Spectralink 84-Series Wireless Telephone

**Avaya Aura Communication Manager**

and

**Avaya Aura Session Manager**

**Interoperability Notes**

(to be used as an addendum to the Avaya DevConnect compliance testing Application Notes)
Copyright Notice

© 2014 Spectralink Corporation All rights reserved. Spectralink™, the Spectralink logo and the names and marks associated with Spectralink's products are trademarks and/or service marks of Spectralink Corporation and are common law marks in the United States and various other countries. All other trademarks are property of their respective owners. No portion hereof may be reproduced or transmitted in any form or by any means, for any purpose other than the recipient's personal use, without the express written permission of Spectralink.

All rights reserved under the International and pan-American Copyright Conventions. No part of this manual, or the software described herein, may be reproduced or transmitted in any form or by any means, or translated into another language or format, in whole or in part, without the express written permission of Spectralink Corporation.

Do not remove (or allow any third party to remove) any product identification, copyright or other notices.

Notice

Spectralink Corporation has prepared this document for use by Spectralink personnel and customers. The drawings and specifications contained herein are the property of Spectralink and shall be neither reproduced in whole or in part without the prior written approval of Spectralink, nor be implied to grant any license to make, use, or sell equipment manufactured in accordance herewith.

Spectralink reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult Spectralink to determine whether any such changes have been made.

NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS DOCUMENT INCLUDING BUT NOT LIMITED TO STATEMENTS REGARDING CAPACITY, RESPONSE-TIME PERFORMANCE, SUITABILITY FOR USE, OR PERFORMANCE OF PRODUCTS DESCRIBED HEREIN SHALL BE DEEMED TO BE A WARRANTY BY SPECTRALINK FOR ANY PURPOSE, OR GIVE RISE TO ANY LIABILITY OF SPECTRALINK WHATSOEVER.

Contact Information

US Location
800-775-5330
Spectralink Corporation
2560 55th Street
Boulder, CO 80301
info@Spectralink.com

European Location
+45 7560 2850
Spectralink Europe ApS
Langmarksvej 34
8700 Horsens, Denmark
infodk@Spectralink.com
Chapter 4: Troubleshooting and Analyzing SIP on the Avaya Aura System ................................................................. 37

How to gather SIP traces on the Aura system through System Manager ................................................. 37
  Dropped packets due to firewall failures ................................................................................................. 42
Working with the Aura Session Manager’s Integrated SIP Firewall ......................................................... 44
Verifying the Registration Status of the 84-Series Handset on the Session Manager ...................................... 50
DSCP Values .............................................................................................................................................. 50
  Call Control DSCP ............................................................................................................................... 51
  Audio DSCP ....................................................................................................................................... 52
Appendix A .................................................................................................................. 55
Sample Configuration Files............................................................................................. 55
  00000000000.cfg........................................................................................................ 55
  site.cfg......................................................................................................................... 56
  00000000000-directory.xml.......................................................................................... 57
  00907a0dccbdf-ext.cfg................................................................................................. 58
  00907a0f1907-ext.cfg.................................................................................................. 58
  00907a0f1907-ext.cfg.................................................................................................. 59

Appendix B .................................................................................................................... 60
Creating a DNS SRV record in a Windows Server.......................................................... 60
Create a DNS A Name record for the Callserver (or Verify an A name record already
exists for the Callserver)............................................................................................... 62
Create a SRV Record for the Callserver......................................................................... 65
DNS Verification and Troubleshooting............................................................................ 68
About This Guide

This guide was written as an addendum to the Avaya DevConnect produced Application Notes describing the procedures for configuring the 8400 by Spectralink for compliance testing with the Avaya Aura Communication Manager and Avaya Aura Session Manager [http://support.Spectralink.com/sites/default/files/resource_files/spectra8400SM63.pdf](http://support.Spectralink.com/sites/default/files/resource_files/spectra8400SM63.pdf).

