1 and the Avaya Aura® SIP trunk solution with Skype Connect, using an Internet-based connection. Businesses may choose to purchase Skype Online Numbers to receive calls from the PSTN. Separately, the Skype User community can dial Skype Business Accounts to connect to self-service applications running on Avaya Voice Portal. Avaya Voice Portal can also transfer these callers to an Avaya Aura® Communication Manager Contact Center to speak with contact center representatives. Access to a broadband Internet connection is required.

In the reference configuration discussed within these Application Notes, PSTN and Skype users dial Skype Online Numbers or Skype Business Accounts to access self-service applications running on the Avaya Voice Portal. In addition, users of Avaya telephones homed on Avaya Aura® Communication Manager, including contact center representatives, can receive transferred calls. Both Avaya Voice Portal 5.1 and Skype Connect 2.0 support G.711 Mu-law, G.711 A-law, and G.729 audio codecs.[1]

For more details regarding the Avaya Aura® SIP trunk solution with Skype Connect, please see **Reference [1]**.

For more information on the Skype Connect service, see **Reference [13]**.

1.1. Reference Configuration

Figure 1 illustrates the reference configuration validated in the Solution and Interoperability Test Lab. All of the Avaya Customer Premise Equipment (CPE) is located on a private IP network. The "inside" interface of the Acme Packet SBC is also connected to this private network. The "outside" interface of the Acme Packet SBC is connected to a Juniper edge router that provides access to the Internet via a traditional T1 connection. This Internet connection is used for traditional Internet access as well as access to the Skype Connect service.

The Avaya CPE is comprised of Avaya Aura® Session Manager and Avaya Voice Portal. In addition, Avaya telephones homed on Avaya Aura® Communication Manager are used as ACD[2] agent positions and can receive calls that are transferred from Voice Portal. It should be noted that, in the reference configuration, inbound calls from the PSTN to Voice Portal do not initially traverse Communication Manager, but are routed by Session Manager from the Acme SBC to Voice Portal. The same holds true for calls from Skype Users to Voice Portal. In the opposite direction, outbound calls originated by Voice Portal are routed by Session Manager to terminate on Communication Manager endpoints or the Acme SBC for routing to the PSTN via the Skype Connect service.[3]

---

[1] Skype delivers all calls with G.729 as the preferred codec. The release of Voice Portal documented in these Application Notes will always answer both G.729 and G.711 calls, picking whichever codec has been offered as the preferred codec in the SDP.

[2] Automatic Call Distribution. Programming of Avaya Aura® Call Center features is beyond the scope of these Application Notes. See **References [11-12]**.

[3] While transfer scenarios are included in these Application Notes, outbound calls to the PSTN initiated by Voice Portal are not included these Application Notes.

After the caller interacts with the self-service application on Avaya Voice Portal, Voice Portal can then transfer the call to Communication Manager Callers using one of three transfer methods: blind, consultative (also known as supervised), or bridged. For blind transfers, Voice Portal uses the SIP REFER method.[4] For supervised transfers, Voice Portal can be programmed to use INVITE with REPLACES or REFER to transfer calls to Communication Manager. Although INVITE with REPLACES is the preferred method in cases where verification of ACD availability is desired prior to transfer, this method does not currently provide audible feedback to the original caller during the transfer in this configuration. Hence, the REFER method is used here for supervised transfers. For bridged transfers, Avaya Voice Portal utilizes a second VoIP channel to initiate a call to the transfer destination and bridges the original caller with the transfer destination. Note that this method of call transfer utilizes two VoIP channels on Voice Portal, one for the original caller and one for the transfer destination for the duration of the call.

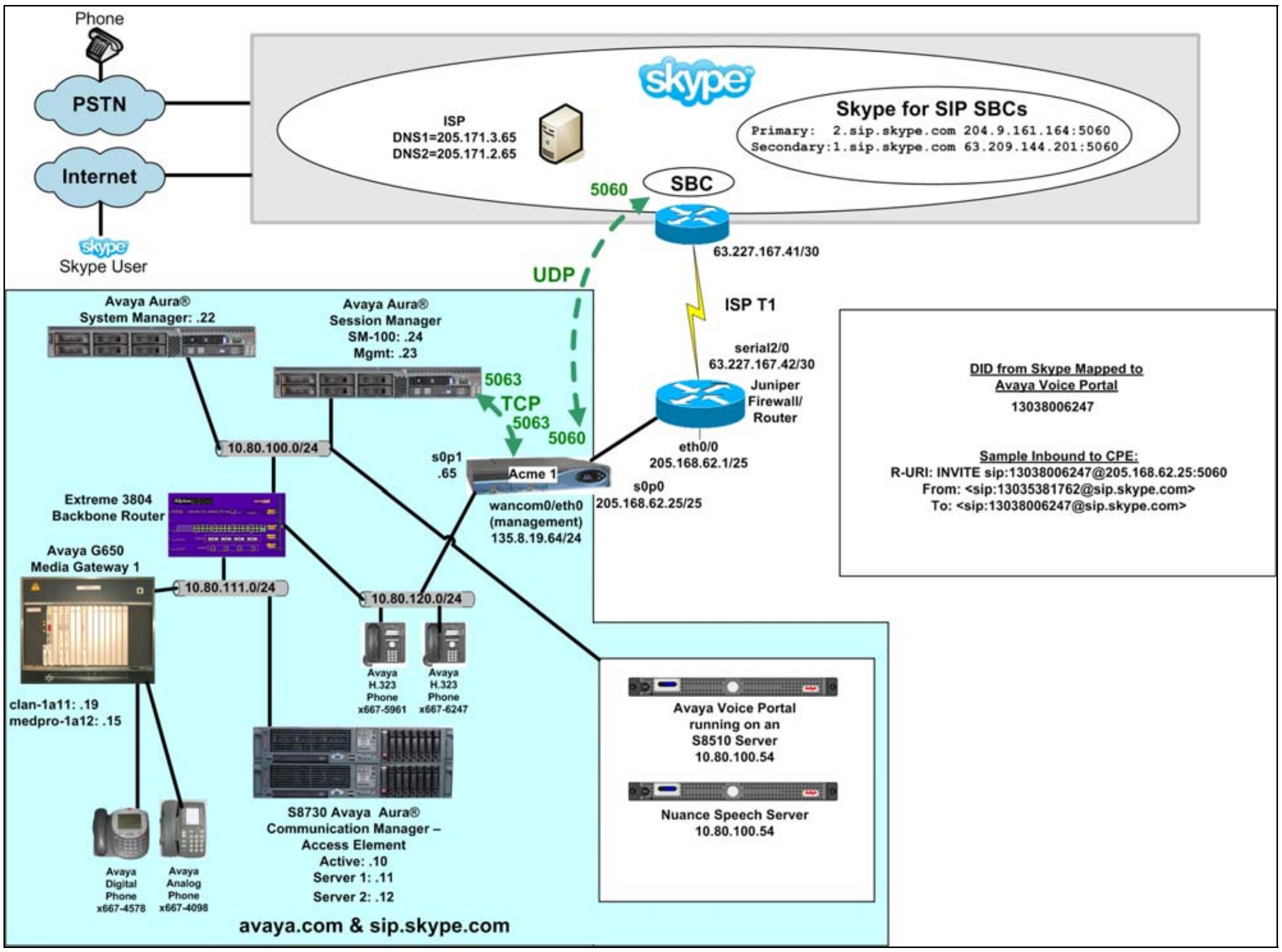

Figure 1: Reference Configuration

In the configuration described in these Application Notes, the Acme SBC is programmed to locally terminate a SIP REFER received from Voice Portal. In this sequence, the SBC does not send the SIP REFER to Skype and is programmed to terminate the SIP REFER locally. In response to the

[4] The Acme Packet SBC was programmed to locally process the SIP REFER method. No SIP REFER method was sent to the Skype network for these call flows. **Reference [15]** contains a detailed description of this capability known as SIP REFER Method Call Transfer.

SIP REFER, the SBC sends a SIP INVITE to Session Manager, inviting the party that is identified in the REFER-TO header of the REFER message. If the identified destination resides on Communication Manager, such as the Communication Manager ACD, Session Manager then routes the call to Communication Manager. The same call flow holds true for calls from Skype users.

The installation and provisioning of the ISP T1 circuit is not part of the Skype Connect service.

The Skype Connect service uses a domain of *sip.skype.com.* The Avaya CPE environment can be assigned a domain of either *sip.skype.com* or *avaya.com*. In this configuration, the Voice Portal is assigned a domain of *sip.skype.com*.

In addition to the components discussed in **Reference [1]**, the following components are used in the reference configuration and are discussed in detail in subsequent sections.

> **Note –** The domains and IP addressing specified in these Application Notes apply only to the reference configuration shown in **Figure 1**. Skype Connect customers will use their own domains and IP addressing as required.

- Skype Online Number for incoming calls to Avaya Voice Portal
- Avaya Voice Portal Audio Codec
- Avaya Voice Portal SIP Connection
- Avaya Voice Portal Application Assignment

1.1.1 Skype Online Number for Incoming Calls

For inbound calls to Avaya Voice Portal, a Skype Online Number is provisioned that provides a Direct Inward Dial (DID) 11 digit number. When calls arrive to this DID number, Avaya Aura® Session Manager uses a Dial Pattern to route the calls to Avaya Voice Portal. See **Section 4.3.8**.

1.1.2 Audio Codec

Skype delivers all calls with G.729 as the preferred codec. The release of Voice Portal documented in these Application Notes will always answer both G.729 and G.711 calls, picking whichever codec has been offered as the preferred codec in the SDP.

For bridged transfers and outbound calls from Voice Portal, the Voice Portal can be programmed to utilize G.729, G.711 Mu-law or G.711 A-law. This can be achieved on Voice Portal by changing the MPP server’s VoIP settings. See **Section 3.2.**

1.2. Dialing Examples

The following are examples of inbound and outbound voice calls.

1.2.1 Inbound Calls to Avaya Voice Portal from the PSTN

A Skype Online Number is used to route calls to Voice Portal. When this number is dialed, Skype delivers the call to the Acme Packet SBC. The Acme Packet SBC then delivers the call to Session Manager for routing. Session Manager then delivers the call to the Voice Portal.

An entry in the Voice Portal Applications table is used to direct calls to the correct self-service application. See **Figure 2**.

<u>Inbound from the PSTN</u>

- PSTN dials Skype Online Number (13038006247) and the Skype Connect service sends the call to the Acme Packet SBC at the Avaya CPE.
- The Acme Packet SBC passes the call to Avaya Aura® Session Manager. Avaya Aura® Session Manager performs Dial Pattern analysis and sends the call to Avaya Voice Portal.
- Avaya Voice Portal launches the assigned self-service application.
- Avaya Voice Portal performs a call transfer to the Avaya Aura® Communication Manager.

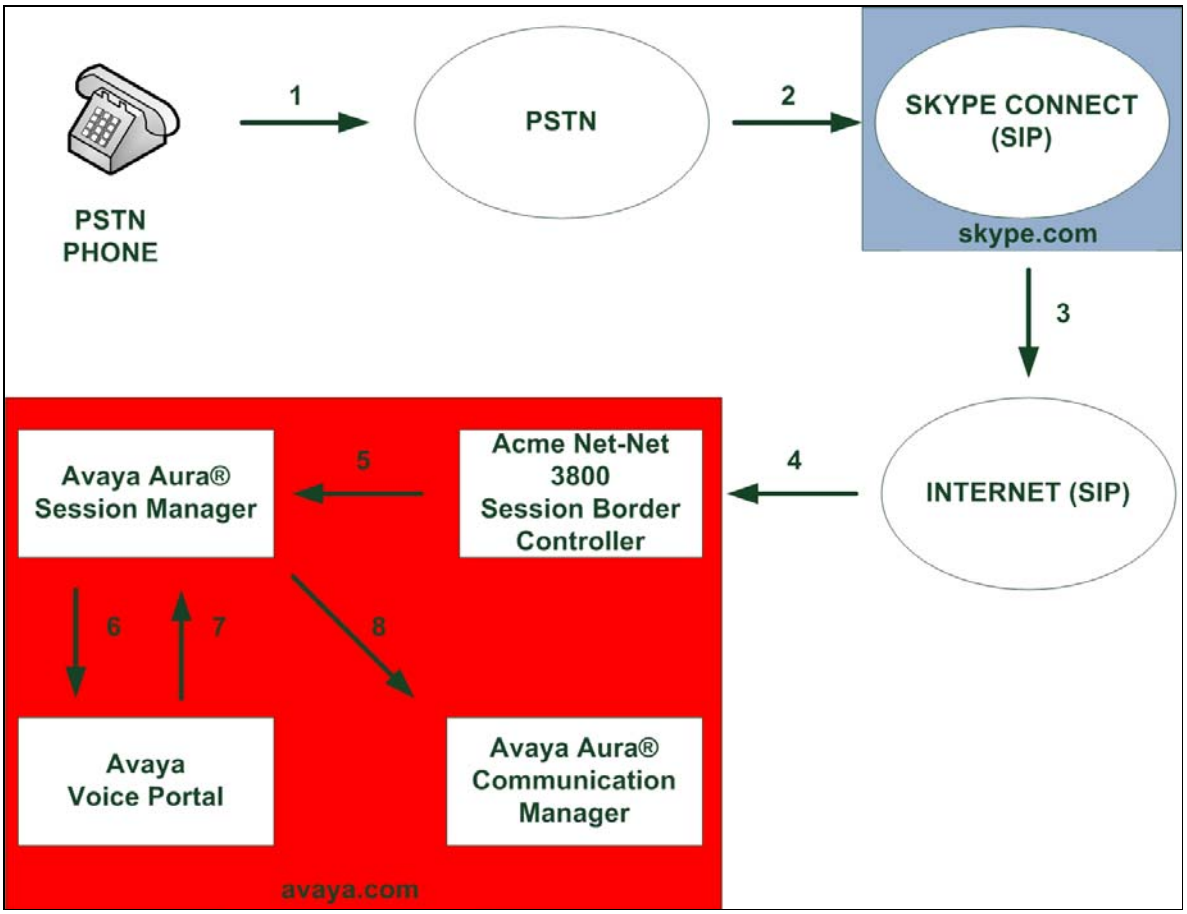

Figure 2: Inbound Call Flow from PSTN Phone

1.2.2 Inbound Calls to Avaya Voice Portal from Skype Users

Skype Users can also dial a Skype Business Account which is used to route calls to Voice Portal. When this Skype Business Account is dialed, Skype delivers the call to the Acme Packet SBC. The Acme Packet SBC then delivers the call to Session Manager for routing. Session Manager then delivers the call to the Voice Portal.

An entry in the Voice Portal Applications table is used to direct the call to the correct self-service application. See **Figure 3**.

Inbound from Skype User

- Skype User dials Skype Business Account (avayavoiceportal) and the Skype Connect service sends the call to the Acme Packet SBC at the Avaya CPE using the assigned number (13038006247).
- The Acme Packet SBC passes the call to Avaya Aura® Session Manager. Avaya Aura® Session Manager performs Dial Pattern analysis and sends the call to Avaya Voice Portal.
- Avaya Voice Portal launches the assigned self-service application.
- Avaya Voice Portal performs a call transfer to the Avaya Aura® Communication Manager.

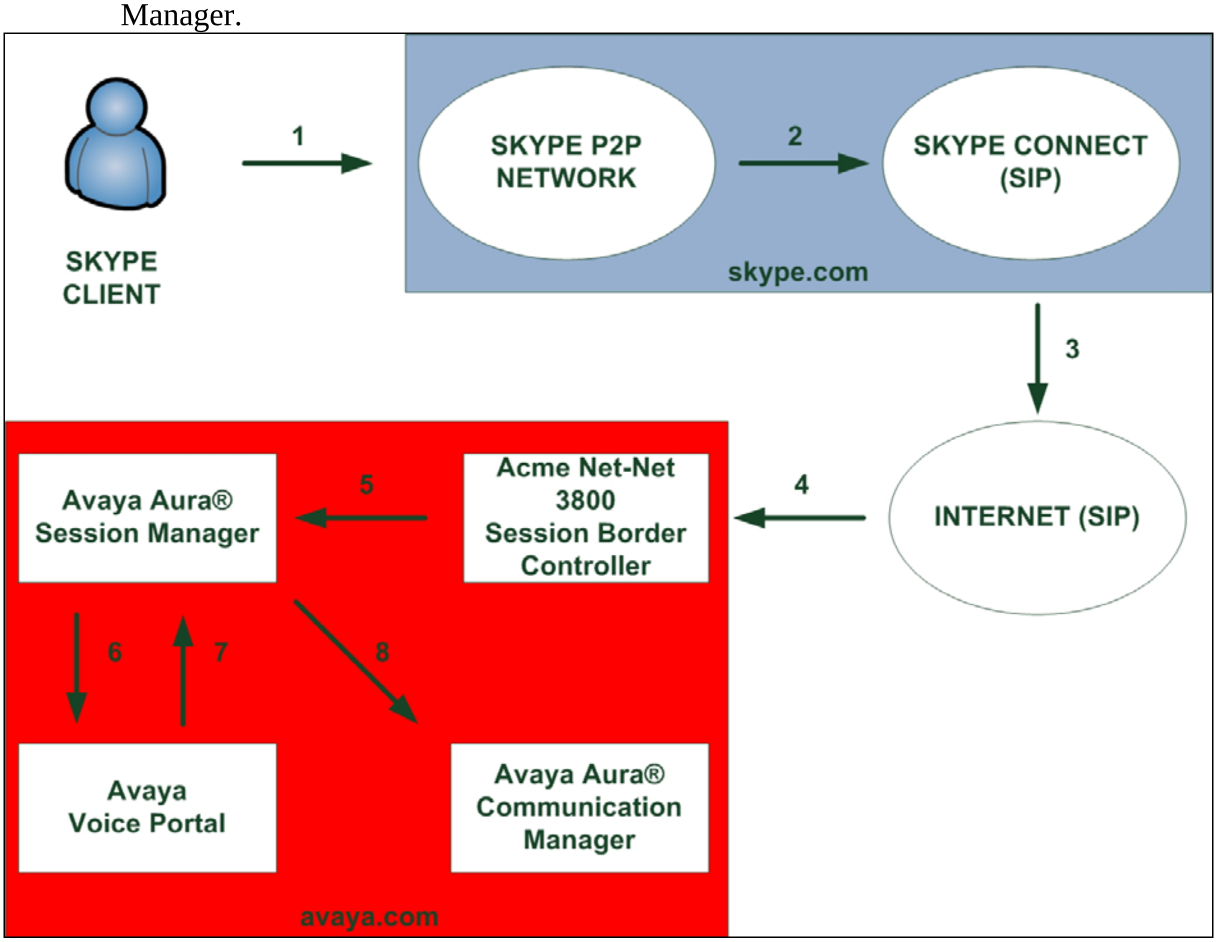

Figure 3: Inbound Call Flow from Skype User

1.2.3 Local to Foreign Domain Conversion for Outbound Calls

As mentioned in **Section 1.1** the Avaya CPE environment used a domain of *avaya.com*, and the Skype Connect service used a domain of *sip.skype.com*. For outbound calls, the Skype Connect service requires that the domain be *sip.skype.com* in the SIP request URI and To: header.

In the reference configuration, this was accomplished in Avaya Voice Portal by setting the **SIP Domain:** field on the **SIP Connection** form to *sip.skype.com*. This does not preclude the use of other methods for Domain conversion.

1.3. Known Limitations

The following limitations are noted for the reference configuration described in these Application Notes:

- See **Reference [1]** for a list of known limitations regarding the general use of the Skype Connect and the Avaya Aura® SIP trunk solution.
- PSTN or Skype callers that are transferred by Voice Portal to a PSTN destination will not hear audible ringback. However, the call will ring at the transfer destination and can be answered normally. This limitation occurs when using blind transfer. Hence, it is recommended that a consultative transfer be used and that audible feedback be played by Voice Portal during the transfer. See **Section 3.4** for details.
- For inbound calls, Voice Portal will always answer both G.729 and G.711 calls, picking whichever codec is listed first in the incoming SIP INVITE.
- For blind or supervised transfers to Communication Manager, the transfer leg of the call must use the same audio codec as the original incoming call. If the transfer leg uses a difference audio codec, ringback will not be heard by the original caller during the transfer while the call is ringing at the Communication Manager destination. After the Communication Manager destination answers the call, two-way audio path is available normally.
- For blind or supervised transfers to Communication Manager ACD via a VDN (Vector Directory Number), the call vector associated with the VDN should be programmed to initiate an announcement or to play music as the first step, if there are no agents immediately available to take the call. This results in the generation of a SIP 200 OK message by Communication Manager and allows the transfer sequence to be completed. In addition, this allows audible feedback to be provided in a timely manner to the original caller.
- On transfers to Communication Manager, the Voice Portal can inject User-to-User (UUI) information in the REFER header using the VoiceXML parameter **aai** (See **Appendix B** for a sample application that uses the **aai** tag.). Header manipulation rules are defined on the SBC to support passing UUI information to Communication Manager. Note that for bridged transfers, Voice Portal does not use REFER, but does use an INVITE which can include User-to-User information. **References [11-12]** provide complete programming information regarding Avaya Aura® Call Center Call Vectoring.
- On supervised transfers to Communication Manager that use REFER, it was observed that occasionally distorted ringback is heard by the original caller. This limitation is under investigation.

- This solution does currently support outbound SIP calls to Skype names. Outbound calls can be placed to a Skype user's Online Number.

Note – These Application Notes describe the provisioning used for the reference configuration shown in **Figure 1**. Other configurations may require modifications to the provisioning described in this document.

2. Equipment and Software Validated

The following equipment and software were used in the reference configuration.

Equipment	Firmware	Software
Avaya S8510 Server running Avaya Voice Portal (VPMS/MPP Single-Server Installation)	-	5.1
Nuance Speech Server running on a Dell PowerEdge 860 Server		5.0
Nuance Recognizer		9.0
Nuance RealSpeak		4.0
Avaya Aura® Session Manager	-	5.2. SP2 (5.2.2.0.522009)
Avaya Aura® System Manager		5.2 SP2 (5.2.2.0.522002)
Avaya S8730 Servers running Avaya Aura® Communication Manager	-	R015x.02.1.016.4 with SP4.01 (18433)
Avaya G650 Media Gateway IPSI – TN2312BP CLAN – TN799DP MEDPRO – TN2302AP VAL – TN2501AP	 HW15 FW49 HW01 FW38 HW2 FW57 HW03 FW021	 - - -
Avaya 9620 and 9630 H.323 IP Telephones	-	3.110b (H.323)
Avaya 2420 Digital Phones	-	-
Analog Phones	-	-
Acme Packet Net-Net 3800	-	SCX6.2.0 MR-3 GA (Build 619)
Skype (for PC)	-	4.2.0.169
Skype Connect		2.0

Table 1: Equipment and Software Used in the Reference Configuration

3. Configure Avaya Voice Portal

This section describes the steps for configuring Avaya Voice Portal with the necessary signaling and media characteristics for the SIP trunk connection with the Skype Connect service.