Administrators and those interested in the 8400 / Avaya Aura integration should begin their investigation using that document, and refer to this document for more in-depth feature descriptions and analysis, troubleshooting tips and tricks, and other interoperability notes discovered in the course of testing in Spectralink’s labs.

Product Support

Spectralink wants you to have a successful installation. If you have questions please contact the Customer Support Hotline at 1-800-775-5330.

The hotline is open Monday through Friday, 6 a.m. to 6 p.m. Mountain Time.

For Technical Support: mailto:technicalsupport@Spectralink.com

For Knowledge Base: [http://support.Spectralink.com](http://support.Spectralink.com)

For Return Material Authorization: mailto:nalarma@Spectralink.com

Spectralink References

All Spectralink documents are available at [http://support.Spectralink.com](http://support.Spectralink.com).
To go to a specific product page:
Select the Product Category and Product Type from the dropdown lists and then select the product from the next page. All resources for that particular product are displayed by default under the All tab. Documents, downloads and other resources are sorted by the date they were created so the most recently created resource is at the top of the list. You can further sort the list by the tabs across the top of the list to find exactly what you are looking for. Click the title to open the link.

Specific Spectralink references

*Spectralink 84-Series SLIC Administration Guide* The SLIC tool provides step-by-step instructions for configuring wireless settings required for the handsets to associate with the wireless LAN.

*Spectralink 84-Series Wireless Telephone Deployment Guide* The Deployment Guide provides sequential information for provisioning and deploying the handsets. It covers deployment using the SLIC tool as well as manual deployment.

*Spectralink 84-Series Wireless Telephone Administration Guide* The Admin Guide is a companion guide to the Deployment Guide. It provides detailed information about settings and options available to the administrator through configuration files and through the handsets. Time-saving shortcuts, troubleshooting tips and other important maintenance instructions are also found in this document.


Avaya DevConnect documents available on the Spectralink Support Site


Application Notes for Avaya Aura Session Manager 6.3 covers the basic configuration and requirements of a SIP based endpoint that will support the 84-Series phone in the Avaya Aura Communications Manager and Session Manager. It also shows how to build and configure a SIP endpoint through the Avaya Communication Manager and System Manager interfaces. These Interoperability Notes are an extension of that document and cover additional variations, features, and configurations you may wish to utilize in order to enhance the 84-Series / Avaya Aura Integration.

*Application Notes for PIVOT by Spectralink (87-Series) Wireless SIP Telephones and Avaya Aura Communication Manager and Avaya Aura Session Manager Document*
While Application Notes for Spectralink 84-Series Wireless Telephones and Avaya Aura Communications Manager and Avaya Aura Session Manager 1.0 provides the steps necessary to provision a SIP endpoint, they do not cover the initial configuration or license requirements for SIP implementation on the Avaya Aura system. These requirements are described in the notes for the 87-Series and can be applied to an 84-Series installation. This document also contains additional details regarding License requirements and initial SIP configuration.

Avaya Documentation

This document does not attempt to cover even a small subset of the features and functionality available in the Avaya Aura Communications Manager and Session Manager. Please navigate to the Avaya support site for the latest Avaya branded documentation:

https://support.avaya.com/documents/

Conventions Used In This Document

Icons

Icons indicate extra information about nearby text.

- **Warning**
  
  The *Warning* icon highlights an action you must perform (or avoid) to avoid exposing yourself or others to hazardous conditions.

- **Caution**
  
  The *Caution* icon highlights information you need to know to avoid a hazard that could potentially impact device performance, application functionality, successful feature configuration and/or affect handset or network performance.

- **Note**
  
  The *Note* icon highlights information of interest or important information that will help you be successful in accomplishing a procedure or understanding a concept.

- **Tip**
  
  The *Tip* icon highlights information that may be valuable or helpful for users to know, such as special techniques, shortcut methods, or information that will make user tasks easier to perform.
Web
The Web Info icon highlights supplementary information available online such as documents or downloads on support.Spectralink.com or other locations.

Timesaver
A time-saving tip is typically used to mention or highlight a faster or alternative method for users who may already be familiar with the operation or method being discussed.

Admin Tip
This tip advises the administrator of a smarter, more productive or alternative method of performing an administrator-level task or procedure.

Power User
A Power User Tip is typically reserved for information directed specifically at high-level users who are familiar with the information or procedure being discussed and are looking for better or more efficient ways of performing the task. For example, this might highlight customization of a feature for a specific purpose.

Troubleshooting
This element can be used in any type of document and is typically used to highlight information to help you solve a relevant problem you may encounter, or to point to other relevant troubleshooting reference information.

Settings
The Settings icon highlights information to help you zero in on settings you need to choose for a specific behavior, to enable a specific feature, or access customization options.

Typography
A few typographic conventions, listed next, are used in this guide to distinguish types of in-text information.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bold</td>
<td>Highlights interface items such as menus, soft keys, file names, and directories. Also used to represent menu selections and text entry to the handset.</td>
</tr>
</tbody>
</table>
This guide also uses a few writing conventions to distinguish conditional information.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;MACaddress&gt;</td>
<td>Indicates that you must enter information specific to your installation, handset, or network. For example, when you see <code>&lt;MACaddress&gt;</code>, enter your handset's 12-digit MAC address. If you see <code>&lt;installed-directory&gt;</code>, enter the path to your installation directory.</td>
</tr>
<tr>
<td>&gt;</td>
<td>Indicates that you need to select an item from a menu. For example, <strong>Settings</strong>&gt; <strong>Basic</strong> indicates that you need to select <strong>Basic</strong> from the <strong>Settings</strong> menu.</td>
</tr>
</tbody>
</table>
Chapter 1: Overview

System Diagram

Below is a system diagram depicting the lab setup used to test the Spectralink 84-Series interoperability with the Avaya Aura system.

Test infrastructure version information

- Avaya Aura System Manager Version: 6.3.7.7.2275
- Avaya Aura Session Manager Version: 6.3.7.0.637008
- Avaya Communications Manager Version: 6.3-03.124.0
- Avaya Communications Manager Messaging Version: 6.3-26.0
- Spectralink 8400 Series Handset Software Version: 4.7.0.2324 and 4.7.0.2327
Motorola 6532 Access Point Software Version: 5.2.3.0-023D

**Feature Configuration and Test Summary**

A description of each feature tested and comments about the functionality can be found in the Feature Configuration and Test details section of this guide.

<table>
<thead>
<tr>
<th>Features Tested</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct to Avaya Aura Session Manager SIP Registration</td>
<td>Y</td>
</tr>
<tr>
<td>SIP Digest Authentication</td>
<td>Y</td>
</tr>
<tr>
<td>Basic Calls</td>
<td>Y</td>
</tr>
<tr>
<td>Voicemail Integration</td>
<td>Y</td>
</tr>
<tr>
<td>Message Waiting Indication (MWI)</td>
<td>Y</td>
</tr>
<tr>
<td>Call Waiting</td>
<td>Y</td>
</tr>
<tr>
<td>Multiple Calls Per Line Key (or per registration)</td>
<td>Y</td>
</tr>
<tr>
<td>Conference: 3-way</td>
<td>Y</td>
</tr>
<tr>
<td>Transfer: Blind</td>
<td>Y</td>
</tr>
<tr>
<td>Transfer: Announced</td>
<td>Y</td>
</tr>
<tr>
<td>Transfer: Attended</td>
<td>Y</td>
</tr>
<tr>
<td>Caller ID</td>
<td>Y</td>
</tr>
<tr>
<td>Hold and Resume</td>
<td>Y</td>
</tr>
<tr>
<td>Music On Hold</td>
<td>Y</td>
</tr>
<tr>
<td>Call Reject</td>
<td>Y</td>
</tr>
<tr>
<td>Do Not Disturb</td>
<td>Y</td>
</tr>
<tr>
<td>Call Park</td>
<td>Y</td>
</tr>
<tr>
<td>DTMF via RFC2833</td>
<td>Y</td>
</tr>
<tr>
<td>Call Forward</td>
<td>Y</td>
</tr>
<tr>
<td>Feature Access Codes</td>
<td>Y</td>
</tr>
<tr>
<td>TCP</td>
<td>Y</td>
</tr>
<tr>
<td>G.711u, G.711a, G.729A, and G.722 Codecs</td>
<td>Y</td>
</tr>
<tr>
<td>Multiple Line Keys (or registrations) per handset</td>
<td>Y</td>
</tr>
<tr>
<td>‘Paired’ lines (shared line, bridged line, etc. – ‘ring both phones’ generally)</td>
<td>Y</td>
</tr>
<tr>
<td>Trunk Calling</td>
<td>Y</td>
</tr>
<tr>
<td>Integration with 46xxsettings file</td>
<td>N</td>
</tr>
<tr>
<td>Personal Profile Manager (PPM) Integration</td>
<td>N</td>
</tr>
<tr>
<td>Avaya Presence</td>
<td>N</td>
</tr>
</tbody>
</table>
### Features Not Tested