Note - The initial installation, configuration, and provisioning of the Avaya server(s) for Avaya Voice Portal are presumed to have been previously completed and are not discussed in these Application Notes.

The procedures for configuring Avaya Voice Portal include the following items:

- Configure VoIP Connection
- Configure the VoIP Audio Format
- Add an Application
- Restart the MPP

It is assumed, that Voice Portal is installed, configured and licensed as per **References [2-5].** The following instructions also assume the user is logged in to the Avaya Voice Portal.

3.1. Configure VoIP Connection

As shown in **Figure 4,** under **System Configuration**, select **VoIP Connections.** Then, select the **SIP** tab (not shown). Click **Add.** Configure the following settings to enable SIP connectivity on Voice Portal:

- Enter a name in the **Name field** (e.g. ASM1-5.2).
- Under **Enable**, select **Yes**.
- Set the **Proxy Transport** drop-down menu to **TCP**.
- Select the **Proxy Servers** radio button.
- Specify the IP address of the Session Manager's SIP interface in the **Proxy Server Address** field (e.g. **10.80.100.24**).
- Set the **Proxy Server Port** and **Listener Port** fields to **5060** for TCP.
- Set the **SIP Domain** field (e.g. **sip.skype.com**).
- Under **Consultative Transfer**, select **REFER**.
- For testing purposes, verify that the **Maximum Simultaneous Calls** is set to a minimum of **1**.
- All other values can be left at their defaults.
- Click **Apply**.
- Click **Save**.

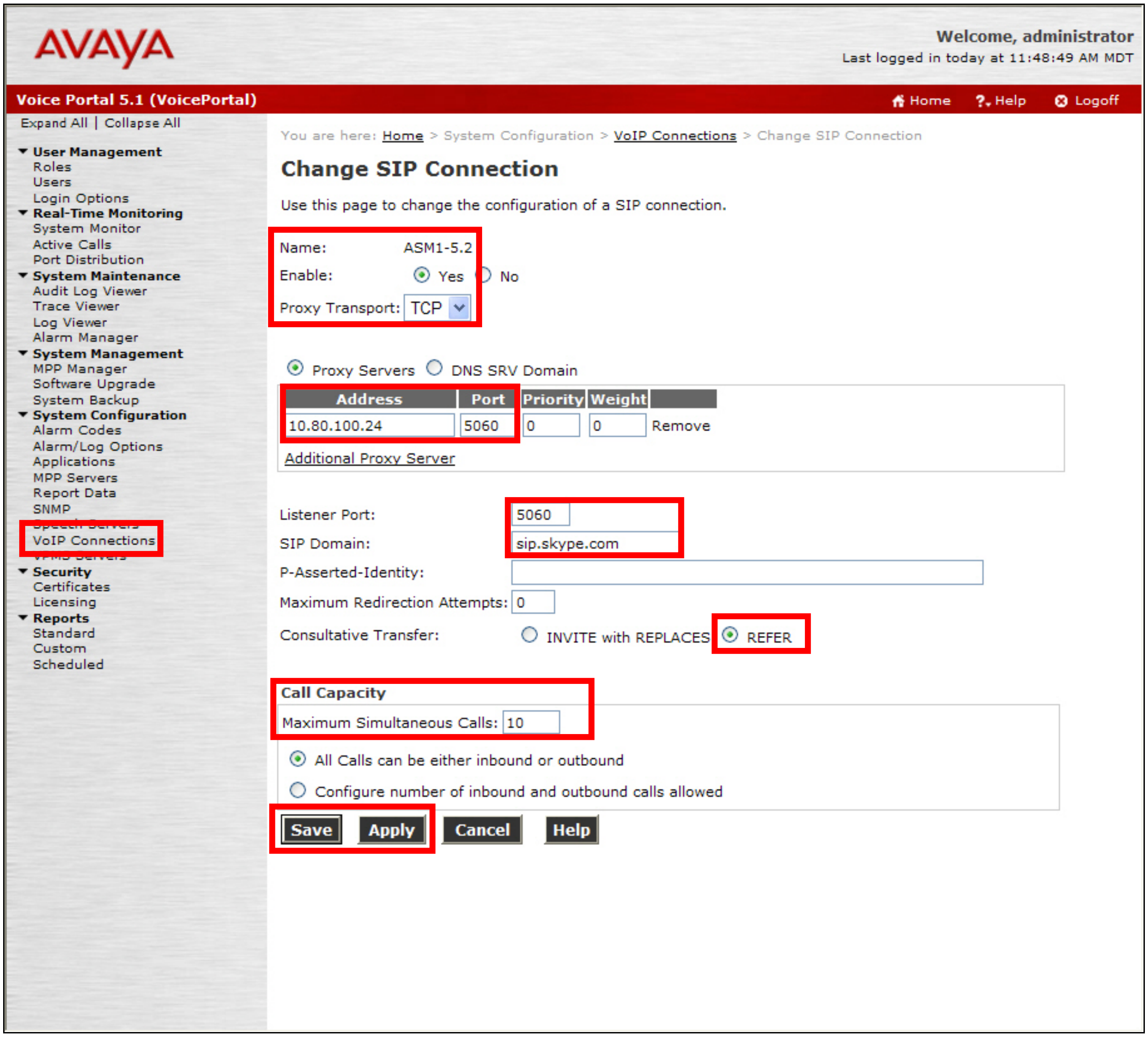

Figure 4: SIP Connection

3.2. Configure the VoIP Audio Format

Under **System Configuration**, select **MPP servers**. From the **MPP Servers** page shown in **Figure 5**, click on **VoIP Settings**.

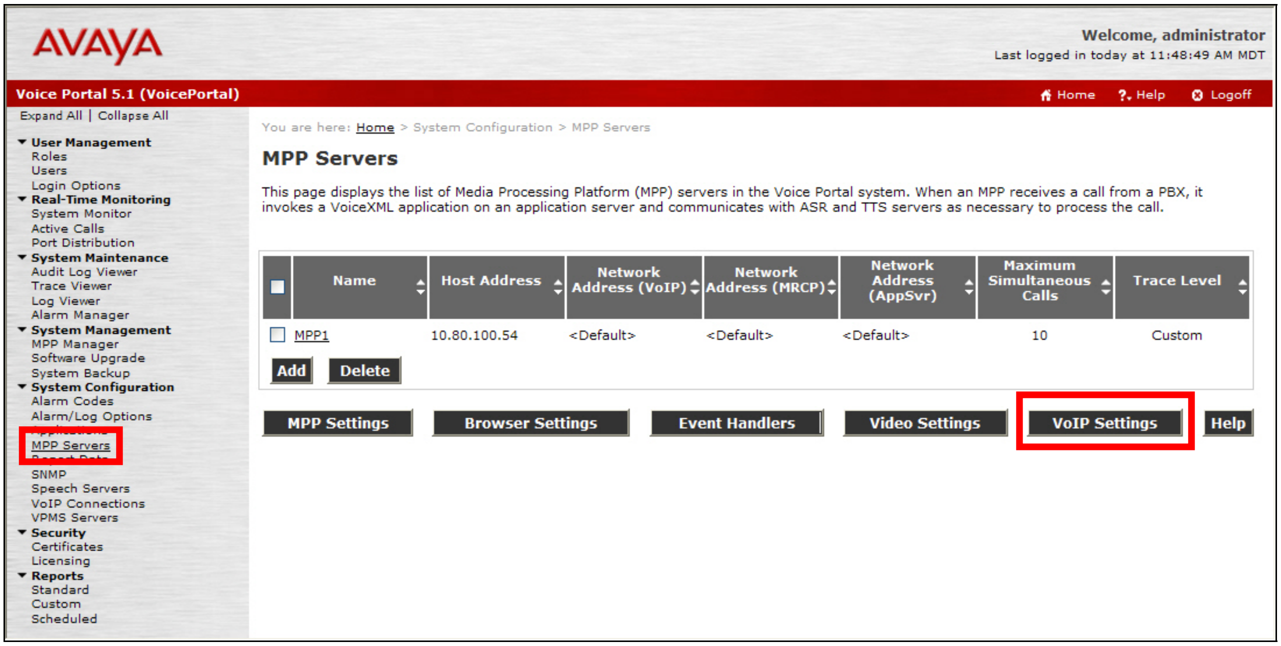

Figure 5: MPP Servers

Next, on the **VoIP Settings** page as shown in **Figure 6**, verify the following settings:

- Under **Audio Codecs**, set the **Packet Time** drop-down menu to **20**.
- Set G729 to **Yes**.
- Set the **First Offered** drop-down menu to **G729**.
- All other values can be left at their defaults.
- Click **Apply**.
- Click **Save**.

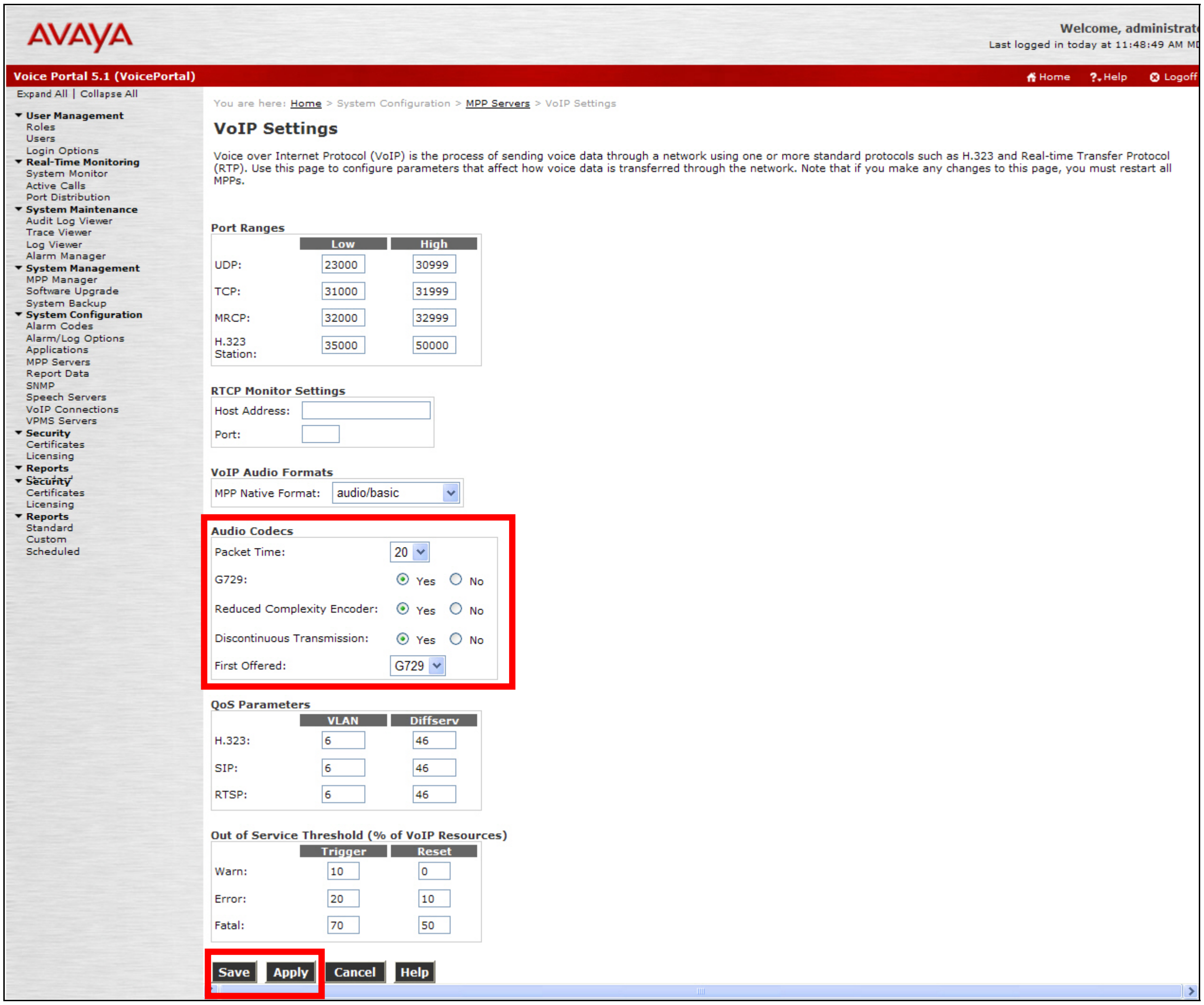

Figure 6: VoIP Settings

3.3. Add an Application

Under **System Configuration**, select **Applications** and click **Add** (not shown). From the **Add Application** page, set the following values:

- Enter a name for the application in the **Name** field (e.g. Test_App).
- Under **Enable**, select **Yes**.
- Set the **Type** drop-down menu to **VoiceXML**.
- Under **URL**, enter the URL that points to the VoiceXML test application on the application server. In this case, a standard test application located on the Voice Portal server is used. See **Appendix B** for a description of a VXML test application used for verification purposes on the Voice Portal.
- Set the **ASR** and **TTS** drop-down menus to **Nuance**.
- Under **Application Launch**, select Inbound.

- In the **Called Number** field, enter the Skype Online Number that is routed to Voice Portal (e.g. 13038006247) and click **Add**.
- All other values can be left at their defaults.
- Click **Save**.
- On the Applications page (not shown), click on the application name added above and verify the values on the **Change Application** page as shown in **Figure 7**.

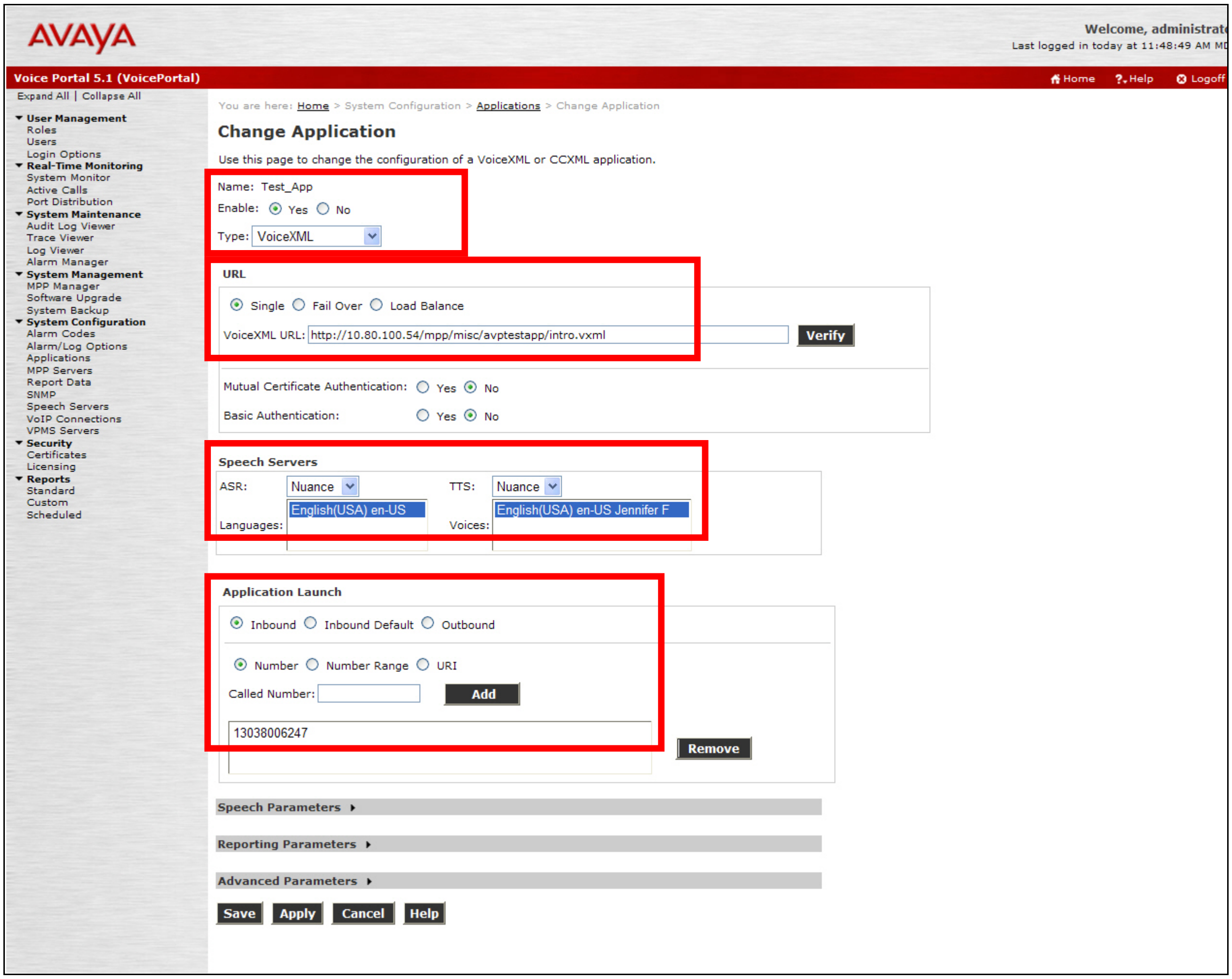

Figure 7: Change Application

3.4. Audible Feedback During Consultative Transfers

In the reference configuration, it was observed that some call scenarios do not provide audible feedback to the original caller while the call is being transferred. In these situations, it is suggested that the Voice Portal application play audio to the original caller while the call is being transferred. This can be achieved using the **consultation** type transfer method and the **transferaudio** parameter. The **transferaudio** parameter instructs Voice Portal to play audible feedback to the original caller during the transfer sequence.

- transferaudio="music.wav"
- type="consultation"

These parameters instruct Voice Portal to play the audio contained in the **music.wav** file to the caller while the consultative transfer is attempted. See **Appendix B** for a sample test application that uses these parameters.

3.5. Restart the MPP

After the configuration changes are made, restart the Voice Portal MPP Server. Under **System Management**, select **MPP Manager** as shown in **Figure 8** and click **Restart**.

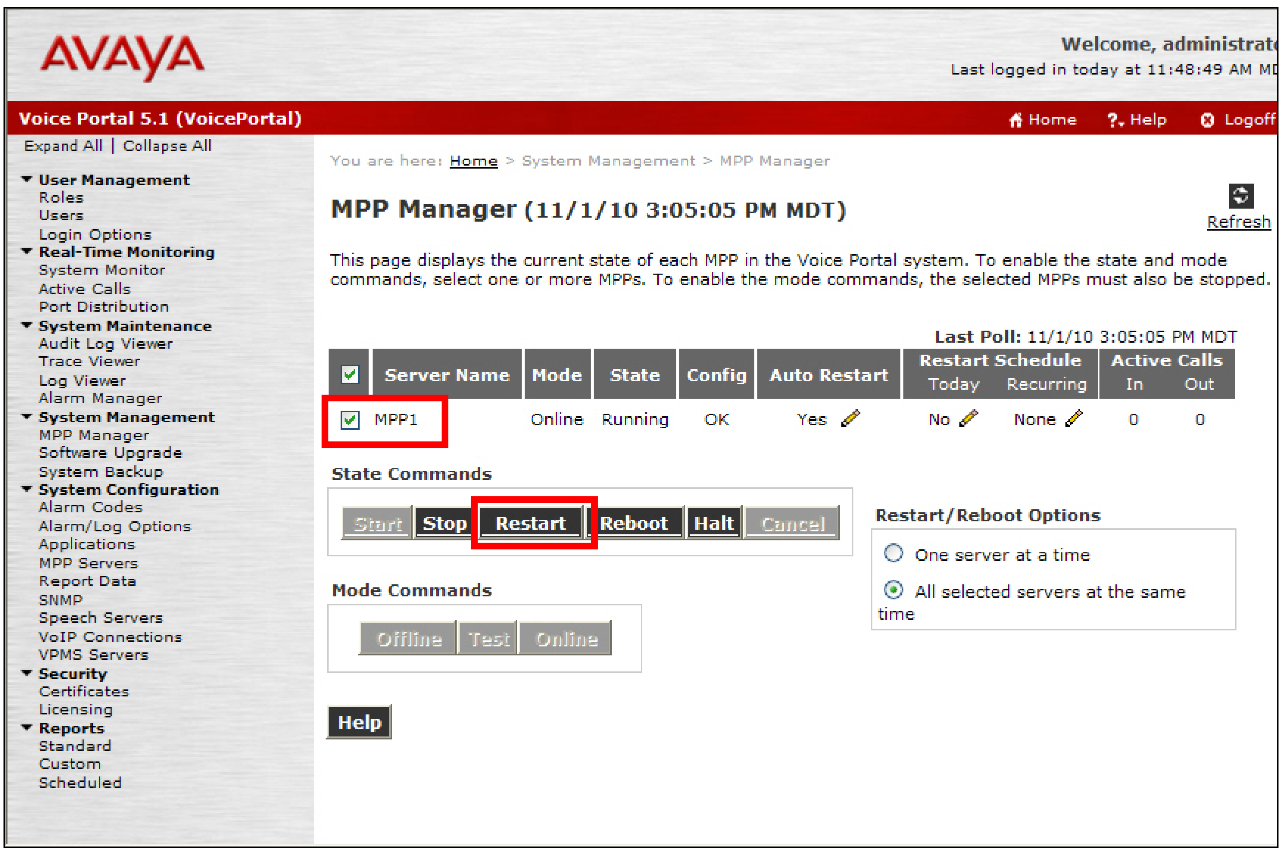

Figure 8: MPP Manager

4. Avaya Aura® Session Manager Provisioning

This section provides the procedures for configuring Avaya Aura® Session Manager as provisioned in the reference configuration. Avaya Aura® Session Manager is comprised of two functional components: the Avaya Aura® Session Manager server and the Avaya Aura® System Manager management server. All SIP call provisioning and system programming for Avaya Aura® Session Manager is performed via the System Manager web interface and are then downloaded into Avaya Aura® Session Manager.

Note – The following sections assume that Avaya Aura® Session Manager and System Manager have been installed and that network connectivity exists between the two platforms. For more information on Avaya Aura® Session Manager see **References [7-10]**.

4.1. Network Interfaces

Avaya Aura® Session Manager 5.2 is comprised of two main components, the server itself and the SM-100 card, which is embedded in the server. **Figure 9** shows the backplane of Avaya Aura® Session Manager.

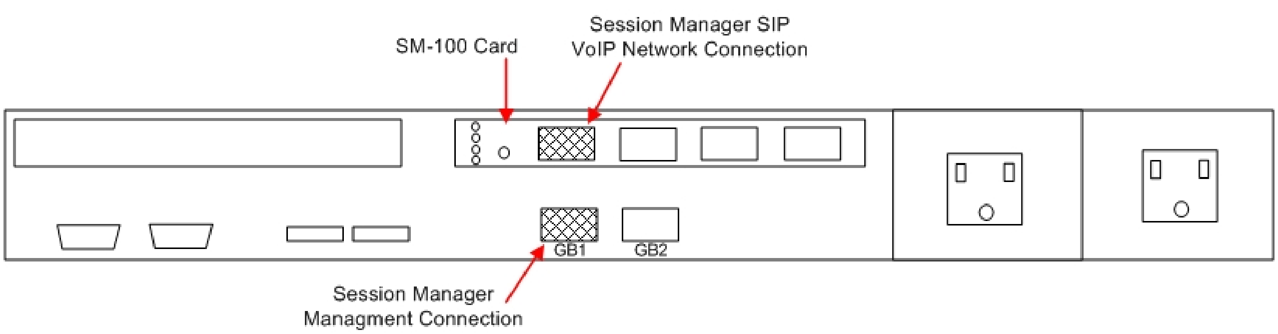

Figure 9: Avaya Aura® Session Manager Network Connections

The Avaya Aura® Session Manager SM-100 card has four network interface ports. The first port is the Avaya Aura® Session Manager connection to the SIP VoIP network. This interface is used for all inbound and outbound SIP signaling and must have network connectivity to all provisioned SIP Entities (see **Section 4.3.4**).