<table>
<thead>
<tr>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failover / Fallback / Redundancy / Resiliency</td>
</tr>
<tr>
<td>TLS</td>
</tr>
<tr>
<td>SRTP</td>
</tr>
</tbody>
</table>
Chapter 2: Before You Start

This document presumes that two conditions exist in your Avaya Aura deployment:

- SIP is provisioned and functioning on the Avaya Aura system.
- SIP User Profiles and associated Endpoints have been provisioned in the Avaya Aura system to support the Spectralink 84-Series handsets.

If both or one of these are not yet done, see the relevant Avaya Aura DevConnect guide(s).
Configure the Spectralink 84-Series Handset

Connect the handsets to the wireless LAN

The first step in connecting the Spectralink 84-Series handset to the Avaya Aura system is to get the handset connected to the wireless LAN. Detailed discussions of this topic are available through the Spectralink support web site: http://support.Spectralink.com/products/wifi/Spectralink-84-series-wireless-telephone. See Spectralink 84-Series Wireless Telephone Deployment Guide or the Spectralink 84-Series SLIC Administration Guide.

Create and modify parameters for your deployment

A significant variable in provisioning the Spectralink 84-Series handsets is setting the parameters in the .cfg files provided with the software. The test scenario conducted in our labs modified two files provided with Spectralink software 4.7.0: Flat Deployment files site.cfg and MACaddress-ext.cfg. We also included a 000000000000-director.xml file that could be used to pre-populate useful Feature Access Codes. The parameters for these files should provide single line functionality and enough information to allow the Spectralink 84-Series’s registration with the Avaya Session Manager.

Use the Spectralink 84-Series Deployment Guide for an in-depth discussion of provisioning server requirements, other provisioning deployment configuration file models, and a high level over-view of the provisioning process.

Per-phone configuration

The MACaddress-ext.cfg file provides per-handset line registration parameters. One file is created for each handset that is deployed. The following parameters were used in the MACaddress-ext.cfg file for the test scenario. These parameters align with those provided by the MACaddress-ext.cfg template provided for the Flat Deployment scenario.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reg.1.address=&quot;<a href="mailto:4615@engr.local">4615@engr.local</a>&quot;</td>
<td>specifies the SIP URI address to be used for registration. Note the inclusion of the SIP Domain Name</td>
</tr>
<tr>
<td>reg.1.displayName=&quot;4615&quot;</td>
<td>specifies the display name used in SIP Signaling for the line</td>
</tr>
<tr>
<td>reg.1.label=&quot;4615&quot;</td>
<td>specifies the label that will appear on the handset for the line</td>
</tr>
<tr>
<td>reg.1.auth.userID=&quot;4615&quot;</td>
<td>specifies the authentication user id to be used for response to digest authentication challenges</td>
</tr>
<tr>
<td>reg.1.auth.password=&quot;4615&quot;</td>
<td>specifies the authentication password to be used for response to digest authentication challenges</td>
</tr>
<tr>
<td>msg.mwi.1.subscribe=&quot;4615&quot;</td>
<td>specifies the value to use in SIP Subscribe messages for this registration when requesting to be notified of Voice Mail Message Waiting Messages</td>
</tr>
</tbody>
</table>
Site-wide configuration

The site.cfg file provides global or site-wide parameters. One file is configured for the facility. The following parameters were used in the test scenario. Some of these conflict with the parameters provided by the site.cfg template provided for the Flat Deployment scenario, as noted. Note that in the example below all of the parameters below np.normal.alert.messageWaiting.tonePattern are simply matters of personal preference, and not necessary for basic registration or functionality. Additional parameters included as personal preferences for logging and polling are also included in the sample config file set in Appendix A. Hopefully they serve to provide examples of the types of parameters you might wish to enable for all handset users at your site.

<table>
<thead>
<tr>
<th>Parameter Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>reg.1.x parameters replace the SIPserver volpProt.x parameters</td>
<td></td>
</tr>
<tr>
<td>reg.1.server.1.address=&quot;172.29.102.92&quot;</td>
<td>specifies the address of the proxy server that we will register with i.e. the Aura Session Manager</td>
</tr>
<tr>
<td>reg.1.server.1.port=&quot;5060&quot;</td>
<td>specifies the port we will use to attempt registrations</td>
</tr>
<tr>
<td>reg.1.server.1.expires=&quot;300&quot;</td>
<td>specifies the desired registration interval. Note that this value significantly reduces the default interval and is recommended due to special circumstances presented with wireless devices that can be turned off.</td>
</tr>
<tr>
<td>volpProt.SIP.use486forReject=&quot;1&quot;</td>
<td>specifies the handset should send 486, rather than 603 messages when rejecting a call</td>
</tr>
<tr>
<td>up.oneTouchVoicemail=&quot;1&quot;</td>
<td>enables one-touch voicemail dialing</td>
</tr>
<tr>
<td>msg.mwi.1.callBackMode=&quot;contact&quot;</td>
<td>specifies that we will contact the server using a call with the number specified by msg.mwi.1.callback. Note that &quot;registration&quot; is the parameters in the template. &quot;Registration&quot; was not tested.</td>
</tr>
<tr>
<td>msg.mwi.1.callBack=&quot;4999&quot;</td>
<td>specifies the number the phone will dial when attempting to contacting the Voice Mail system</td>
</tr>
<tr>
<td>np.normal.alert.messageWaiting.tonePattern=&quot;silent&quot;</td>
<td>specifies that when message waiting indications are received, the phone will not play an audible tone</td>
</tr>
</tbody>
</table>

Parameters below are optional and are included in the site.cfg template, except as noted.

<table>
<thead>
<tr>
<th>Parameter Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dialplan.digitmap=&quot;&quot;</td>
<td>Nulls the existing dialplan</td>
</tr>
<tr>
<td>dialplan.impossibleMatchHandling=&quot;2&quot;</td>
<td>Forces the phone to wait for the user to press the send key before delivering digits to the reg.1.server</td>
</tr>
<tr>
<td>up.screenCapture.enabled=&quot;1&quot;</td>
<td>enables screen captures on your handset - useful for bug reporting. Not provided in the site.cfg template.</td>
</tr>
</tbody>
</table>
The 000000000000-directory.xml file provides a way to populate handsets with Feature Access Codes or directory numbers you would like all handsets to receive. Note that once a handset is deployed and in use, it will create a `<MAC-Address>-directory.xml` file and will ignore the 000000000000-directory.xml file, so this method of provisioning is useful only for pre-populating these values at the time of deployment. Please see Appendix A for a sample file containing these types of values.