The Avaya Aura® Session Manager server has two network interface ports labeled “GB1” and “GB2”. The “GB1” port is used for management/provisioning of Avaya Aura® Session Manager. This port must have network connectivity to System Manager.

Note –In the reference configuration the SM-100 interface and the Avaya Aura® Session Manager server interface were both connected to the same IP network. If desired, the System Manager/Avaya Aura® Session Manager management connection may use a different network than the SM-100 connection.

4.2. System Manager

Reference [1] contains details on the base configuration required to implement the Avaya SIP trunk solution with Skype Connect. Following is a summary of the provisioning which is performed via System Manager to enable SIP trunking:

- **Network Routing Policy**
  - **SIP Domains** - Define FQDNs that may send calls to Avaya Aura® Session Manager.
  - **Locations** – Logical/physical areas that may be occupied by SIP Entities.
  - **SIP Entities** – Typically devices corresponding to the SIP telephony systems including Avaya Aura® Session Manager and other devices such as SBCs.
  - **Entity Links** – Connection information which define the SIP trunk parameters used by Avaya Aura® Session Manager when routing calls to/from other SIP Entities.
  - **Dial Patterns** – Matching digit patterns which govern to which SIP Entity a call is routed.
  - **Routing Policies** - Policies that determine call routing between the SIP Entities based on applicable Dial Patterns.
  - **Time Ranges** – Specified windows during which SIP call processing is permitted for particular Routing Policies.
- **Avaya Aura® Session Manager** – Information corresponding to the Avaya Aura® Session Manager Server to be managed by System Manager.

In System Manager Release 5.2, the URL to access the browser-based GUI of System Manager is ***https://<ip-address of System Manager>/SMGR***. Log in with the appropriate credentials.

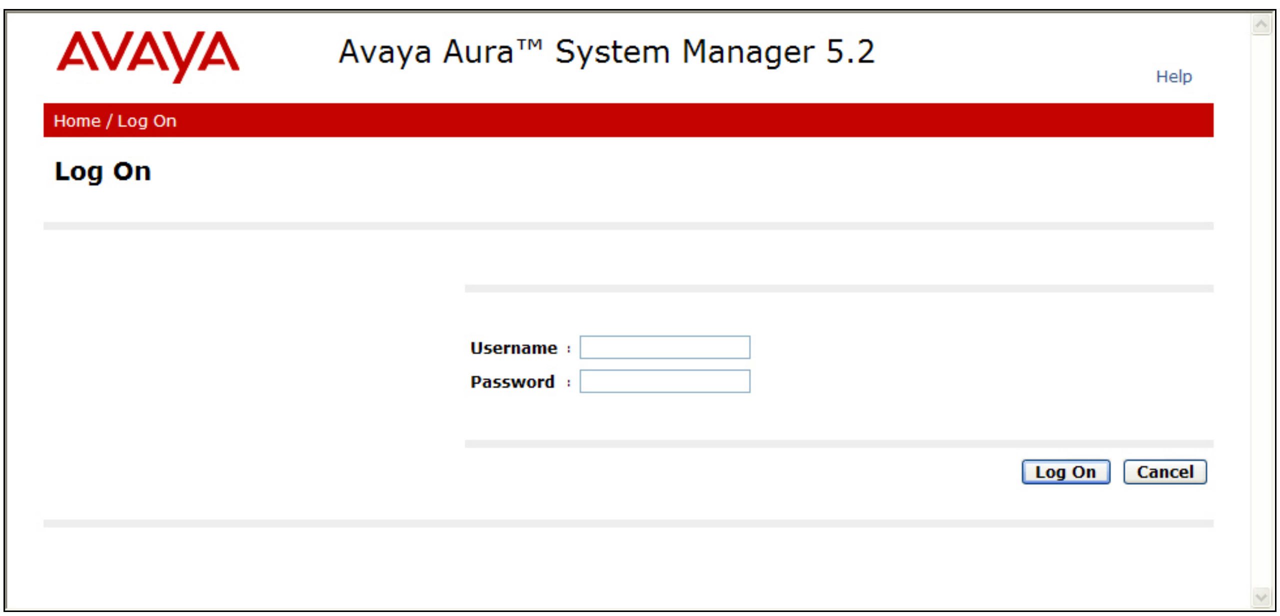

Figure 10: System Manager GUI Log On Screen

4.3. Network Routing Policy

After logging in, the menu shown in **Figure 11** is displayed. Expand the **Network Routing Policy Link** on the left side as shown.

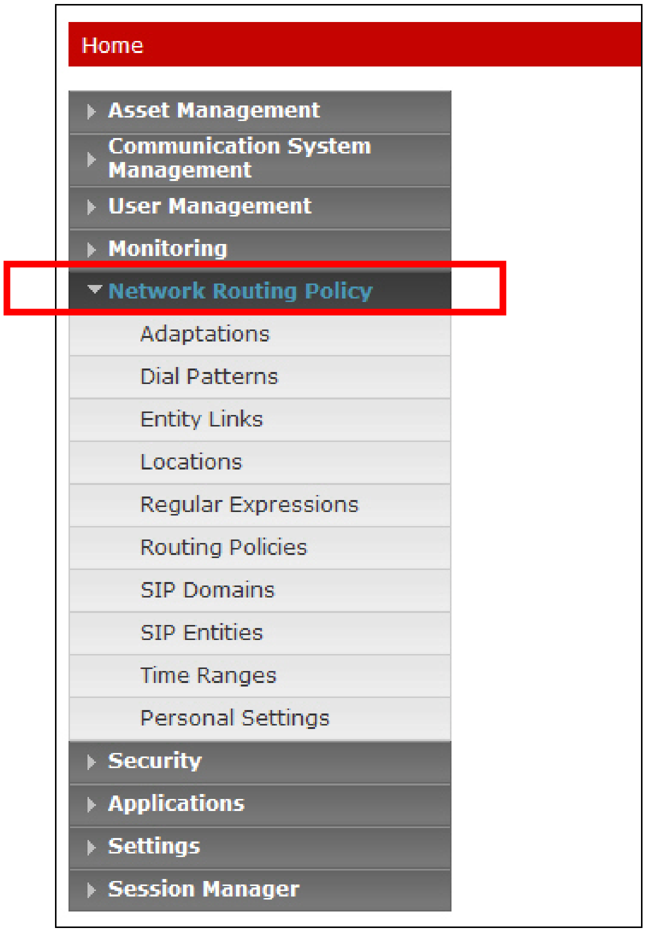

Figure 11: Network Routing Policy Menu

4.3.1 SIP Domains

In the reference configuration two SIP domains (FQDNs) are used. The Avaya CPE location is *avaya.com* and the Skype Connect service is *sip.skype.com*. The Skype Connect domain *sip.skype.com* is used for bi-directional calls between the Avaya CPE and the Skype Connect service. The Avaya CPE location uses *avaya.com for* calls internal to the Avaya CPE location. Therefore, both of these FQDNs must be provisioned in Avaya Aura® Session Manager.

1. Select **SIP Domains** from the menu.
2. Select **New**.
3. Enter the SIP Domain in the **Name** field.
4. Enter a description in the **Notes** field if desired.
5. Repeat these steps for each SIP Domain. When completed, the SIP Domain window will look like **Figure 12**.
6. Click on the **Commit** button.

Note – On most of the following forms, to edit or delete an entry, click the box next to the item to select it, to make the Edit and Delete buttons available.

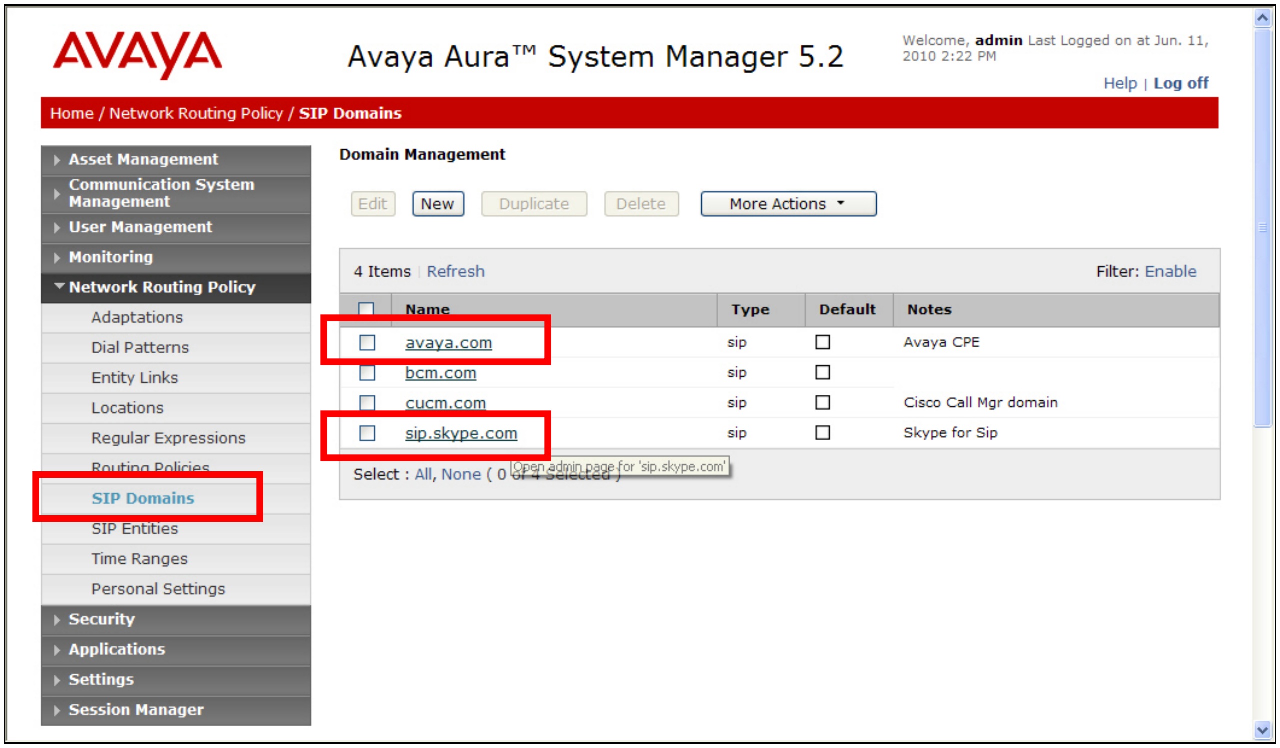

Figure 12: SIP Domain Menu

4.3.2 Adaptations

In the reference configuration, no adaptations are used between Voice Portal and Session Manager.

4.3.3 Locations

Locations are used to identify logical and/or physical locations where SIP Entities reside. Named locations are assigned with an IP Address Pattern. Locations may also be used for bandwidth management purposes for outbound calls from Avaya CPE to Skype, if required. In the reference configuration, multiple locations are defined for the Avaya CPE and one location is defined for the Acme Packet SBC. However, the bandwidth management capability was not utilized.

To add a Location, select **Locations** in the left **Network Routing Policy** menu and click on the **New** button on the right. The screen shown in **Figure 13** will open.

1. Enter a descriptive Location name in the **Name** field (e.g. AvayaCPE).
2. Enter a description in the **Notes** field if desired.
3. Under the **Location Pattern** heading, click on **Add**.
4. Enter IP address information for the Location (e.g. **10.80.100.***)
5. Enter a description in the **Notes** field if desired.
6. Repeat steps 3 to 5 if the Location has multiple IP segments.

7. Modify the remaining values on the form, if necessary; otherwise, use the default values.
8. Click on the **Commit** button.
9. Repeat all the steps for each new Location.

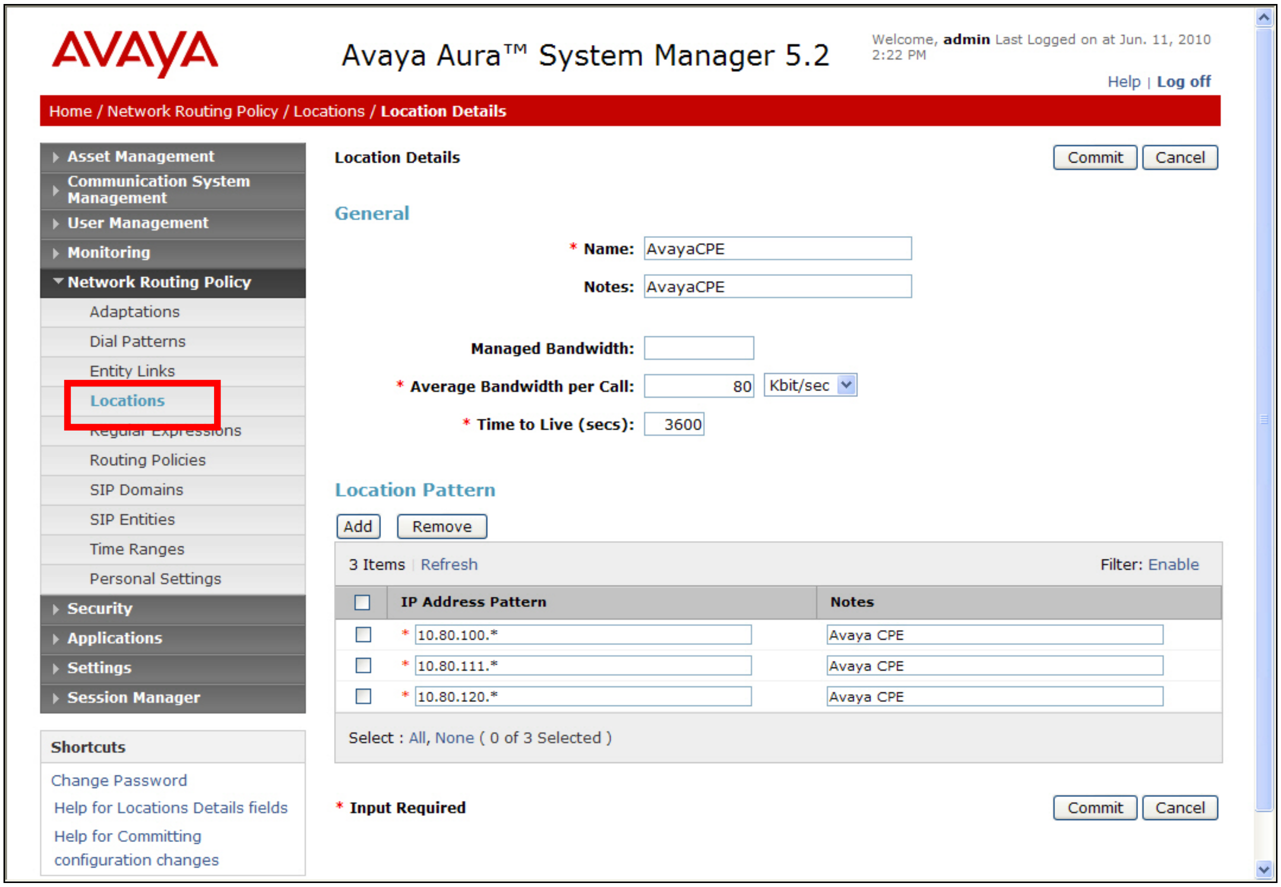

Figure 13: Location Details

4.3.4 SIP Entities

A SIP Entity must be added for Avaya Aura® Session Manager and for each network component that has a SIP trunk provisioned to Avaya Aura® Session Manager. In the reference configuration, SIP Entities are provisioned for:

- Avaya Voice Portal
- Acme Packet SBC
- Avaya Aura® Session Manager
- Avaya Aura® Communication Manager. This SIP Entity is defined in the base configuration documented in **Reference [1]**.

To add a SIP Entity, select **SIP Entities** on the left **Network Routing Policy** menu and click on the **New** button on the right. The screen shown in **Figure 14** is displayed.

1. **General** Section

a. Enter a descriptive name in the **Name** field (e.g. **Voice Portal**).
b. Enter the IP address for the SIP Entity (e.g. **10.80.100.54**).
c. From the **Type** drop down menu select a type that best matches the SIP Entity (e.g. **Voice Portal**).
d. Enter a description in the **Notes** field if desired.
e. From the **Adaptations** drop down menu, select the adaptation required for this Entity (see **Section 4.3.2**).
    i. For the Voice Portal Entity, no adaptation is defined in the reference configuration.
    ii. For the Acme SBC Entity, no adaptation is defined in the reference configuration.
f. From the Locations drop down menu select **AvayaCPE**.
g. Select the appropriate time zone.
h. Accept the other default values.

2. **Sip Link Monitoring** section
    a. Accept the default values.
3. Click on **Commit**.
4. Repeat these steps for each SIP Entity

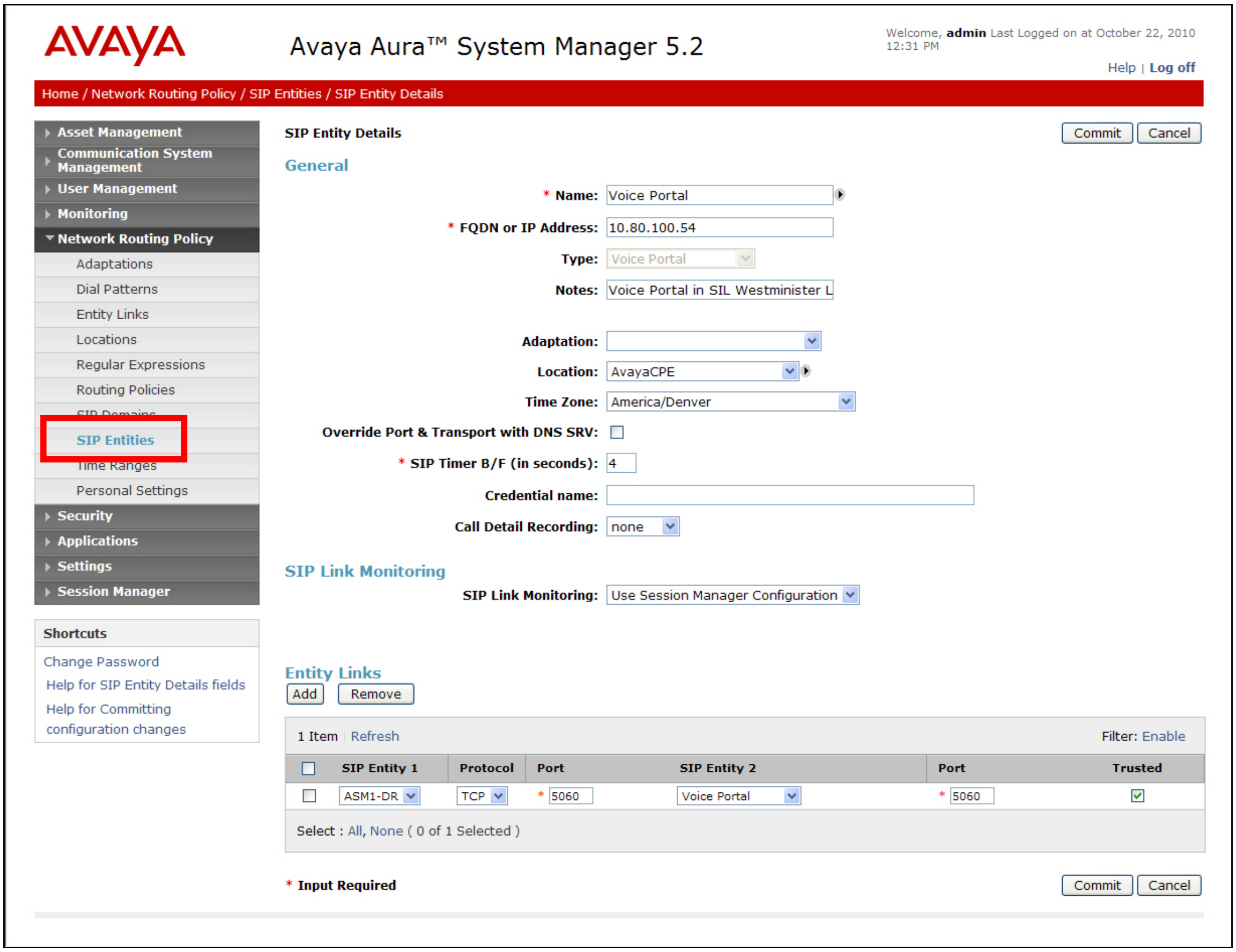

Figure 14: Voice Portal SIP Entity Details

Note – When defining a SIP Entity for Avaya Aura® Session Manager itself and SM is selected from the Type drop down menu, an additional section called Ports will appear. In this section add the transport protocol, port and FQDN used by Avaya Aura® Session Manager. In the reference configuration the values used are 5060, TCP and the Avaya domain.

The following SIP Entity values are specified in the reference configuration. SIP Entity Type "Other" can be used for the Acme Packet SBC SIP Entity.