Advanced feature configuration

For more advanced feature configuration recommendations related directly to the Avaya Aura System please consult the SIP Feature Configuration and Configuration Parameter Test Details section of this document.

For feature configuration items such as dial plan configuration recommendations, Push To Talk (PTT), logging level changes or other general configuration items please consult the Spectralink 84-Series Deployment Guide, and the Spectralink 84-Series Administration Guide.

**Verify Configuration and Registration**

**Configuration files are received by the handset**

1. On the Handset, Navigate to **Settings > Status > Platform > Configuration**.

   The configuration screen displays the IP address of the server, the provisioning protocol being used, the .cfg files it is using, and detailed information about the number of parameters accepted from each configuration file, any duplicate parameters, and any errors.

2. Ensure that any errors or unintended duplicate parameters are corrected before leaving the site. Also ensure that the number of parameters accepted from each file aligns with the number of parameters expected.

**The handset is registered**

3. Verify Registration Status on the handset by checking to see if the Spectralink 84-Series has successfully registered all configured line appearances.

4. Look in the upper right hand corner of the idle screen for the green checkmark. This indicates the handset believes it has successfully registered all lines:
You can also see the status of each individual line registered (useful for multi-line handsets) by Navigating to **Settings > Status > Lines**.
Test the Solution

Once the device’s registration has been confirmed, a basic functionality test should be performed. Spectralink recommends running the following tests at a minimum in order to verify proper 84-Series handset / Avaya Aura system interaction:

- Confirm Registration Status on the handset and in the Session Manager
- Basic Call to and from the 84-Series handset to another Avaya device
- Call Transfer the 84-Series handset to another device, and use the 84-Series handset to conduct a transfer
- Perform a conference with the 84-Series handset, using the 84-Series handset as the conference initiator and test using the 84-Series handset as a conference participant
- Hold and resume a call
- Leave a voicemail for the 84-Series handset (if equipped) – Ensure message Waiting Indication is delivered. Call the voicemail system from the 84-Series handset and retrieve the call.
- Place a call to a PSTN number equipped with a menu system and verify the functionality of DTMF tones to navigate the menus.
- Verify other functionality of interest
Chapter 3: SIP Feature Configuration and Configuration Parameter Test Details

Avaya Aura SIP Registration Interval

Spectralink 84-Series handsets register directly to the Avaya Aura Session Manager SIP Entity. The default registration period requested by the 84-Series handsets is 3600s, or an hour. However, by shortening this interval, the Aura system can more quickly realize that a phone is no longer powered up or in range of a wireless access point and make more intelligent call routing decisions based on that data. The trade off is that requiring re-registration too often can also create extra traffic on a network and create additional SIP Notifications. Spectralink lab testing has found 5 minutes to be a reasonable amount of time to require re-registration. This behavior can be set using the parameter:

```
reg.1.server.1.expires="300"
```

This parameter will cause the phone to request a session expires value of 5 minutes in registration and voicemail subscription events. The default (3600) will work, but does not give the advantage of allowing the server to more reliably track the phones’ current state.

The Aura system also requires that the SIP Domain (a full SIP URI) be included in registration requests and invites from connected SIP endpoints. Below we will discuss two separate ways to handle this.

Method 1

This is the method included in the sample configuration file in Appendix A. This method is “shorter” but not quite as flexible. In this method we utilize the following two parameters to ensure that registration requests and invites are sent to the Aura system utilizing the correct format:

```
reg.1.server.1.address="172.29.102.92"
reg.1.address="4615@mydomain.local"
```

In this example you would replace the 172.29.102.92 address with the ip address or DNS A-Name of your sites’ Session Manager. The 8400 will direct registration requests, subscriptions, and invites to this location.

In this example you would replace the 4615 value with your extension number and the mydomain.local with your SIP Domain. Including the domain in the reg.1.address as shown here is required for correct interoperation with the Aura Callserver unless Method #2 below (SRV records) is utilized to achieve SIP registration.
Method 2

This method allows us to utilize a DNS SRV record to locate the PBX, determine the protocol (TCP or UDP) and port number to use for SIP communication with the PBX, and might also be utilized to attempt registration to a secondary Session Manager in case of failures (this latter part was untested in Spectralink’s labs as only one Session Manager was available for testing, but this configuration method might be interesting for further exploring this option.)

If we adapt the above example where we have a Callserver with an ip address of 172.29.102.92 and it requires a domain of mydomain.local, then there are two main tasks we will need to accomplish; first we need to configure the 84-Series handset appropriately to find and resolve this address, and then we need to create a DNS SRV record in the customer provided DNS server that will point requests made to the SIP Domain name along to the correct Callserver. Appendix B of this document should provide an example of the type of steps you might take to do this on the Windows server itself, but on the 8400 phone we would need to specify:

```
reg.1.server.1.address="mydomain.local"
```

In this example you would replace mydomain.local with your SIP domain name, and ensure there is a supporting DNS SRV record in the DNS server that will resolve queries looking for sip service at mydomain.local to the correct Callserver address, protocol and port number. The SIP Domain name corresponds to the Domain name that you specified for Session Manager’s Signaling Group created in the Communication Manager.

```
reg.1.address="4615"
```

In this example you would replace the 4615 value with your extension number.

Note: If utilizing SRV records do NOT specify a reg.1.server.1.port value or a reg.1.server.1.transport value as these will prevent the Spectralink 84-Series phone from attempting SRV lookups.

**SIP Digest Authentication**

The Session Manager requires the use of SIP Digest Authentication.

```
reg.1.auth.userId="4615"
```

In this example you would replace the 4615 value with your extension number. The reg.1.auth.userId is the SIP Digest Authentication username, and corresponds to the Login Name field of your User in System Manager.

```
reg.1.auth.password="1234"
```

In this example you would replace 1234 with your SIP Digest Authentication Password. This value corresponds to the Communication Profile Password you created for your User in System Manager.
**Basic Calls**

This functionality was tested by calling between Spectralink 84-Series handsets as well as to and from an Avaya 9650 phone. No special 8400 configuration parameters should be required in order to realize this ability.

**Voicemail Integration**

In many PBX integrations, SIP devices are automatically subscribed to receive Message Waiting Indicators (MWI's) when they register. In the course of our testing, however, we determined that it was necessary to create a separate subscription to the Session Manager for message waiting notifications in order to receive MWI notifications for voice mail messages. Mailboxes built in our labs were built and configured as with other Avaya extensions. Communications Manager Messaging did not provide notifications including the number of waiting messages. MWI notifications are delivered to SIP endpoints with a simple yes / no status. The below parameters were found to help optimize the CMM Voicemail integration with the 8400 phones:

```plaintext
msg.mwi.1.callBack="4999"
```

In this example, you will need to replace the number 4999, with the voicemail system's pilot number. This is the number the phone should dial to access the Voice Mail system.

```plaintext
msg.mwi.1.subscribe="4615"
```

In this example, you will need to replace the number 4615, with the extension number of your phone. This is the extension number that will be utilized in the phones' Message-summary Subscribe requests.

```plaintext
msg.mwi.callBackMode="contact"
```

This tells the phone to dial the msg.mwi.1.callBack number rather than the phones' own extension number in order to access the VM system

```plaintext
up.oneTouchVoiceMail="1"
```

This tells the phone to skip the message summary screen (Urgent / New / Old). Since the Avaya only indicates messages in a yes / no format, this screen does not seem to provide valuable information for our Message summaries.

```plaintext
np.normal.alert.messageWaiting.tonePattern="silent"
```

In our lab testing, we observed that every time the phone subscribes and receives another Message Waiting = yes Notify message from the Callserver it will tweedle again. So, we can set the MWI tone to silent, but must understand that if we do this, we won't hear the tone in the first place; though the envelope indicating that you do have a message will, of course be present. Provided your end users are utilizing the "Normal" notification profile, we could silence this pesky alert tone using the above setting. Note that this won't keep the phones' screen from illuminating every time it receives one of these, as it considers this a new alert. Additionally, if your end users are utilizing a different notification profile, they would need to set the same thing
for that profile. This setting can also be performed through the phones’ UI, but you may wish to set this through a site level config file.