Name	IP Address	Type	Adap-tation	Location	Port	Protocol	Domain
Voice Portal	10.80.100.54	Voice Portal	-	AvayaCPE	5060	TCP	Avaya
ASM1-DR	10.80.100.24	Session Manager	-	AvayaCPE	5060	TCP	Avaya
ACME1	10.80.120.65	Other	-	AvayaCPE	5063	TCP	Skype Connect

Name	IP Address	Type	Adap-tation	Location	Port	Protocol	Domain
S8730-port-5063[5]	10.80.111.19	Communication Manager	Skype DigitConversionAdapter	AvayaCPE	5063	TCP	Skype Connect

Table 2: SIP Entity Provisioning

Figure 15 shows a complete SIP Entities list. The SIP Entities relevant to the reference configuration are listed in **Table 2.**

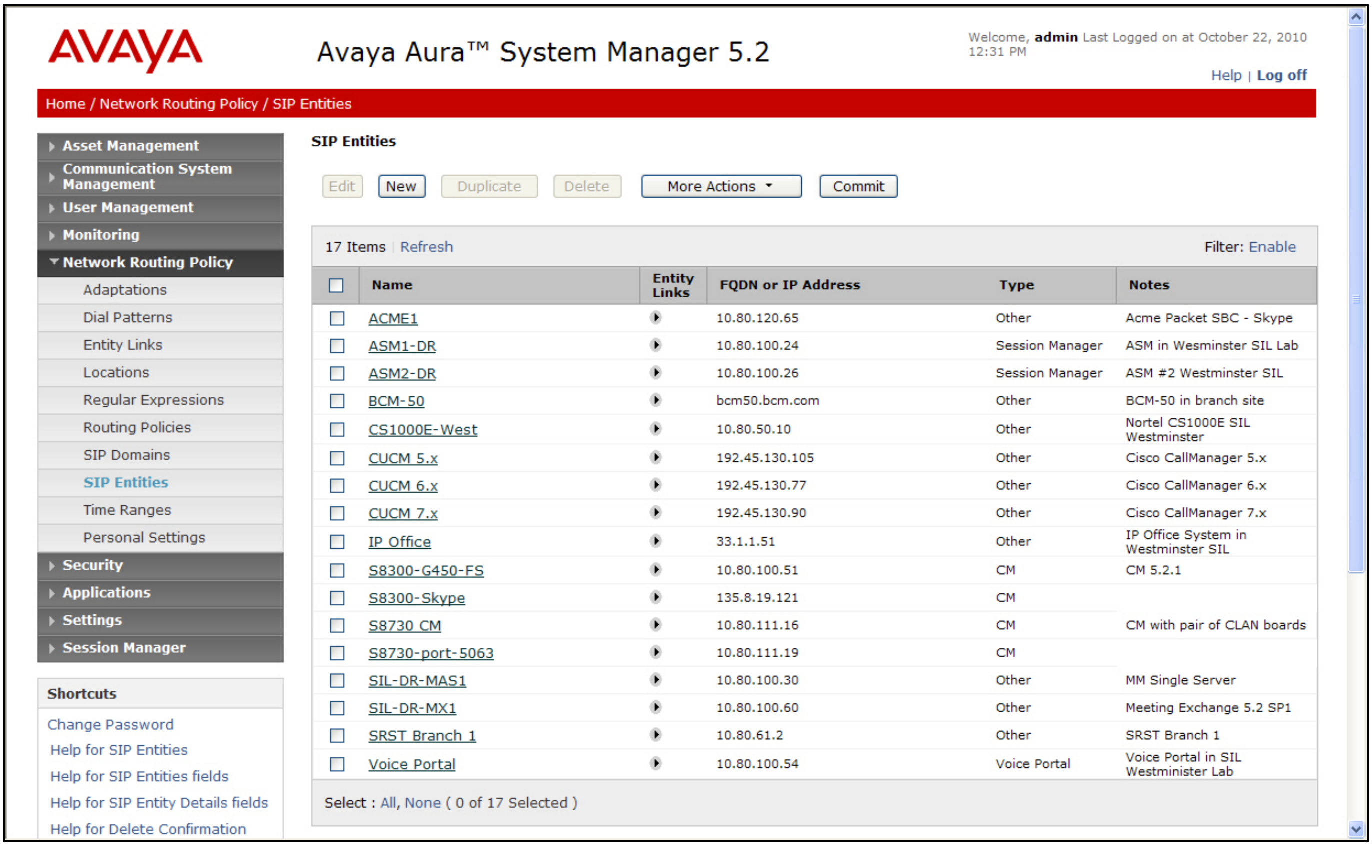

Figure 15: SIP Entities List

4.3.5 Entity Links

Entity Links defined the connections between the SIP Entities and Avaya Aura® Session Manager. In the reference configuration, Entity Links are defined between Avaya Aura® Session Manager and:

- Avaya Voice Portal (Voice Portal)
- Acme Packet SBC (ACME1)
- Avaya Aura® Communication Manager

[5] This SIP Entity and associated Entity Link is defined in detail as part of the base configuration documented in **Reference [1]**.

To add an Entity Link, select **Entity Links** on the left **Network Routing Policy** menu and click on the **New** button on the right. The screen shown in **Figure 16** is displayed.

1. Enter a descriptive name in the **Name** field.
2. In the **SIP Entity 1** drop down menu select the Avaya Aura® Session Manager SIP Entity created in **Section 4.3.4** (e.g. **ASM1-DR**).
3. In the **Port** field enter **5060**.
4. In the **SIP Entity 2** drop down menu select the **Voice Portal** SIP Entity created in **Section 4.3.4**.
5. In the **Port** field enter **5060**.
6. Check the **Trusted** box.
7. In the **Protocol** drop down menu select **TCP**[6].
8. Enter a description in the **Notes** field if desired (not shown).
9. Click on the **Commit** button.

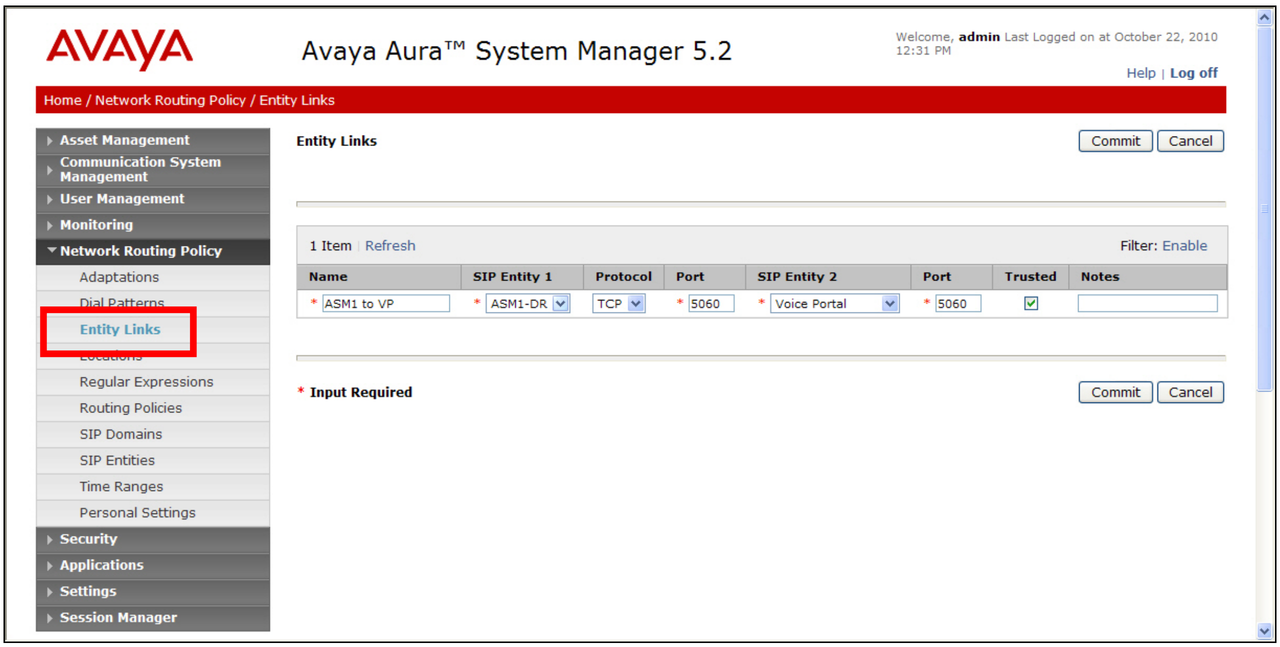

Figure 16: Entity Link – Voice Portal

[6] TCP protocol is used in the reference configuration.

10. Click on **New** and repeat steps 1 to 9 for the **ACME1** Entity Link, specifying **ACME1** in the **SIP Entity 2** drop down menu. Note that, in the reference configuration, port 5063 is used for the Entity Link between the Session Manager and the Acme Packet SBC.

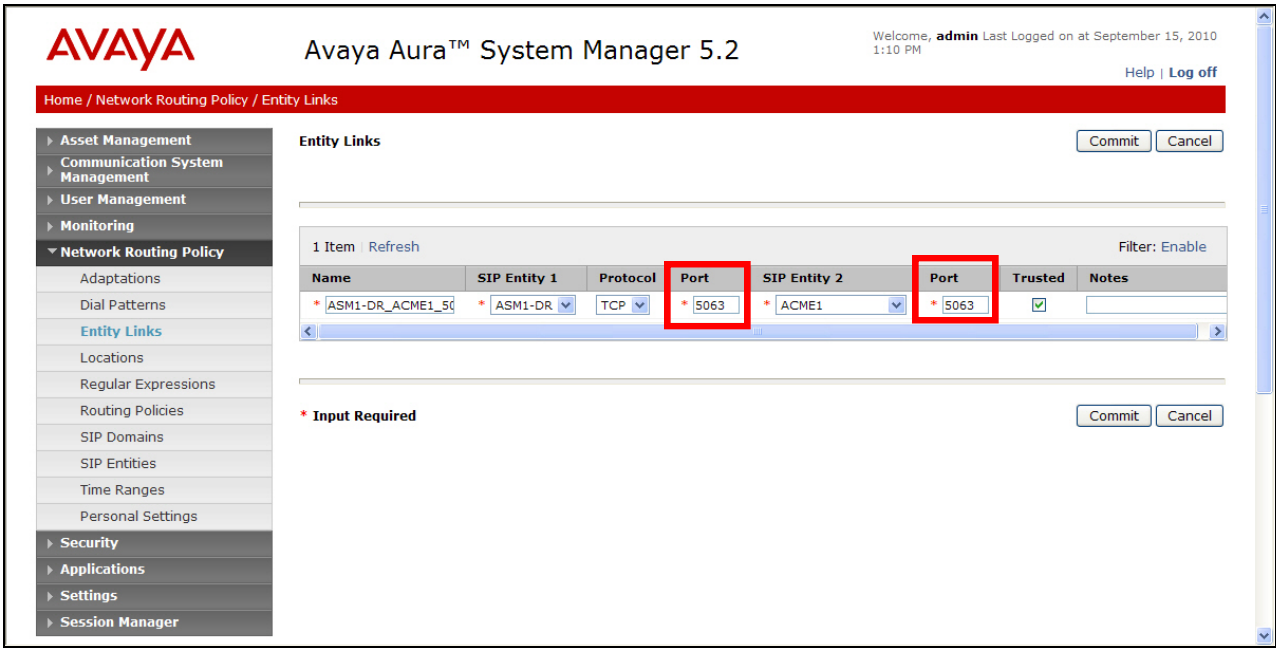

Figure 17: Entity Link – Acme Packet SBC

When completed, the Entity Links list will look like **Figure 18**.

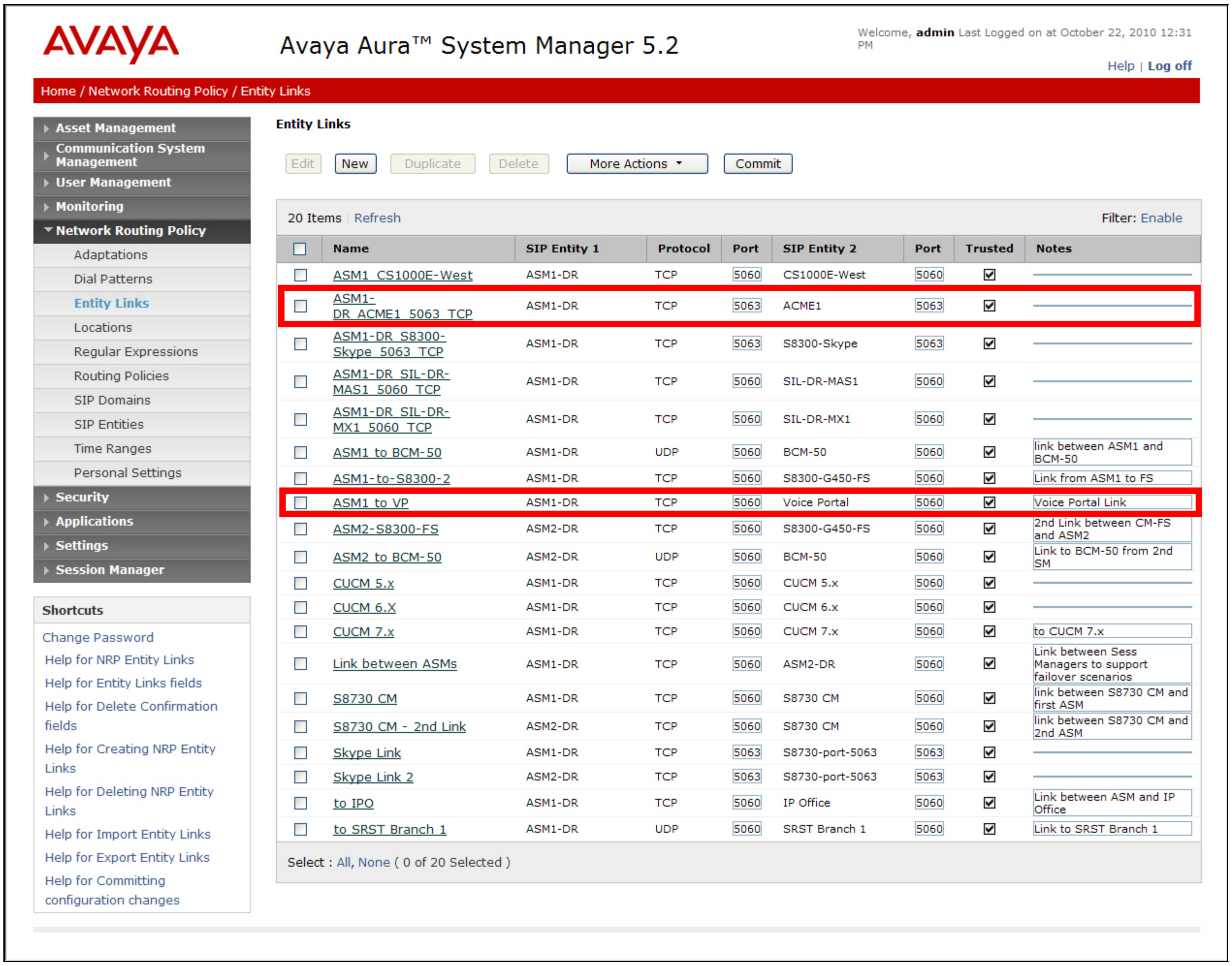

Figure 18: Entity Links List

4.3.6 Time Ranges

The Time Ranges form allows admission control criteria to be specified for Routing Policies (**Section 4.3.7**). In the reference configuration no restrictions were used.

To add a Time Range, select **Time Ranges** on the left **Network Routing Policy** menu and click on the **New** button on the right. The screen shown in **Figure 19** is displayed.

1. Enter a descriptive name in the **Name** field (e.g. **24/7**).
2. Check each day of the week.
3. In the **Start Time** field enter **00:00**.
4. In the **End Time** field enter **23:59**.
5. Enter a description in the **Notes** field if desired.
6. Click the **Commit** button.

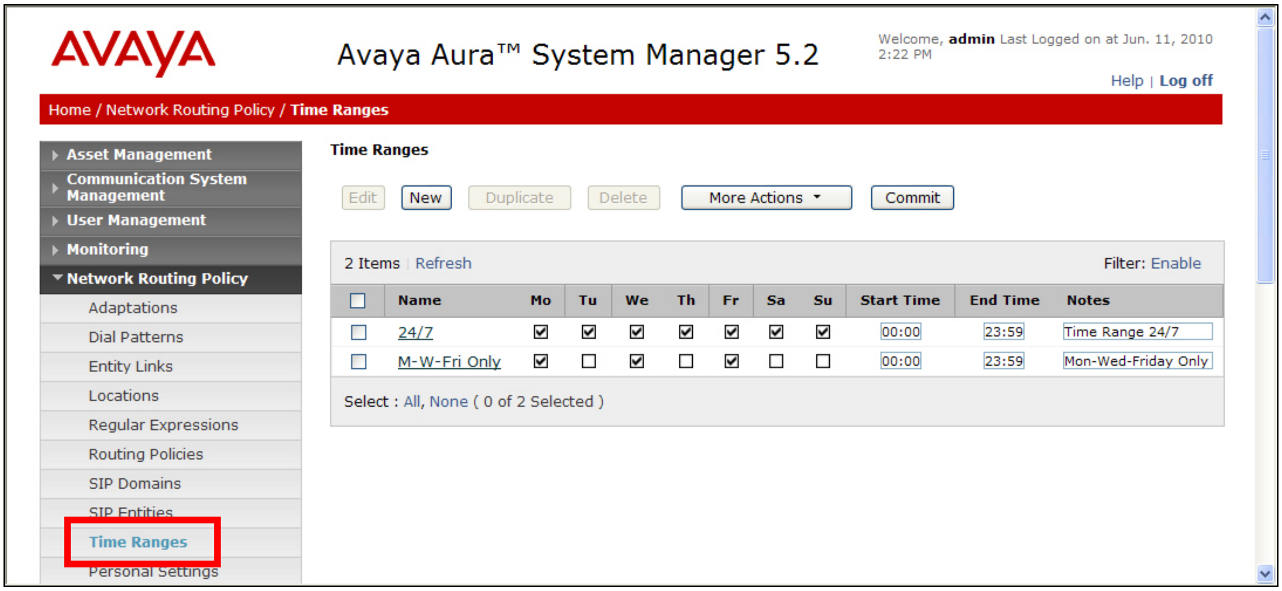

Figure 19: Time Ranges

4.3.7 Routing Policies

Routing Policies associate destination SIP Entities (**Section 4.3.4**) with Time of Day admission control parameters (**Section 4.3.6**) and Dial Patterns (**Section 4.3.8**). In the reference configuration Routing Policies are defined for:

- Inbound calls to SIP Entity **Voice Portal** (to Avaya Voice Portal)
- Transferred calls to SIP Entity **S8730-port-5063** (transferred calls to Communication Manager ACD)

Note – In the reference configuration the **Regular Expressions** parameters are not used.

Name	SIP Entity Destination	Time Of Day	Dial Pattern(s)	Notes
to_Voice Portal	Voice Portal	24/7	13038006247	Any call to this dial pattern will route to Avaya Voice Portal and use port 5060.
to_S8730_5063[7]	S8730-port-5063	24/7	6670201	All matching dial patterns will route to Communication Manager ACD via VDN 6670201.

Table 3: Routing Policy Provisioning

To add a Routing Policy, select **Routing Policies** on the left **Network Routing Policy** menu and click on the **New** button on the right. The window shown in **Figure 20** will open.

[7] This Routing Policy is defined in detail in **Reference [1]**.

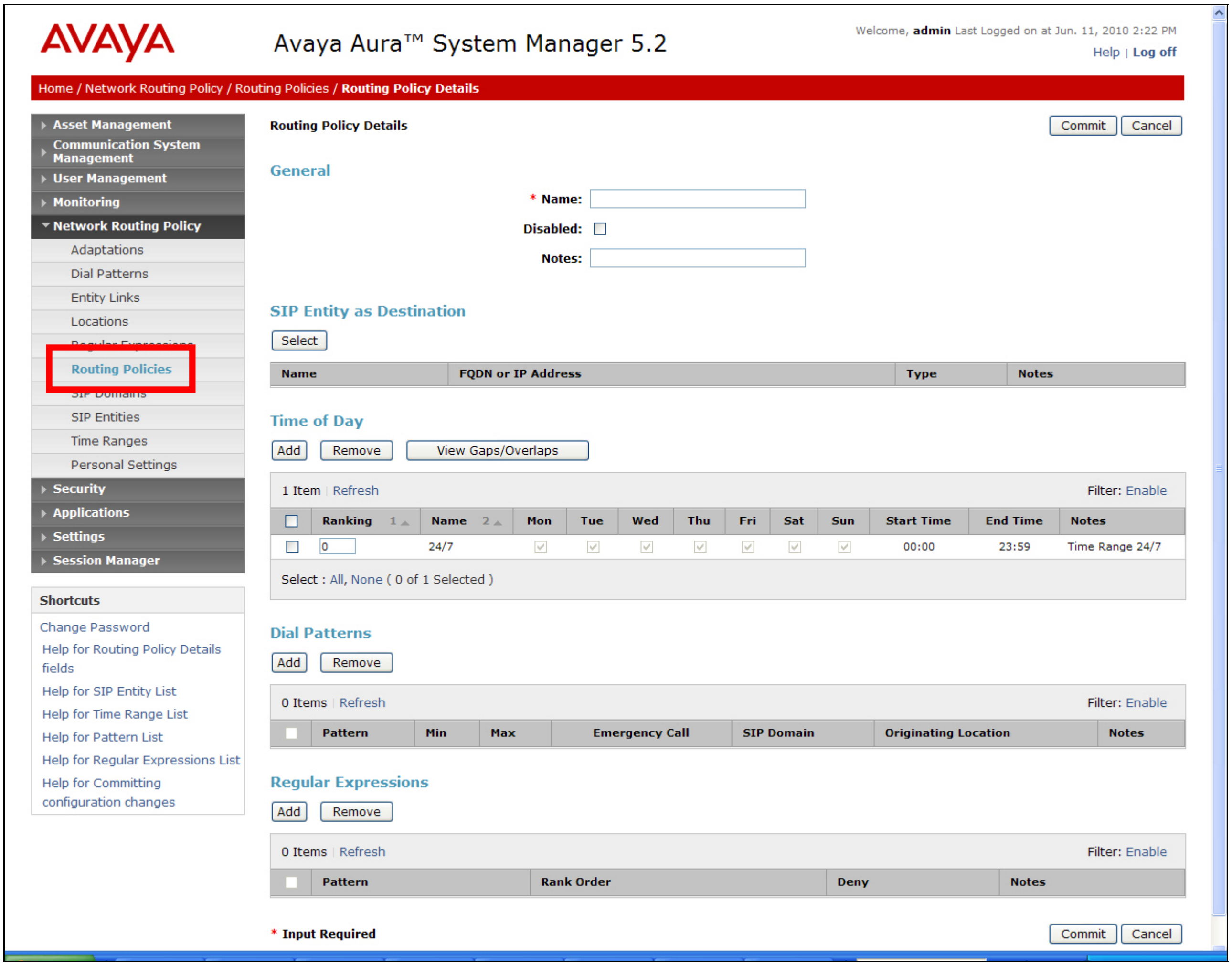

Figure 20: Routing Policy Details

1. **General** section
   a. Enter a descriptive name in the **Name** field (e.g. **to_Voice Portal**).
   b. Enter a description in the **Notes** field if desired.
2. **SIP Entity as Destination** section
   a. Click the **Select** button.
   b. Select the SIP Entity that will be the destination for this call (e.g. **Voice Portal**)
   c. Click the **Select** button and return to the Routing Policy Details form.
3. **Time of Day** section
   a. Click the **Add** button and select the **Time Range** for this Routing Policy.
   b. Click on **Select** and return to the Routing Policy Details form.

Note – Multiple time ranges may be selected and a Ranking value applied (0 is the highest).

4. **Dial Pattern** section
   a. Click the **Add** button and select the **Dial Pattern** for this Routing Policy.
   b. Click on **Select** and return to the Routing Policy Details form. The form will look like **Figure 21**.

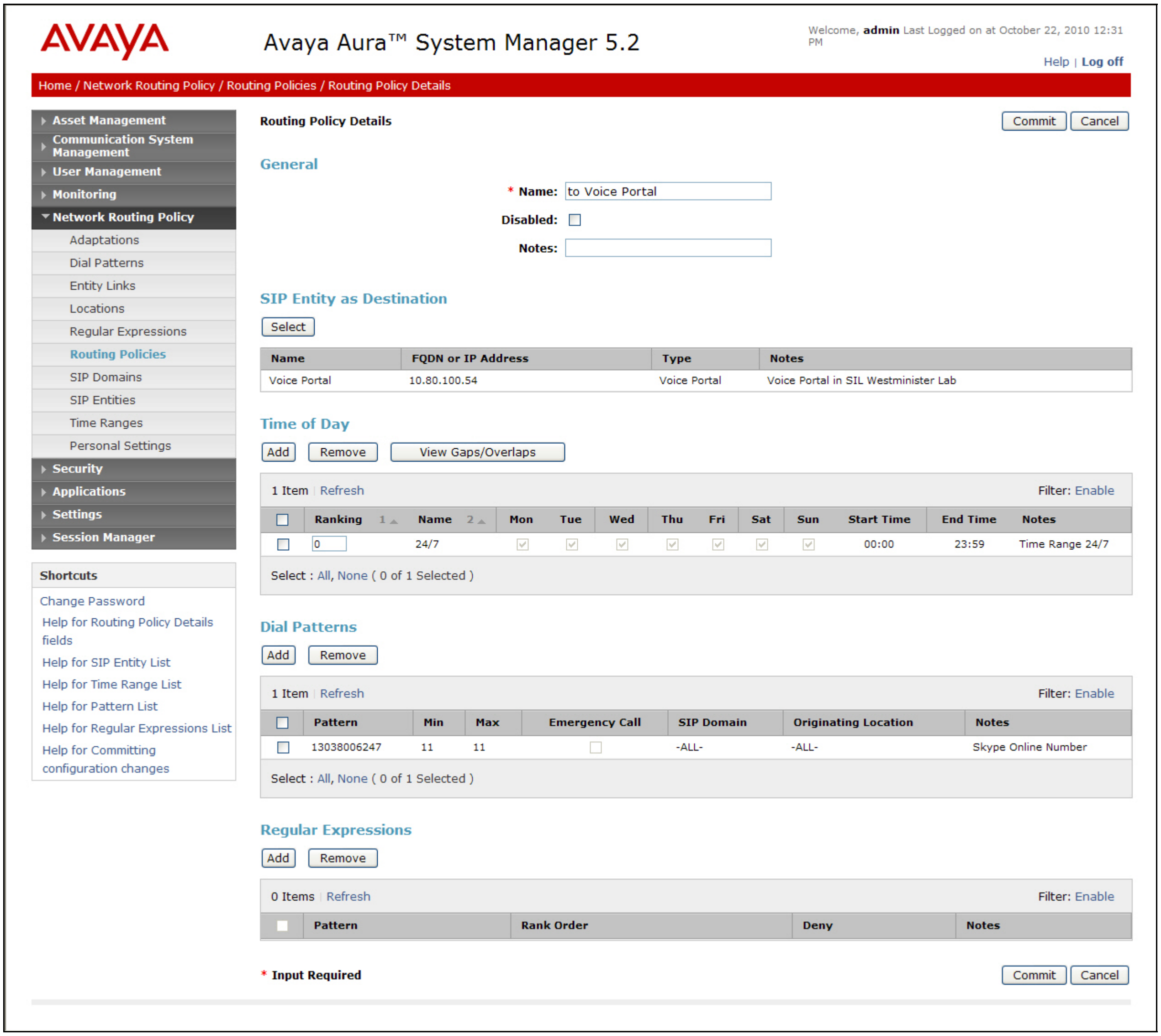

Figure 21: Routing Policy Details - Completed

5. Click the **Commit** button.
6. Repeat steps 1 to 5 for each Routing Policy. When completed the form will look like **Figure 22**. The routing policies relevant to the reference configuration are listed in **Table 3**.
7. Click the **Commit** button.

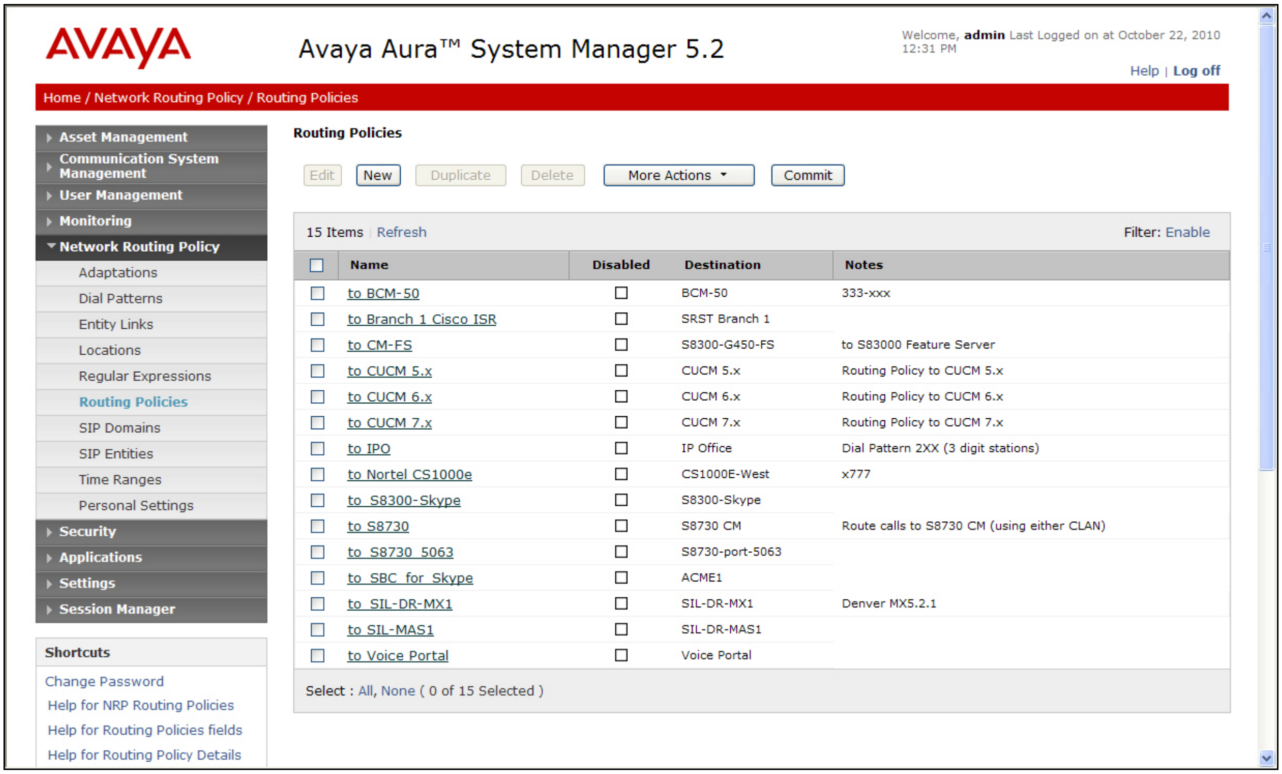

Figure 22: Routing Policies List

4.3.8 Dial Patterns

Dial Patterns define digit strings to be matched for inbound and outbound calls. In addition, the FQDN in the request URI is also examined.

Note – The Dial Pattern digit string with the most complete match will be selected. As an example, if a 7 digit string with matching pattern 667 is defined first in the list, and a 7 digit string with matching pattern 6675961 is defined last, a call for 6675961 will match on the 6675961 pattern.

The following Dial Patterns were provisioned in the reference configuration.

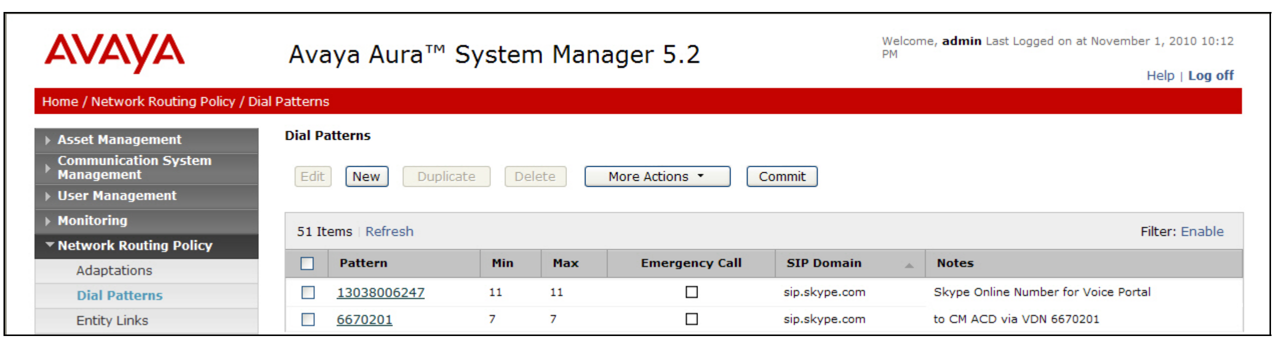

Figure 23: Completed Dial Patterns

To add a Dial Pattern, select **Dial Patterns** on the left **Network Routing Policy** menu and click on the **New** button on the right. The screen shown in **Figure 24** is displayed. This example would match a SIP INVITE with a Request URI to 13038006247 and sent by *sip.skype.com* (this would be an inbound call from Avaya Aura® Session Manager to Avaya Voice Portal).

1. **General** Section
   a. Enter a unique pattern in the **Pattern** field (e.g. **13038006247**).
   b. In the **Min** column enter the minimum number of digits (e.g. **11**.
   c. In the **Max** column enter the maximum number of digits (e.g. **11**).
   d. In the **SIP Domain** field drop down menu select the FQDN that will be contained in the Request URI *received* by Avaya Aura® Session Manager from Avaya Voice Portal (see **Section 3.1**).
   e. Enter a description in the **Notes** field if desired.

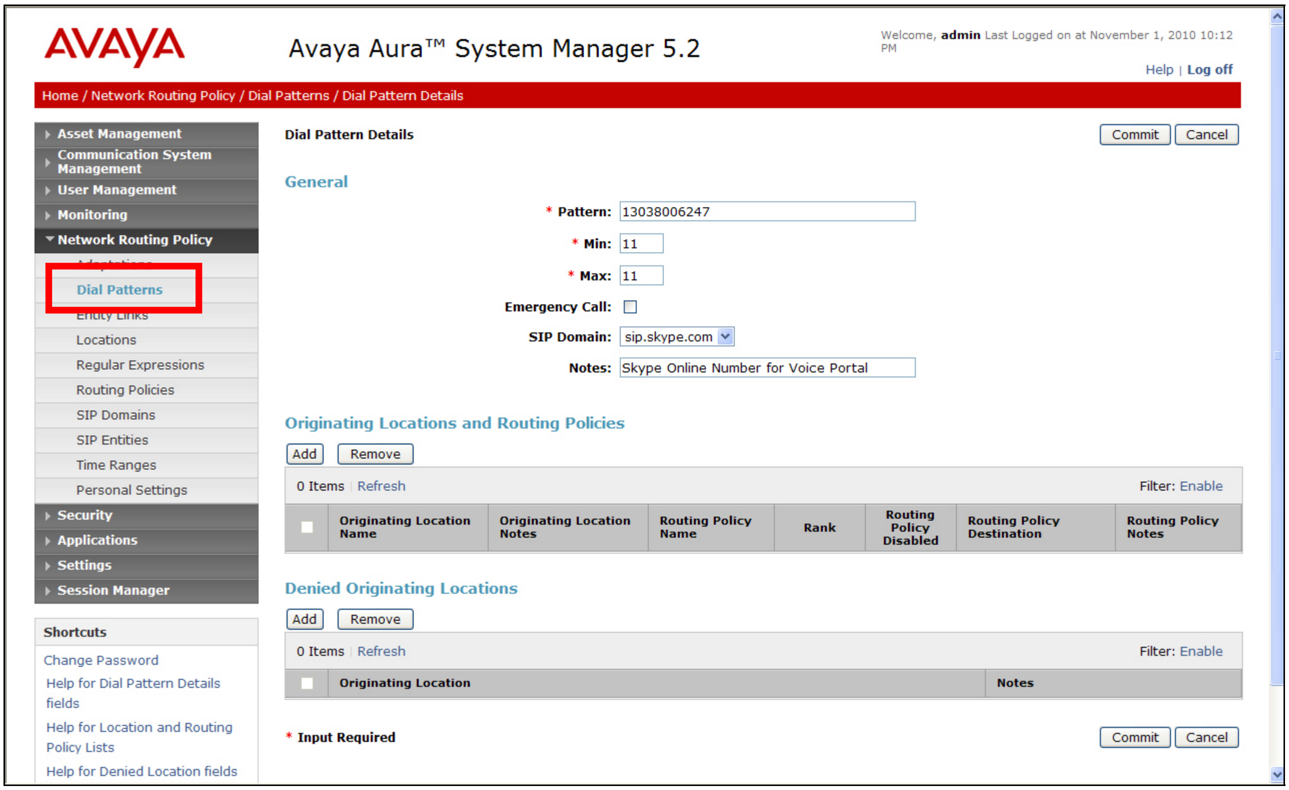

Figure 24: Dial Pattern Details - General

2. **Originating Locations and Routing Policies** Section
   a. Click on the Add button and the window in **Figure 25** will open.
   b. Click on the boxes for the appropriate Originating Locations (see **Section 4.3.3**), and Routing Policies (see **Section 4.3.7**) that pertain to this Dial Pattern.
      i. Location **AvayaCPE**
      ii. Routing Policy **to_Voice Portal** (Voice Portal).
   c. Click on the **Select** button and return to the Dial Pattern window.

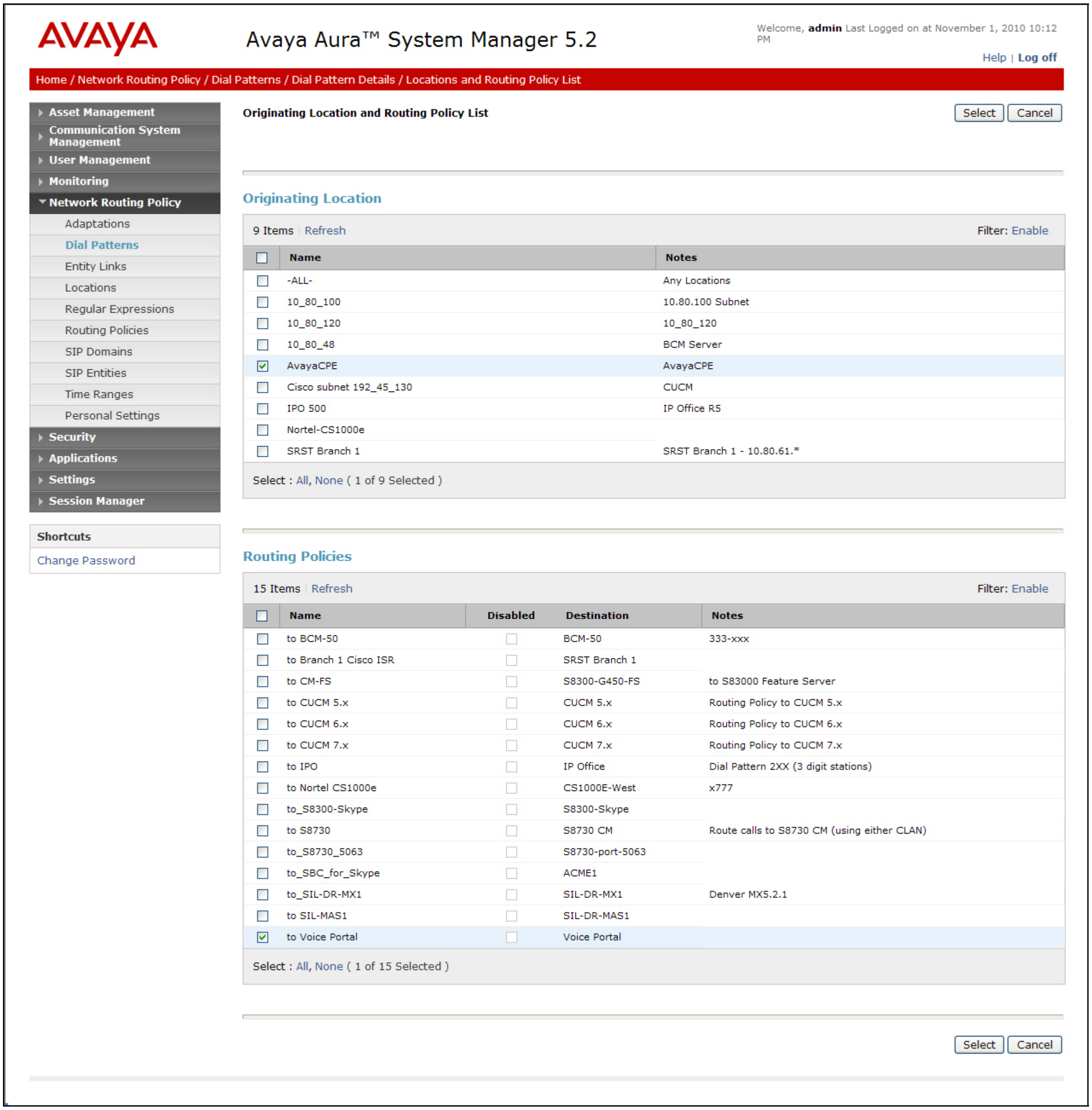

Figure 25: Dial Pattern Details – Originating Locations and Routing Policies

In the reference configuration, a SIP INVITE with a Request URI of *13038006247@sip.skype.com* would match and be sent to Voice Portal.

3. Click the **Commit** button
4. Repeat steps 1 to 3 for the remaining Dial Patterns listed in **Table 3**. The completed Dial Pattern screen will look like **Figure 23**.

4.4. Avaya Aura® Session Manager

To complete the Avaya Aura® Session Manager configuration, add an Avaya Aura® Session Manager instance. Note that this step is part of standard product installation and provisioning and may have already been performed. To add an Avaya Aura® Session Manager, select **Session Manager Administration** on the left **Session Manager** menu and click on the **New** button. The screen shown in **Figure 26** is part of the **Edit Session Manager** screen and contains the same fields as the **Add Session Manager** screen.

1. **General** section
    a. Select the **SIP Entity Name** field (e.g. **ASM1-DR**).
    b. Enter an optional description in the **Description** field.
    c. In the **Management Access Point Host Name/IP** field enter the IP address of the management interface of the Avaya Aura® Session Manager server. (e.g. **10.80.100.23**).
2. **Security Module** section
    a. Enter the **Network Mask** (e.g. **255.255.255.0**)
    b. Enter the **Default Gateway** (e.g. **10.80.100.1**)
    c. In the **Speed & Duplex** drop down menu verify **Auto** is selected (default).
3. Use all other default parameters.
4. Click the **Save** button and the completed form shown in **Figure 27** will be displayed.

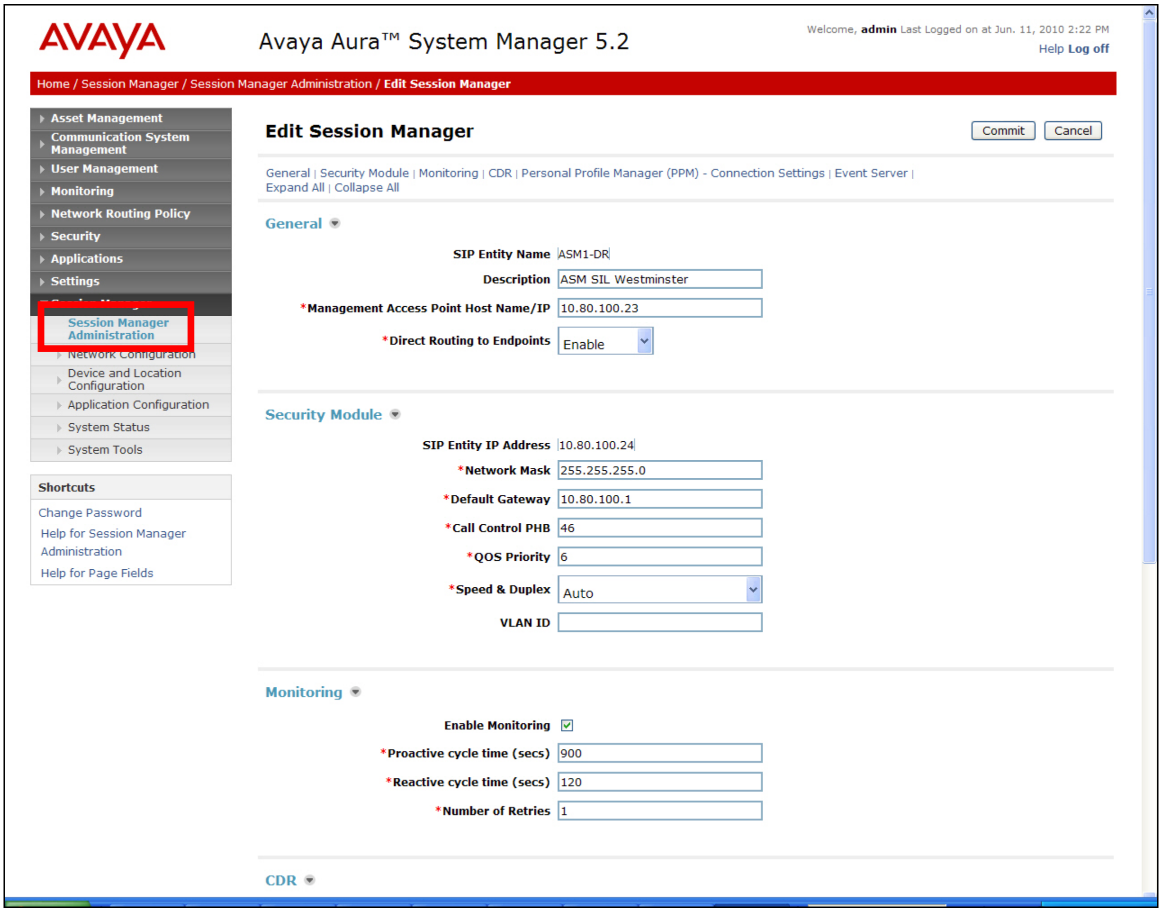

Figure 26: Edit Session Manager

Note – The SIP Entity IP address (under the Security Module heading) is automatically populated with the IP address defined for the Avaya Aura® Session Manager SIP Entity (**ASM1-DR**) in **Section 4.3.4**.

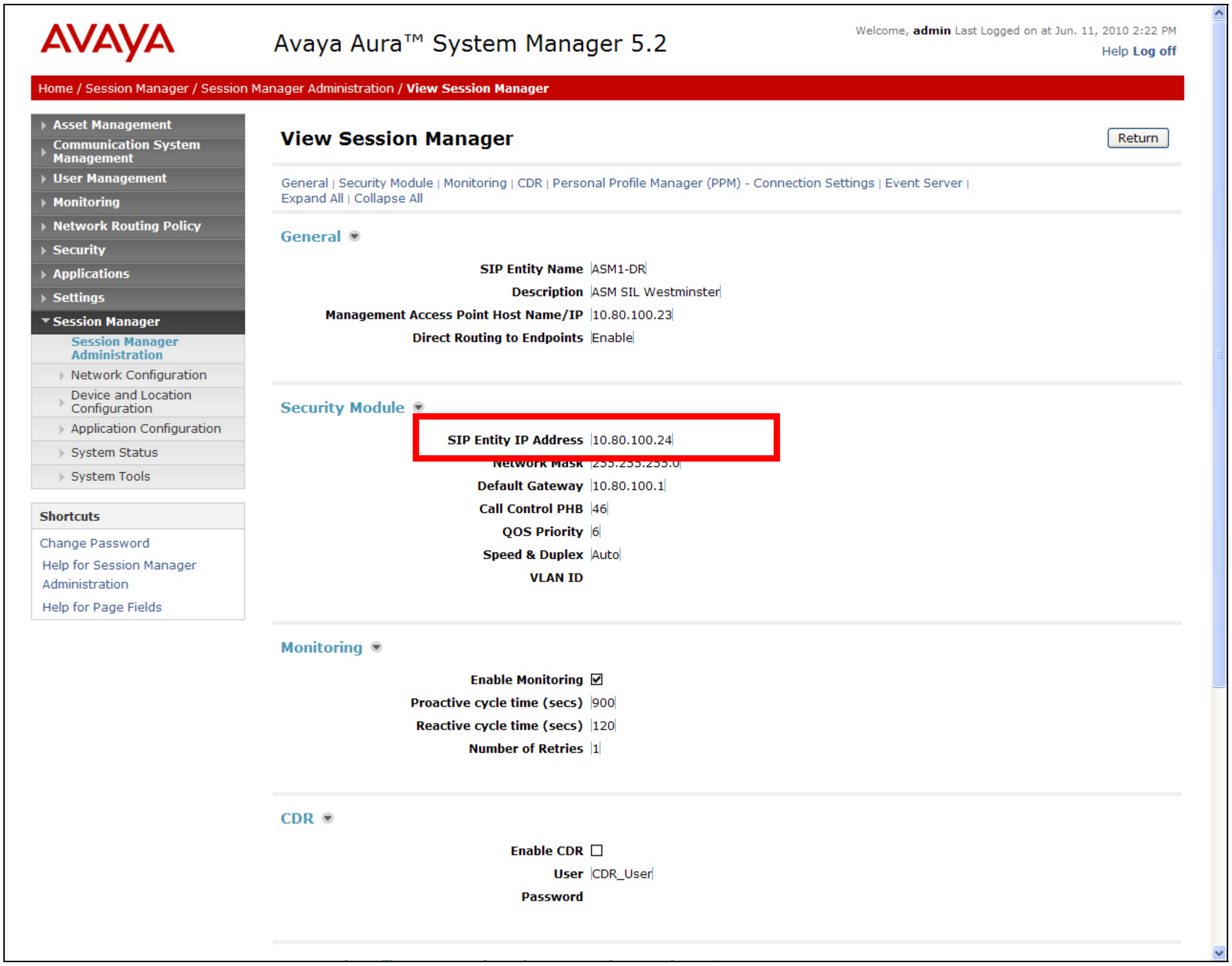

Figure 27: Completed Session Manager Form

5. Acme Packet Net-Net 3800

Reference [1] contains the complete description and programming used for the Acme Packet Net-Net 3800 in the implementation of Avaya SIP trunk architecture with Skype Connect. **Appendix A** of **Reference [1]** also contains the complete configuration of the Acme Packet Net-Net 3800.

5.1. Acme Packet Service States

In the reference configuration, the Acme Packet SBC requests and provides service state by sending out and responding to, SIP *OPTIONS* messages. Acme Packet sends the OPTIONS message with the hop count (SIP Max-Forwards) set to zero.

- Acme/Avaya Aura® Session Manager
  - Acme Packet sends OPTIONS → Avaya Aura® Session Manager responds with 200 OK
  - Avaya Aura® Session Manager sends OPTIONS → Acme Packet responds with 404 Not Found which is accepted by Session Manager as a valid "Up" Link Status response
- Acme/Skype Connect
  - Acme Packet to Skype Connect > OPTIONS messages are disabled.
  - Skype Connect does not send SIP OPTIONS messages.

5.2. Acme Packet Network Interfaces

Figure 28 shows the Acme Packet network interface connections used in the reference configuration. The physical and network interface provisioning for the "EXTERNAL" (to Skype Connect) and "INTERNAL" (to Avaya CPE) interfaces is described in **Sections 5.3.3** and **5.3.4** of **Reference [1].**

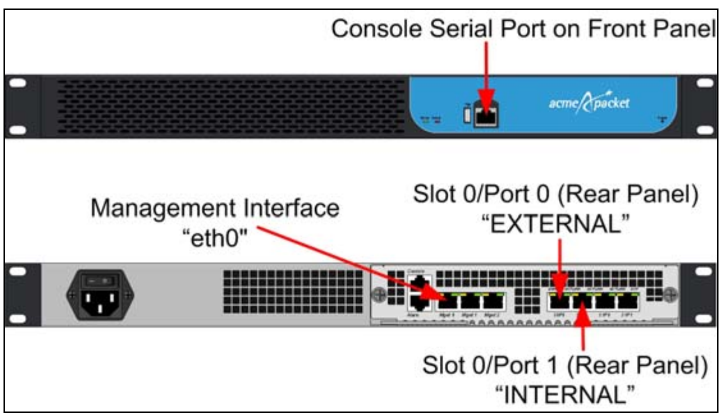

Figure 28: Acme Packet Network Interfaces

5.3. Acme Packet Provisioning

Note – Only the Acme Packet provisioning required for the reference configuration is described in these Application Notes. For more information on Acme Packet configuration see **References [14-16]**.

Note – The following Sections describe only that additional provisioning required on the Acme Packet SBC to support call transfers from Voice Portal using SIP REFER. See **Reference [1]** for complete details of programming the Acme Packet SBC in accordance with the Avaya SIP trunk architecture.

The Acme Packet SBC was configured using the Acme Packet CLI via a serial console port connection. An IP remote connection to a management port is also supported. The following are the generic steps for configuring various elements.

1. Log in with the appropriate credentials.
2. Enable the Superuser mode by entering **enable** command and the appropriate password (prompt will end with #).
3. In Superuser mode, type **configure terminal** and press <ENTER>. The prompt will change to (*configure)#*.
4. Type the name of the element that will be configured (e.g., **session-router**).
5. Type the name of the sub-element, if any (e.g., **session-agent**).
6. Type the name of the parameter followed by its value (e.g., **ip-address**).
7. Type **done**.
8. Type **exit** to return to the previous menu.
9. Repeat steps 4-8 to configure all the elements. When finished, exit from the configuration mode by typing **exit** until returned to the Superuser prompt.
10. Type **save-configuration** to save the configuration.
11. Type **activate-configuration** to activate the configuration.

Once the provisioning is complete, the configuration may be verified by entering the ***show running-config*** command.

5.3.1 SIP REFER Method Call Transfer

The Net-Net SBC has a configuration parameter giving it the ability to provision the handling of REFER methods as call transfers. This parameter is called **refer-call-transfer**. See **Reference [13]** for more information. Also, see **Appendix A** for a screenshot of the configuration parameter as shown on the Acme Packet SBC.

1. Enter **session-router → session-agent**
2. Enter **select → 10.80.100.24**
3. Enter **refer-call-transfer → enabled**
4. Enter **done**
5. Enter **exit**
6. Enter **exit**
7. Enter **exit**

5.3.2 Header Manipulation for UUI from Voice Portal to Communication Manager

A two-step header manipulation rule was defined on the Acme Packet SBC at the session agent associated with the Avaya Aura® Session Manager in order to pass User-to-User information from

Voice Portal to Communication Manager. The first step copies the User-to-User tag contained in the Refer-to header in the SIP REFER message received from Voice Portal and makes the data part of the existing uri-user data. The second step edits the SIP INVITE generated by the SBC and deletes the appended UUI data from the following: Request-URI, To, and Route headers. Finally, the manipulation adds a User-to-User header with the UUI data. See **Appendix A** for a screenshot of the manipulation rules as shown on the Acme Packet SBC.

5.3.2.1 Avaya-incoming

The existing **Avaya-incoming** manipulation defined in **Reference [1]** is modified with the following commands.

1. Enter **session-router → sip-manipulation**
2. Enter **select → Avaya-incoming**
3. Enter **header-rules**
4. Enter **name → requri**
5. Enter **header-name -> Refer-To**
6. Enter **action → manipulate**
7. Enter **comparison-type → case-sensitive**
8. Enter **msg-type → request**
9. Enter **methods → REFER**
10. Enter **element-rule**
11. Enter **name → getUUI**
12. Enter **parameter-name → User-to-User**
13. Enter **type → uri-header**
14. Enter **action → store**
15. Enter **match-val-type → any**
16. Enter **comparison-type → case-sensitive**
17. Enter **done**
18. Enter **name → appenduriuser**
19. Enter **type → uri-user**
20. Enter **action → replace**
21. Enter **match-val-type → any**
22. Enter **comparison-type → boolean**
23. Enter **match-value → $requri.$getUUI.$0**
24. Enter **new-value → $ORIGINAL+UUI+$requri.$getUUI.$0**
25. Enter **exit**
26. Enter **exit**
27. Enter **y** when prompted to save changes
28. Enter **exit**
29. Enter **y** when prompted to save changes
30. Enter **exit**
31. Enter **exit**

5.3.2.2 AVP-manip-out

A new manipulation called **AVP-manip-out** is defined with the following steps.

1. Enter **session-router → sip-manipulation**
2. Enter **name → AVP-manip-out**
3. Enter **header-rules**
4. Enter **name → get_UUI**
5. Enter **header-name → Request-URI**
6. Enter **action → manipulate**
7. Enter **comparison-type → case-sensitive**
8. Enter **msg-type → request**
9. Enter **methods → INVITE**
10. Enter **element-rule**
11. Enter **name → store_UUI**
12. Enter **type → uri-user**
13. Enter **action → store**
14. Enter **match-val-type → any**
15. Enter **comparison-type → case-sensitive**
16. Enter **match-value → (.*)(UUI)(.*)**
17. Enter **done**
18. Enter **name → get_UUI**
19. Enter **type → uri-user**
20. Enter **action → find-replace-all**
21. Enter **match-val-type → any**
22. Enter **comparison-type → case-sensitive**
23. Enter **match-value → (.*)(UUI)(.*)**
24. Enter **new-value → $get_UUI.$store_UUI.$1**
25. Enter **done**
26. Enter **exit**
27. Enter **done**
28. Enter **name → get_UUI_To**
29. Enter **header-name → To**
30. Enter **action → manipulate**
31. Enter **comparison-type → case-sensitive**
32. Enter **msg-type → request**
33. Enter **methods → INVITE**
34. Enter **element-rule**
35. Enter **name → store_UUI**
36. Enter **type → uri-user**
37. Enter **action → store**
38. Enter **match-val-type → any**
39. Enter **comparison-type → case-sensitive**
40. Enter **match-value → (.*)(UUI)(.*)**
41. Enter **done**
42. Enter **name → get_UUI**
43. Enter **type → uri-user**
44. Enter **action → find-replace-all**
45. Enter **match-val-type → any**
46. Enter **comparison-type → case-sensitive**

47. Enter **match-value → (.*)(UUI)(.*)**
48. Enter **new-value → $get_UUI.$store_UUI.$1**
49. Enter **done**
50. Enter **exit**
51. Enter **done**
52. Enter **name → get_UUI_Route**
53. Enter **header-name → Route**
54. Enter **action → manipulate**
55. Enter **comparison-type → case-sensitive**
56. Enter **msg-type → request**
57. Enter **methods → INVITE**
58. Enter **element-rule**
59. Enter **name → store_UUI**
60. Enter **type → uri-user**
61. Enter **action → store**
62. Enter **match-val-type → any**
63. Enter **comparison-type → case-sensitive**
64. Enter **match-value → (.*)(UUI)(.*)**
65. Enter **done**
66. Enter **name → get_UUI**
67. Enter **type → uri-user**
68. Enter **action → find-replace-all**
69. Enter **match-val-type → any**
70. Enter **comparison-type → case-sensitive**
71. Enter **match-value → (.*)(UUI)(.*)**
72. Enter **new-value → $get_UUI.$store_UUI.$1**
73. Enter **done**
74. Enter **exit**
75. Enter **done**
76. Enter **name → add_UUI**
77. Enter **header-name → User-to-User**
78. Enter **action → add**
79. Enter **comparison-type → boolean**
80. Enter **msg-type → request**
81. Enter **methods → INVITE**
82. Enter **match-value → $get_UUI.$store_UUI**
83. Enter **new-value → $get_UUI.$store_UUI.$3**
84. Enter **done**
85. Enter **exit**
86. Enter **exit**
87. Enter **y** when prompted to save changes
88. Enter **exit**
89. Enter **exit**

5.3.2.3 Assign Manipulations to Session Agent

The new manipulations need to be assigned to the Session Agent.

1. Enter **session-router → session-agent**
2. Enter **select → 10.80.100.24**
3. Enter **in-manipulationid → Avaya-incoming**
4. Enter **out-manipulationid → AVP-manip-out**
5. Enter **done**
6. Enter **exit**
7. Enter **exit**
8. Enter **exit**

6. Skype Connect

Information regarding the Skype Connect service offer can be found at http://www.skype.com.

6.1. Skype Manager

The Skype Connect service provisioning is performed using Skype Manager, a self-service, web-based provisioning tool. The procedures documented in **Reference [1]** can be followed to establish a Skype Connect profile and basic configuration of the Skype Connect service.

To access the Skype Manager, navigate to https://manager.skype.com and log in with the appropriate credentials. The **Sign In** screen is displayed as shown in **Figure 29**.

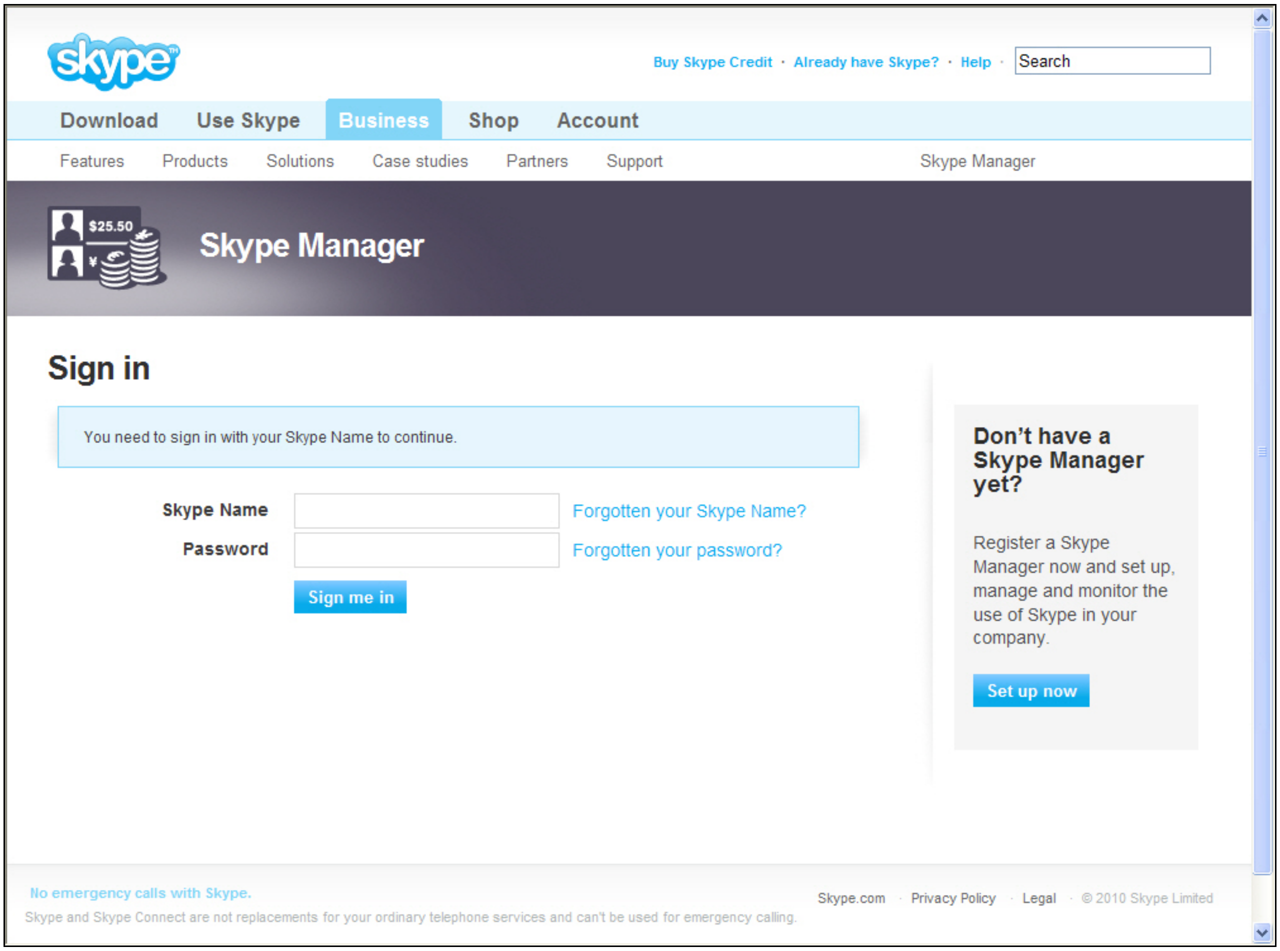

Figure 29: Skype Manager Sign In Screen

6.2. Skype Connect Profile

After logging in, the Dashboard screen is displayed as shown in **Figure 30**. Click on **Skype Connect.**

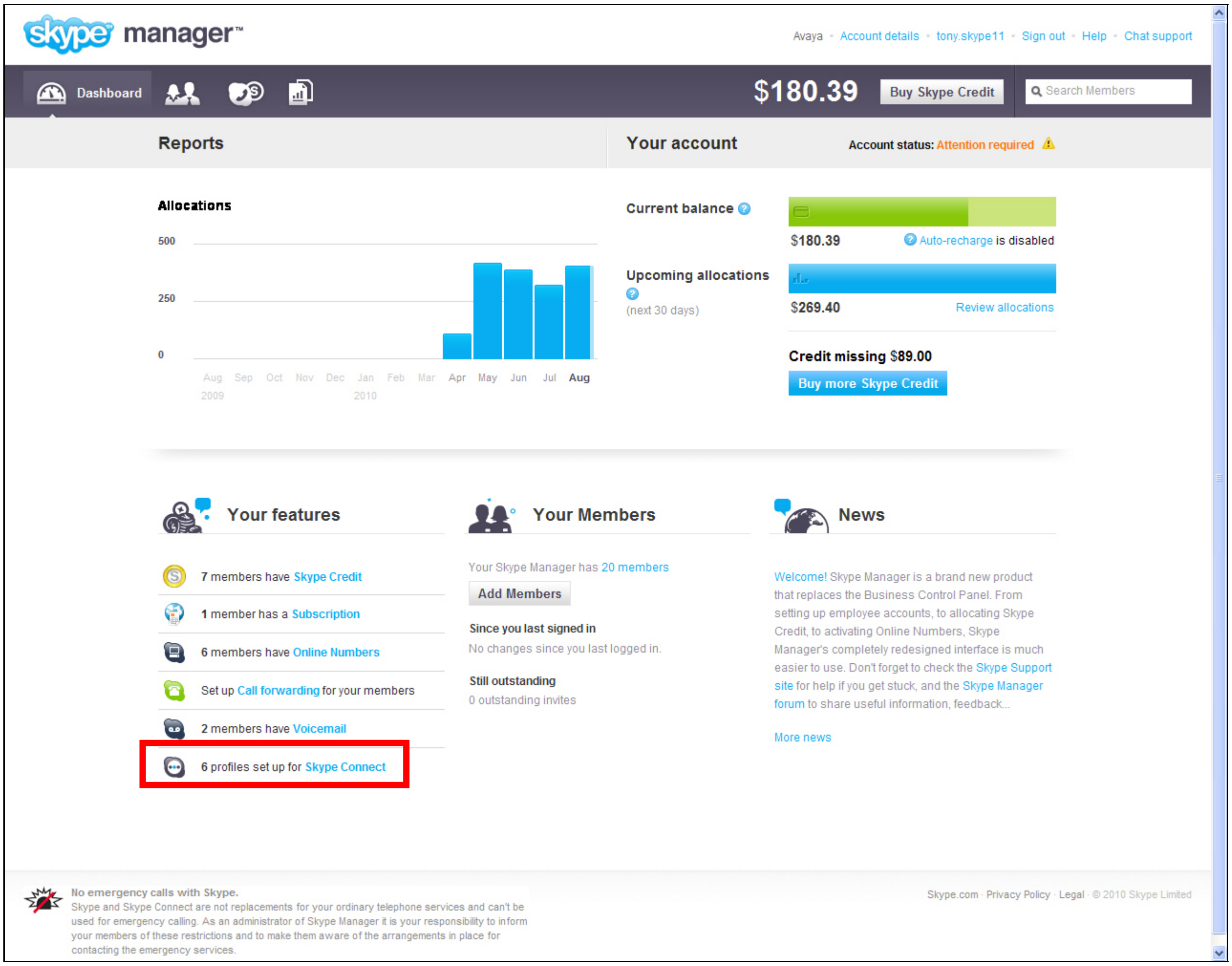

Figure 30: Skype Manager Dashboard Screen

The current Skype Connect profiles will be displayed (not shown). It is assumed that a Skype Connect profile has been configured per **Reference [1]**. Click on **View Profile** (not shown).

6.3. Incoming calls

Skype Online Numbers can be purchased from Skype. The process of subscribing to a Skype Online Number will assign it to the Skype Connect profile. When these Skype Online Numbers are dialed from the PSTN or from Skype Users, Skype will deliver the call to the Avaya CPE. These Skype Online Numbers are listed in the **Incoming calls** section of the Skype Connect profile. **Section 4.3.7** describes how Avaya Aura® Session Manager routes calls from Skype Connect to Voice Portal. As shown in **Figure 31,** a Skype Online Number of **13038006247** is associated with the profile.

6.3.1 Incoming calls – Skype Business Acount

Skype Connect enables a Business Account (Skype name) to be assigned to a SIP profile so other Skype users can make free calls to a SIP user's Skype name (Skype to Skype calls). Calls are routed from the Skype P2P network to the Skype Connect profile's User Agent. As shown in **Figure 31**, a Skype P2P call to "avayavoiceportal.avaya.com" is mapped to number **13038006247.** This is accomplished by entering **13038006247** in the **Extension number** field[8]**.** The dialed number **13038006247** is the destination number delivered in the Request URI of the SIP INVITE. These calls are delivered as inbound calls from Skype Connect to the Avaya CPE.

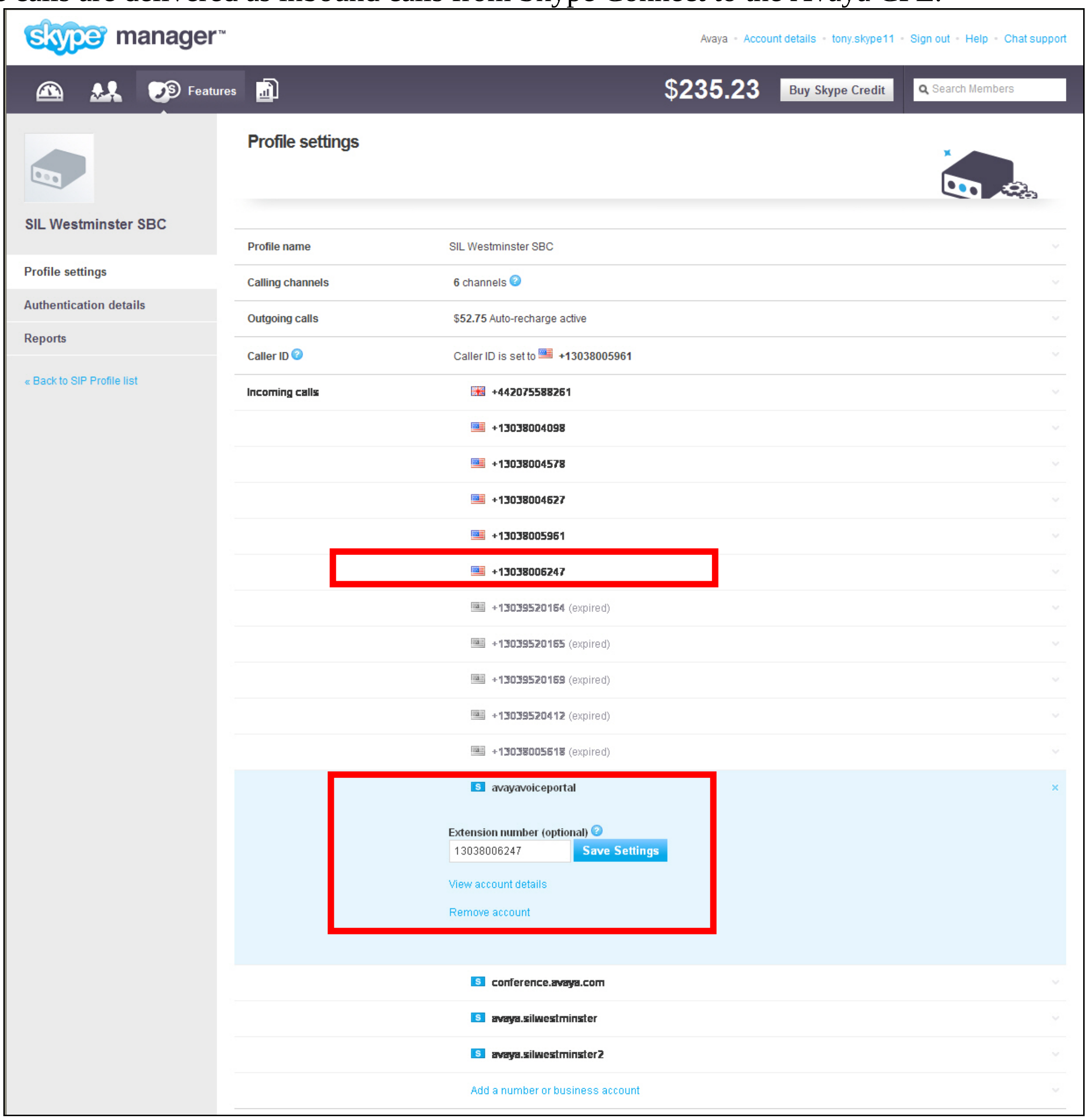

Figure 31: Skype Profile – Incoming Calls

[8] When no extension number is specified, Skype delivers the Skype-assigned SIP User name in the Request URI of the SIP Invite. In this case, additional Dial Patterns will be required to handle the Skype SIP User Name per **Section 4.3.8** and additional entries will be required in the Voice Portal Application table per **Section 3.3**.

7. Verification Steps

This section provides the verification steps that may be performed to verify basic operation of the Avaya Voice Portal with the Skype Connect service. Note that additional verification procedures are documented in **Reference [1]**.

7.1. Verify Avaya Voice Portal – System Monitor

From the Voice Portal web interface, select **System Monitor** under **Real-Time Monitoring**. For the MPP server, verify the Mode is **Online** and the State is **Running**.

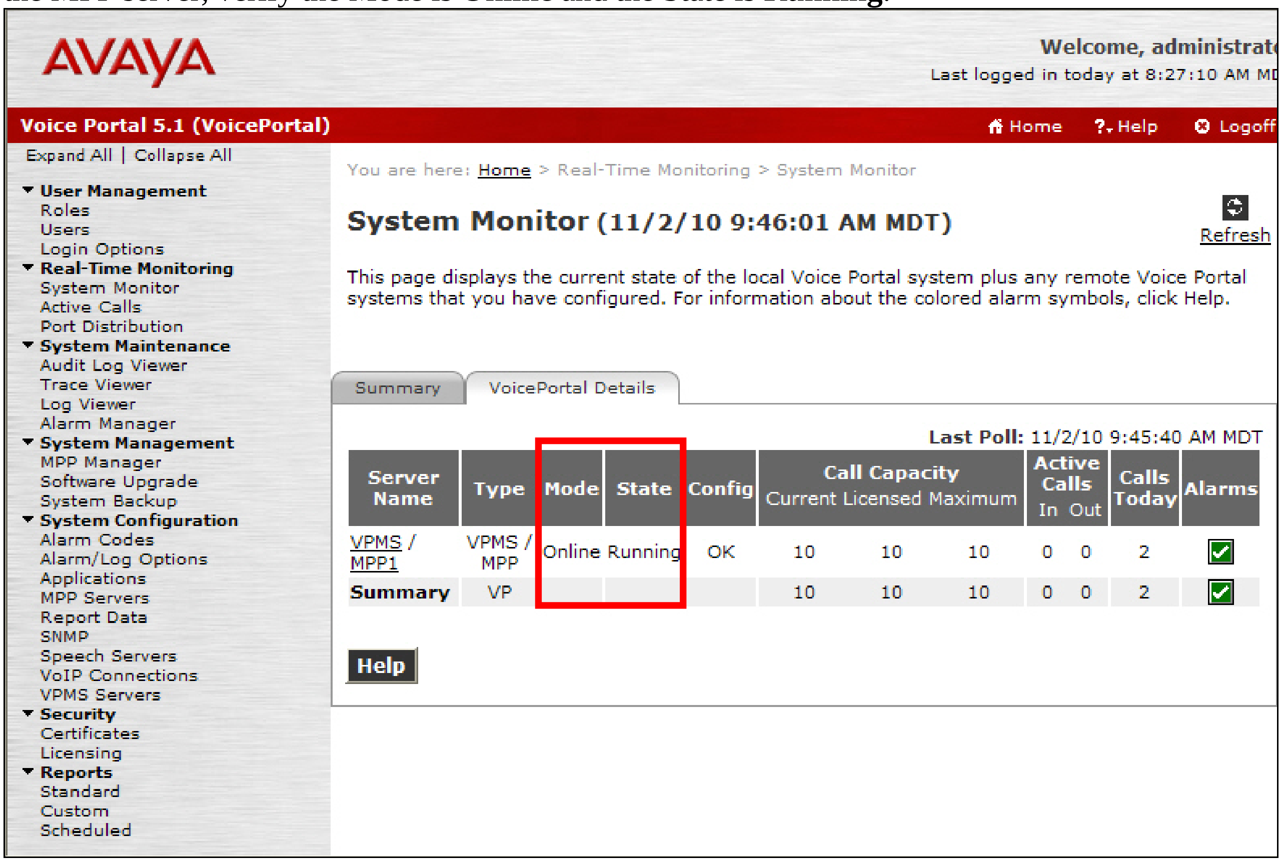

Figure 32: Voice Portal System Monitor

7.2. Verify Avaya Aura® Session Manager

Monitoring of Avaya Aura® Session Manager is performed via Avaya Aura® System Manager.

7.2.1 Verify SIP Entity Link Status

Expand the **Session Manager** menu and under **System Status**, click **SIP Entity Monitoring**. Verify that none of the links to the defined SIP entities assigned on Session Manager **ASM1-DR** are down (as indicated by **0/16** in **Figure 33)**, indicating that they are all reachable for call routing.

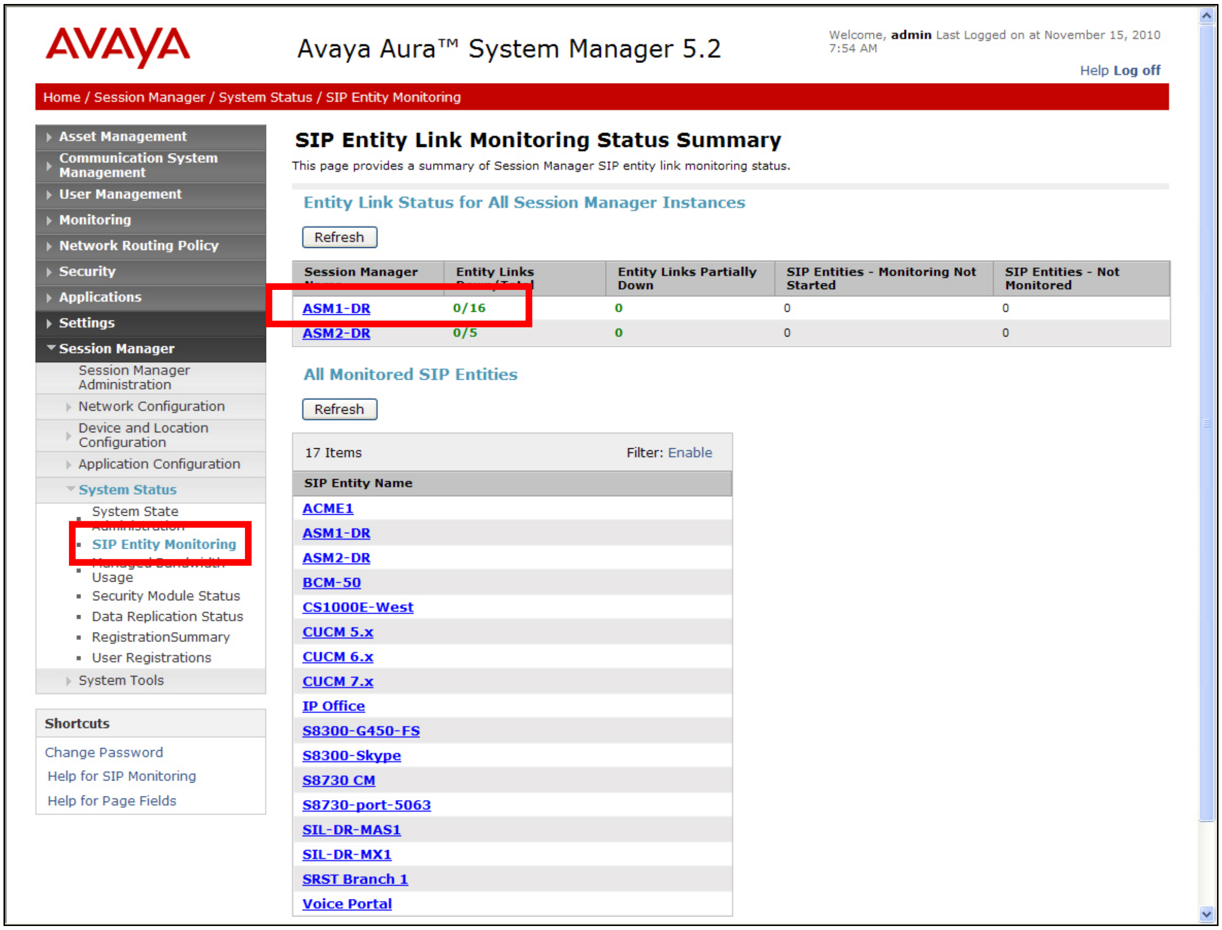

Figure 33: SIP Entity Link Monitoring - Summary

Selecting a monitored SIP Entity from the list will display its status (e.g. **Voice Portal**). **Figure 34** displays a **Conn. Status** of “Up”, a **Reason Code** of “200 OK”, and a **Link Status** of “Up” for SIP Entity **Voice Portal**.

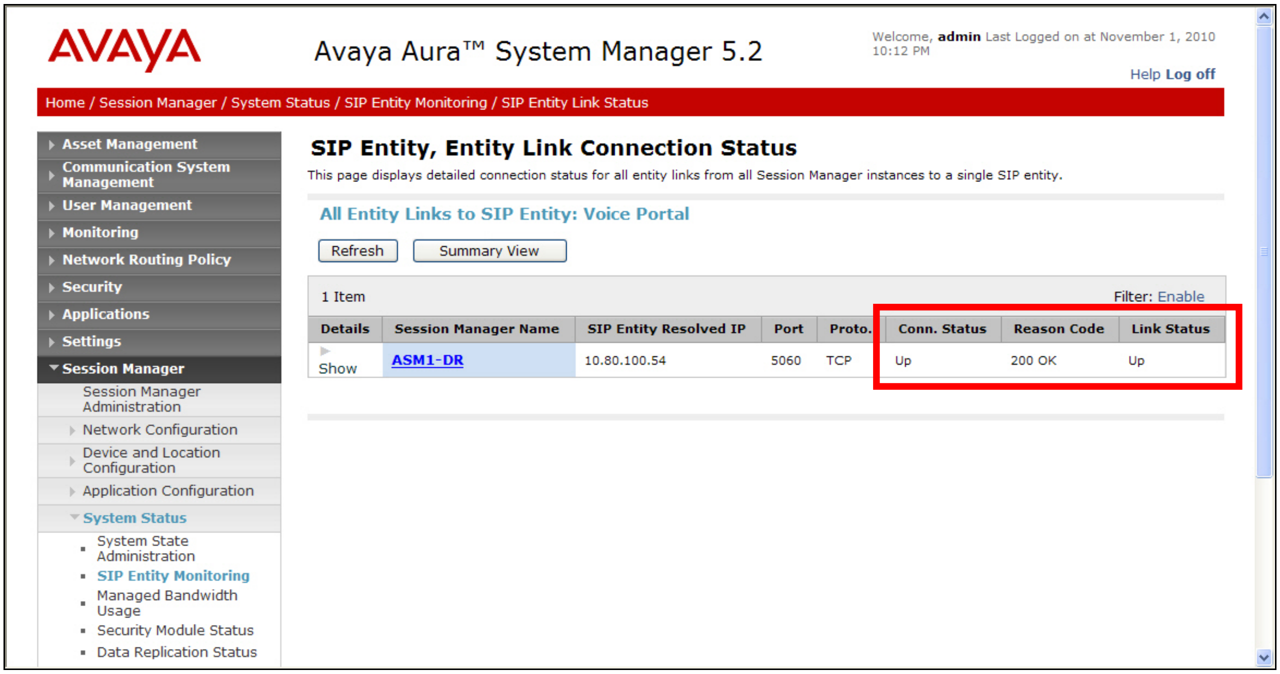

Figure 34: SIP Entity Link Connection Status

7.2.2 Verify System State

Expand the **Session Manager** menu and click **System State Administration**. Verify that the Management State is Management Enabled and the Service State is **Accept New Service** as shown in **Figure 35**.

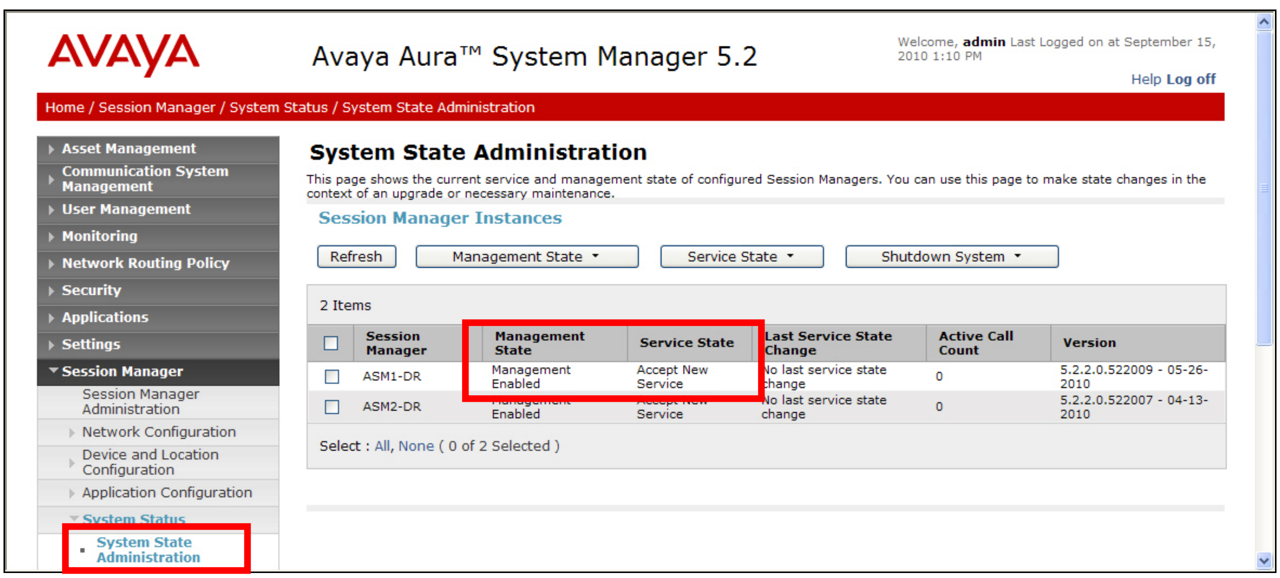

Figure 35: System State

7.2.3 Call Routing Test

The Call Routing Test verifies that the call routing/dial pattern for a particular source and destination is correctly provisioned. In this example a call from the PSTN to Skype Online Number 13038006247 is routed to the Avaya Voice Portal.

As shown in **Figure 36**, expand the **Session Manager** menu and, under **System Tools**, click **Call Routing Test**. Populate the fields as follows:

- **Called party URI – 13038006247@sip.skype.com** → This is the request URI sent by the Acme Packet SBC to Avaya Aura® Session Manager.
- **Calling Party URI – 13035381762@sip.skype.com**→ This is the contents of the Skype SIP INVITE From header.
- **Calling Party Address – 10.80.120.65** → This is the source IP address of the call (Acme Packet SBC).
- **Session Manager Listening Port – 5063**→ This is the port provisioned for Session Manager.
- **Day of the week –** Since no time restrictions were defined for the reference configuration (see **Section 4.3.6**) any day value may be selected.
- **Time –** Since no time restrictions were defined for the reference configuration (see **Section 4.3.6**) any time value may be selected.
- **Transport Protocol –** Select the transport protocol used (e.g., **TCP**).
- **Called Session Manager Instance –** Select the Session Manager used for the call. In the reference configuration only one Session Manager is defined (**ASM1-DR**).

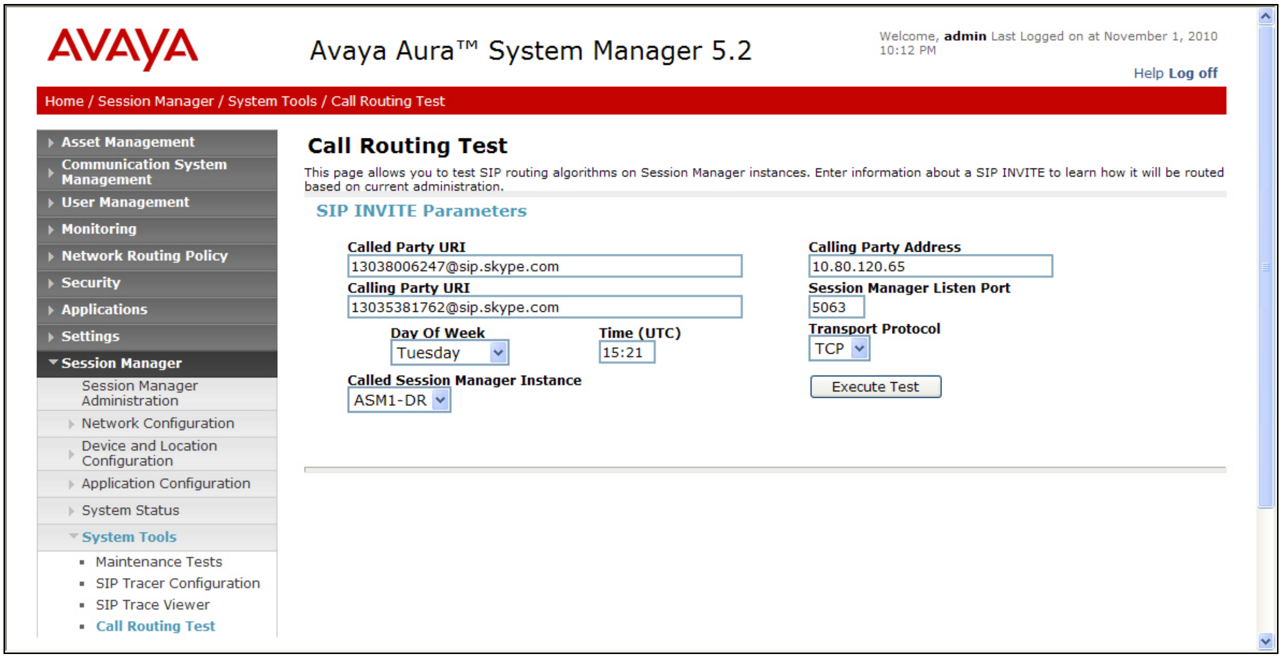

Figure 36: Call Routing Test

Then click on the **Execute Test** button. System Manager will check the routing algorithms and report on the success or failure of the provisioning.

The results of the test are then displayed as shown in Figure 37. At the top of the list, the heading **Routing Decisions** shows the final result. In the example, the call will be sent to SIP Entity **Voice Portal**. The next heading Routing Decision Process shows all the routing algorithm calculations.

Note that additional call routing tests can be performed. For example, to verify routing for calls transferred to the Communication Manager ACD, specify the VDN number (**6670201@sip.skype.com** without the "+" sign) in the **Called Party URI** field. All other values can be left as shown in **Figure 37**. When this test is run, the call will be sent to SIP Entity **S8730-port-5063**, which is the SIP Entity to route calls to Communication Manager.

Routing Decisions

Route < sip:13038006247@sip.skype.com > to SIP Entity Voice Portal (10.80.100.54). Terminating Location is AvayaCPE.

Routing Decision Process

Checking NRP to determine if this is a call to an emergency number.
Originating Location is 10_80_120. Using digits < 13038006247 > and host < sip.skype.com > for routing.
NRP Dial Patterns: No matches for digits < 13038006247 > and domain < sip.skype.com >.
NRP Dial Patterns: No matches for digits < 13038006247 > and domain < skype.com >.
NRP Dial Patterns: No matches for digits < 13038006247 > and domain < null >.
NRP Dial Patterns: No matches found for 10_80_120. Trying again using NRP Dial Patterns that specify -ALL- NRP Locations.
NRP Dial Patterns: Found a Dial Pattern match for pattern < 13038006247 > Min/Max length 11/11 and domain < sip.skype.com >.
NRP Routing Policies: Ranked destination NRP Sip Entities: Voice Portal.
NRP Routing Policies: Removing disabled routes.
NRP Routing Policies: Ranked destination NRP Sip Entities: Voice Portal.
NRP Adaptations: no Incoming Adaptation administered.
NRP Sip Entities: Originating SIP Entity is ACME1.
Originating Location is 10_80_120. Using digits < 13038006247 > and host < sip.skype.com > for routing.
NRP Dial Patterns: No matches for digits < 13038006247 > and domain < sip.skype.com >.
NRP Dial Patterns: No matches for digits < 13038006247 > and domain < skype.com >.
NRP Dial Patterns: No matches for digits < 13038006247 > and domain < null >.
NRP Dial Patterns: No matches found for 10_80_120. Trying again using NRP Dial Patterns that specify -ALL- NRP Locations.
NRP Dial Patterns: Found a Dial Pattern match for pattern < 13038006247 > Min/Max length 11/11 and domain < sip.skype.com >.
NRP Routing Policies: Ranked destination NRP Sip Entities: Voice Portal.
NRP Routing Policies: Removing disabled routes.
NRP Routing Policies: Ranked destination NRP Sip Entities: Voice Portal.
Adapting and proxying for SIP Entity Voice Portal.
NRP Entity Links: Found direct link to destination. Link uses TCP to port 5060.
NRP Adaptations: no Outgoing Adaptation administered.
Route < sip:13038006247@sip.skype.com > to SIP Entity Voice Portal (10.80.100.54). Terminating Location is AvayaCPE.

Figure 37: Call Routing Test - Results

7.3. Troubleshooting Tools

SIP protocol analyzers, such as Wireshark, can be used to capture SIP traces at the various interfaces. SIP traces can be instrumental in understanding SIP protocol issues resulting from configuration problems.

Chapter 14 of **Reference [10]** contains detailed steps for capturing SIP traces within Session Manager. SIP message tracing can be used to troubleshoot or monitor a selected Session Manager instance. SIP tracing logs incoming and outgoing SIP messages in the SM100 framework. Messages belonging to a user or Call ID for a call, or for a selected Session Manager instance, can be captured.

7.4. Verification Call Scenarios

Verification scenarios for the configuration described in these Application Notes include:

- Voice Portal access from the PSTN and from the Skype P2P Network using G.729.

  - Inbound call from Skype P2P user to Skype Business Account delivered to Avaya Voice Portal
- Voice Portal blind, consultative (supervised), and bridged transfers to Communication Manager ACD
- DTMF tone support

8. Conclusion

As illustrated in these Application Notes, Avaya Voice Portal, Avaya Aura® Session Manager 5.2, and Acme Packet Session Border Controllers can be configured to interoperate successfully with the Skype Connect service. This solution provides users of Avaya Voice Portal the ability to implement self-service applications over a Skype Connect trunk service connection.

9. Support

9.1. Avaya

For technical support on the Avaya VoIP products described in these Application Notes visit http://support.avaya.com

9.2. Skype

For technical support on the Skype Connect service, visit their online support at http://www.skype.com/support

10. References

10.1. Avaya

The following Avaya product documentation is available at http://support.avaya.com.

[1] *Application Notes for Avaya Aura® Communication Manager 5.2.1, Avaya Aura® Session Manager 5.2, and Acme Packet Net-Net 3800 Integration with Skype Connect R1.3 – Issue 1.0*

[2] *Planning for Voice Portal, June 2010*

[3] *Administering Voice Portal, June 2010*

[4] *Implementing Voice Portal on a single server, June 2010*

[5] *Implementing Voice Portal on multiple servers, June 2010*

[6] *Troubleshooting Voice Portal, June 2010*

[7] *Avaya Aura® Session Manager Overview, Doc ID 03-603323, Issue 2, Release 5.2, November 2009*

[8] *Installing Avaya Aura® Session Manager, Doc ID 03-603473, Issue 1.3, Release 5.2, January 2010*

[9] *Administering Avaya Aura® Session Manager, Doc ID 03-603324, Issue 2.1, Release 5.2, August 2010*

[10] *Maintaining and Troubleshooting Avaya Aura® Session Manager, Doc ID 03-603325, Issue 1.4, Release 5.2, September 2010*

[11] *Avaya Aura® Call Center Release 5.2, Automatic Call Distribution Reference, Doc ID 07-602568, Release 5.2, April 2009*

[12] *Avaya Aura® Call Center 5.2 Call Vectoring and Expert Agent Selection (EAS) Reference, Doc ID 07-600780, Release 5.2, April 2009*

10.2. Skype Connect

The following documents may be obtained by contacting your Skype Business Account Representative.

[13] *Skype Connect® Product Datasheet, Version 3.0,* 2010.

10.3. Acme Packet

The following Acme Packet product documentation is available at:
https://support.acmepacket.com/

[14] *Net-Net® 4000, ACLI Reference Guide, Release Version S-C6.1.0*
[15] *Net-Net® 4000 ACLI, Configuration Guide, Release Version S-C6.2.0*
[16] *Net-Net® 4000 Maintenance and Troubleshooting Guide, Release S-C6.2.0*

11. Appendix A – Acme Packet Net-Net 3800 Configuration

In addition to the SBC configuration identified in **Appendix A** of **Reference [1]**, the following additional modifications are made to support SIP REFER Method Call Transfer.

```
ANNOTATION: The Session Agent definition is associated with the Avaya Aura®
Session Manager.  In the reference configuration the following parameter is
enabled:

refer-call-transfer

In addition, note that the following header manipulations are set:

in-manipulationid        Avaya-incoming
out-manipulationid       AVP-manip-out
```

```
session-agent
      hostname                      10.80.100.24
      ip-address                    10.80.100.24
      port                          5063
      state                         enabled
      app-protocol                  SIP
      app-type
      transport-method              StaticTCP
      realm-id                      INTERNAL
      egress-realm-id
      description                   Avaya Aura Session Manager
      carriers
      allow-next-hop-lp             enabled
      constraints                   disabled
      max-sessions                  0
      max-inbound-sessions          0
      max-outbound-sessions         0
      max-burst-rate                0
      max-inbound-burst-rate        0
      max-outbound-burst-rate       0
      max-sustain-rate              0
      max-inbound-sustain-rate      0
      max-outbound-sustain-rate     0
      min-seizures                  5
      min-asr                       0
      time-to-resume                0
      ttr-no-response               0
      in-service-period             0
      burst-rate-window             0
      sustain-rate-window           0
      req-uri-carrier-mode          None
      proxy-mode
      redirect-action
      loose-routing                 enabled
      send-media-session            enabled
      response-map
      ping-method                   OPTIONS
      ping-interval                 300
      ping-send-mode                keep-alive
```

```
      ping-all-addresses            disabled
      ping-in-service-response-codes
      out-service-response-codes
      media-profiles
      in-translationid
      out-translationid
      trust-me                      disabled
      request-uri-headers
      stop-recurse                  408,486
      local-response-map
      ping-to-user-part
      ping-from-user-part
      li-trust-me                   disabled
      in-manipulationid             Avaya-incoming
      out-manipulationid            AVP-manip-out
      manipulation-string
      manipulation-pattern
      p-asserted-id
      trunk-group
      max-register-sustain-rate     0
      early-media-allow
      invalidate-registrations      disabled
      rfc2833-mode                  none
      rfc2833-payload               0
      codec-policy
      enforcement-profile
      refer-call-transfer           enabled
      reuse-connections             NONE
      tcp-keepalive                 none
      tcp-reconn-interval           10
      max-register-burst-rate       0
      register-burst-window         0
      sip-profile
      sip-isup-profile
      last-modified-by              admin@135.8.19.107
      last-modified-date            2010-10-14 18:54:43
```

ANNOTATION: In the reference configuration, the following header manipulations are added to the **Avaya-incoming** header manipulation rule to support passing User-to-User information to Communication Manager. See **Reference [1]** for the base **Avaya-incoming** header manipulation rule.

```
header-rule
      name                          requri
      header-name                   Refer-To
      action                        manipulate
      comparison-type               case-sensitive
      msg-type                      request
      methods                       REFER
      match-value
      new-value
      element-rule
            name                          getUUI
            parameter-name                User-to-User
            type                          uri-header
            action                        store
```

```
            match-val-type                  any
            comparison-type                 case-sensitive
            match-value
            new-value
      element-rule
            name                            appenduriuser
            parameter-name
            type                            uri-user
            action                          replace
            match-val-type                  any
            comparison-type                 boolean
            match-value                     $requri.$getUUI.$0
            new-value                       $ORIGINAL+UUI+$requri.$getUUI.$0
```

ANNOTATION: In the reference configuration, an additional header manipulation rule was created to support passing User-to-User information to Communication Manager.

```
sip-manipulation
      name                            AVP-manip-out
      description
      split-headers
      join-headers
      header-rule
            name                            get_UUI
            header-name                     Request-URI
            action                          manipulate
            comparison-type                 case-sensitive
            msg-type                        request
            methods                         INVITE
            match-value
            new-value
            element-rule
                  name                            store_UUI
                  parameter-name
                  type                            uri-user
                  action                          store
                  match-val-type                  any
                  comparison-type                 case-sensitive
                  match-value                     (.*)(UUI)(.*)
                  new-value
            element-rule
                  name                            get_UUI
                  parameter-name
                  type                            uri-user
                  action                          find-replace-all
                  match-val-type                  any
                  comparison-type                 case-sensitive
                  match-value                     (.*)(UUI)(.*)
                  new-value                       $get_UUI.$store_UUI.$1
      header-rule
            name                            get_UUI_To
            header-name                     To
            action                          manipulate
            comparison-type                 case-sensitive
            msg-type                        request
```

```
        methods                         INVITE
        match-value
        new-value
        element-rule
              name                            store_UUI
              parameter-name
              type                            uri-user
              action                          store
              match-val-type                  any
              comparison-type                 case-sensitive
              match-value                     (.*)(UUI)(.*)
              new-value
        element-rule
              name                            get_UUI
              parameter-name
              type                            uri-user
              action                          find-replace-all
              match-val-type                  any
              comparison-type                 case-sensitive
              match-value                     (.*)(UUI)(.*)
              new-value                       $get_UUI_To.$store_UUI.$1
  header-rule
        name                            get_UUI_Route
        header-name                     Route
        action                          manipulate
        comparison-type                 case-sensitive
        msg-type                        any
        methods                         INVITE
        match-value
        new-value
        element-rule
              name                            store_UUI
              parameter-name
              type                            uri-user
              action                          store
              match-val-type                  any
              comparison-type                 case-sensitive
              match-value                     (.*)(UUI)(.*)
              new-value
        element-rule
              name                            get_UUI
              parameter-name
              type                            uri-user
              action                          find-replace-all
              match-val-type                  any
              comparison-type                 case-sensitive
              match-value                     (.*)(UUI)(.*)
              new-value                       $get_UUI_Route.$store_UUI.$1
  header-rule
        name                            add_UUI
        header-name                     User-to-User
        action                          add
        comparison-type                 boolean
        msg-type                        request
        methods                         INVITE
        match-value                     $get_UUI.$store_UUI
        new-value                       $get_UUI.$store_UUI.$3
```

12. Appendix B – Voice Portal Test Application

Presented below is a sample Voice Portal Test Application used to verify basic operations, including bridged, blind, and consultative (supervised) transfers. These transfer methods include User-to-User data by setting the “aai=” parameter. Note that files containing VXML code typically reside on an external web application server. In the reference configuration, these files resided on the VPMS/MPP server in the following directory:

/opt/Avaya/VoicePortal/MPP/web/misc/avptestapp

For testing purposes, the VPMS/MPP server also served as a web application server.

12.1. “intro.vxml”

```
<?xml version="1.0" ?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml" xml:lang="en-US" >

<form id="form0">

        <field name="test_type">

                <prompt bargein="true" cond="session.connection.ccxml.values.test_page == 'true'">
                        <audio src="prompts/introccxml.wav"/>
                </prompt>

                <prompt bargein="true" cond="session.connection.ccxml.values.test_page ==
undefined">
                        <audio src="prompts/introvxml.wav"/>
                </prompt>

                <grammar src="builtin:dtmf/digits" />

                <filled>
                        <if cond="test_type == 1">
                                <goto next="asrtest.vxml"/>
                        <elseif cond="test_type == 2"/>
                                <goto next="ttstest.vxml"/>
                        <elseif cond="test_type == 3"/>
                                <goto next="testbridgetransfer.vxml"/>
                        <elseif cond="test_type == 4"/>
                                <goto next="testblindtransfer.vxml"/>
                        <elseif cond="test_type == 5"/>
                                <goto next="testconsulttransfer.vxml"/>
                        <elseif cond="test_type == 6"/>
                                <goto next="playprompts.vxml"/>
                        <elseif cond="session.connection.ccxml.values.test_page == 'true'"/>
                                <if cond="test_type > 9">
                                        <prompt bargein="false">
                                                <audio src="prompts/commonSorry.wav"/>
                                        </prompt>
                                        <clear namelist="test_type"/>
                                <elseif cond="test_type == 0"/>
                                        <prompt bargein="false">
                                                <audio src="prompts/Exit.wav"/>
                                        </prompt>
                                <else/>
                                        <exit namelist="test_type"/>
                                </if>
                        <else/>
                                <if cond="test_type == 7">
                                        <log expr="'Getting Ready To Exit'"/>
                                        <prompt bargein="false">
                                                <audio src="prompts/Exit.wav"/>
                                        </prompt>
                                        <exit/>
```

```
                                <else/>
                                        <prompt bargein="false">
                                                <audio src="prompts/commonSorry.wav"/>
                                        </prompt>
                                        <clear namelist="test_type"/>
                                </if>
                        </if>
                </filled>
```

12.2. “testbridgetransfer.vxml”

```
<?xml version="1.0" ?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml" xml:lang="en-US" >

        <var name="var1" expr="'tel:'"/>

        <form id="get_number">

                <field name="phone_number">

                        <prompt bargein="true">
                                <audio src="prompts/TransferGetNumber.wav"/>
                        </prompt>

                        <grammar src="builtin:dtmf/digits?minlength=1;maxlength=11" />

                        <noinput>
                                <prompt bargein="false">
                                        <audio src="prompts/TransferNoNumberSorry.wav"/>
                                </prompt>
                                <reprompt/>
                        </noinput>

                </field>

                      <transfer name="bridgetransfer" destexpr="var1 + phone_number"
transferaudio="prompts/monday_night.wav" type="bridge"
aai="04313233343536373839303132333435363738393031323334353637383930313233343536373839303132333435 36
373839303132333435363738393031323334353637383930313233343536373839303132333435363738393031323334353
6%3Bencoding%3Dhex">
                        <prompt bargein="true">
                                <audio src="prompts/bridgePerforming.wav"/>
                        </prompt>

                        <grammar src="builtin:dtmf/digits" />

                        <filled>
                                <if cond="bridgetransfer == 'busy'">
                                        <audio src="prompts/lineBusy.wav"/>
                                        <log> busy </log>
                                <elseif cond="bridgetransfer == 'noanswer'"/>
                                        <audio src="prompts/noAnswer.wav"/>
                                        <log> noanswer </log>
                                <elseif cond="bridgetransfer == 'network_busy'"/>
                                        <audio src="prompts/nwBusy.wav"/>
                                        <log> network_busy </log>
                                <elseif cond="bridgetransfer == 'near_end_disconnect'"/>
                                        <audio src="prompts/nearEndDisc.wav"/>
                                        <log> near_end_disconnect </log>
                                <elseif cond="bridgetransfer == 'unknown'"/>
                                        <audio src="prompts/failedUnknown.wav"/>
                                        <log> unknown </log>
                                <elseif cond="bridgetransfer == 'maxtime_disconnect'"/>
                                        <audio src="prompts/maxTimeDisc.wav"/>
                                        <log> maxtime_disconnect </log>
                                <elseif cond="bridgetransfer == 'network_disconnect'"/>
                                        <audio src="prompts/nwDisc.wav"/>
                                        <log> network_disconnect </log>
                                <elseif cond="bridgetransfer == 'far_end_disconnect'"/>
                                        <audio src="prompts/farEndDisconnect.wav"/>
```

```
                                        <log> far_end_disconnect </log>
                                </if>

                                <prompt bargein="false">
                                        <audio src="prompts/bridgeThanks.wav"/>
                                </prompt>

                                <goto next="intro.vxml"/>
                        </filled>

                </transfer>

                <catch event="connection.disconnect.hangup">
                        <log> connection.disconnect.hangup </log>
                        <exit />
                </catch>
                <catch event="error.connection.noauthorization">
                        <log> error.connection.noauthorization </log>
                        <goto next="intro.vxml"/>
                </catch>
                <catch event="error.connection.baddestination">
                        <log> error.connection.baddestination </log>
                        <goto next="intro.vxml"/>
                </catch>
                <catch event="error.unsupported.transfer.bridge">
                        <log> error.unsupported.transfer.blind </log>
                        <goto next="intro.vxml"/>
                </catch>
                <catch event="error.unsupported.uri">
                        <log> error.unsupported.uri </log>
                        <goto next="intro.vxml"/>
                </catch>
                <catch event="error.connection.noroute">
                        <log> error.connection.noroute </log>
                        <goto next="intro.vxml"/>
                </catch>
                <catch event="error.connection.noresource">
                        <log> error.connection.noresource </log>
                        <goto next="intro.vxml"/>
                </catch>
        </form>
</vxml>
```

12.3. “testconsulttransfer.vxml”

```
<?xml version="1.0" ?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml" xml:lang="en-US" >

        <var name="var1" expr="'tel:'"/>

        <form id="get_number">
                <field name="phone_number">
                <prompt bargein="true">
                        <audio src="prompts/TransferGetNumber.wav"/>
                </prompt>

        <grammar src="builtin:dtmf/digits?minlength=1;maxlength=12" />

        <noinput>
                <prompt bargein="false">
                        <audio src="prompts/TransferNoNumberSorry.wav"/>
                </prompt>
                <reprompt/>
        </noinput>

        </field>
                <transfer name="consultationtransfer" destexpr="var1 + phone_number"
transferaudio="prompts/monday_night.wav" type="consultation"
aai="0431323334353637383930313233343536373839303132333435363738393031323334353637383930313233343536
373839303132333435363738393031323334353637383930313233343536373839303132333435363738393031323334353
6%3Bencoding%3Dhex">
                        <prompt bargein="false">
                                <audio src="prompts/consultPerforming.wav"/>
                        </prompt>

                        <filled>
                                <if cond="consultationtransfer == 'busy'">
                                        <audio src="prompts/lineBusy.wav"/>
                                        <log> busy </log>
                                        <goto next="intro.vxml"/>
                                <elseif cond="consultationtransfer == 'noanswer'"/>
                                        <audio src="prompts/noAnswer.wav"/>
                                        <log> noanswer </log>
                                        <goto next="intro.vxml"/>
                                <elseif cond="consultationtransfer == 'near_end_disconnect'"/>
                                        <audio src="prompts/nearEndDisc.wav"/>
                                        <log> near_end_disconnect </log>
                                        <goto next="intro.vxml"/>
                                <elseif cond="consultationtransfer == 'network_busy'"/>
                                        <audio src="prompts/nwBusy.wav"/>
                                        <log> network_busy </log>
                                        <goto next="intro.vxml"/>
                                <elseif cond="consultationtransfer == 'unknown'"/>
                                        <audio src="prompts/failedUnknown.wav"/>
                                        <log> unknown </log>
                                        <goto next="intro.vxml"/>
                                </if>
                        </filled>

                </transfer>

                <catch event="connection.disconnect.hangup">
                        <log> connection.disconnect.hangup </log>
                        <goto next="intro.vxml"/>
                </catch>

                <catch event="error.connection.noauthorization">
                        <log> connection.disconnect.transfer </log>
                </catch>

                <catch event="error.connection.noauthorization">
                        <log> error.connection.noauthorization </log>
```

```
                        <goto next="intro.vxml"/>
                </catch>

                <catch event="error.connection.baddestination">
                        <log> error.connection.baddestination </log>
                        <goto next="intro.vxml"/>
                </catch>

                <catch event="error.connection.noroute">
                        <log> error.connection.noroute </log>
                        <goto next="intro.vxml"/>
                </catch>

                <catch event="error.connection.noresource">
                        <log> error.connection.noresource </log>
                        <goto next="intro.vxml"/>
                </catch>

                <catch event="error.unsupported.uri">
                        <log> error.unsupported.uri </log>
                        <goto next="intro.vxml"/>
                </catch>

                <catch event="error.unsupported.transfer.consultation">
                        <log> error.unsupported.transfer.consultation </log>
                        <goto next="intro.vxml"/>
                </catch>

        </form>
</vxml>
```

12.4. “testblindtransfer.vxml”

```
<?xml version="1.0" ?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml" xml:lang="en-US" >

        <var name="var1" expr="'tel:'"/>

        <form id="get_number">

                <field name="phone_number">

                        <prompt bargein="true">
                                <audio src="prompts/TransferGetNumber.wav"/>
                        </prompt>

                        <grammar src="builtin:dtmf/digits?minlength=1;maxlength=11" />

                        <noinput>
                                <prompt bargein="false">
                                        <audio src="prompts/TransferNoNumberSorry.wav"/>
                                </prompt>
                                <reprompt/>
                        </noinput>

                </field>

               <transfer name="blindtransfer" destexpr="var1 + phone_number" type="blind"
aai="043132333435363738393031323334353637383930313233343536373839303132333435363738393031323334353637383930313233343536
373839303132333435363738393031323334353637383930313233343536373839303132333435363738393031323334353
6%3Bencoding%3Dhex">

                        <prompt bargein="false">
                                <audio src="prompts/blindPerforming.wav"/>
                        </prompt>

                        <filled>
                                <if cond="blindtransfer == 'near_end_disconnect'">
                                        <audio src="prompts/nearEndDisc.wav"/>
                                        <log> near_end_disconnect </log>
```

```
                                <elseif cond="blindtransfer == 'unknown'"/>
                                        <audio src="prompts/failedUnknown.wav"/>
                                        <log> unknown </log>
                                </if>
                                <goto next="intro.vxml"/>
                        </filled>

                </transfer>

                <catch event="connection.disconnect.transfer">
                        <log> connection.disconnect.transfer </log>
                        <exit />
                </catch>
                <catch event="error.connection.noauthorization">
                        <log> error.connection.noauthorization </log>
                        <goto next="intro.vxml"/>
                </catch>
                <catch event="error.connection.baddestination">
                        <log> error.connection.baddestination </log>
                        <goto next="intro.vxml"/>
                </catch>
                <catch event="error.unsupported.uri">
                        <log> error.unsupported.uri </log>
                        <goto next="intro.vxml"/>
                </catch>
                <catch event="error.unsupported.transfer.blind">
                        <log> error.unsupported.transfer.blind </log>
                        <goto next="intro.vxml"/>
                </catch>
        </form>
</vxml>
```

Please e-mail any questions or comments pertaining to these Application Notes along with the full title name and filename, located in the lower right corner, directly to the Avaya Solution & Interoperability Test Lab at interoplabnotes@list.avaya.com