**Message Waiting Indication (MWI)**

Parameters described in the Voicemail Integration section above were all that we found to be required to realize successful Message Waiting Indications.

**Call Waiting**

By default, when you build an extension, the Avaya Aura system places three call-appearances on each phone.

The Aura also seems to hold in reserve or disallow additional inbound calls on one of these appearances so that you will be able to transfer, conference, or place an outbound call from your phone even if you already have several active calls. So, the formula for the number of inbound calls your device can receive is equal to the number of call appearances minus one. This means that if you use the default of three call appearances per device, and you are talking to one person, and have another call on hold, you will not be able to receive a third inbound call, though you will be able to place one additional outbound call or perform a transfer or conference.

A third caller in this default configuration would be forwarded to the call coverage location specified in the Communications Manager system (typically the voicemail system).

This default can be modified, by editing the endpoint to support additional call appearances if desired. There is no special configuration required on the 84-Series phone to support this feature.

**To modify the default number of call appearances on your extension**

1. **Browse and login to System Manager.**
2. **Navigate To Elements > Communication Manager > Endpoints > Manage Endpoints**
3. **Find your extension number in the list and put a check in the box to the left of your name, then select the Edit Button**
4. **Navigate to the Button Assignment Tab**
   In the Main Buttons section each button that says *call-appr* should allow one call
   Since you can only potentially fill out six of these as *call-appr* buttons as long as you are configuring your phone as a 9620, you should be able to receive up to five calls and still place one outbound call if you make all lines a call appearance button
   If you have added additional *call-appr* buttons ensure you select the commit button when you are done.
Multiple Calls per Line Key or Maximum Calls per Line

The guidelines specified in the Call Waiting section above should cover Multiple Calls and Maximum Calls per Line Key.

Conference 3-way

In a three way conference, the 84-Series handset will take and merge the appropriate audio streams locally. No special treatment is required from the Avaya Aura system. It should be noted that if the 84-Series handset is the conference initiator and ends the conference by hanging up, the default behavior of the 8400 phone is to invite the other two conference participants to a peer to peer call. If it is desired to have all end points drop out of conference when the conference initiator leaves, administrators should invoke the following parameter:

call.transferOnConferenceEnd="0"

This parameter will cause all members of a conference call to be dropped when the conference initiator ends the call.
**Transfer: Blind**

This type of transfer occurs when Phone A calls Phone B and they are in call. Phone B then presses the **Features** fly out and selects the **Blind Transfer** button. This action places Phone A on Hold, and Phone B now dials the number for Phone C, followed by pressing the **Send** key (the key that looks like an off-hook). Phone B never talks to Phone C and Phone C begins ringing with the call from Phone A. If Phone C answers he will be in call with Phone A.

**Transfer: Announced**

This type of transfer occurs when Phone A calls Phone B and they are in call. Phone B then presses the **Transfer** button, placing Phone A on Hold, and dials the number for Phone C, followed by pressing the **Send** key. Phone C begins ringing with the call from Phone B, and if Phone C answers he will be in call with Phone B. Phone B can then “announce” that he is going to connect Phone C to Phone A. Phone B then presses the **Transfer** key again to complete the transfer. The result is that Phone C and Phone A are in call.

**Transfer: Attended**

This type of transfer is really a conference, where the conference initiator drops out of the call after the conference has been established and the two parties that were conferenced are re-invited to a peer to peer call. It should be noted that if the administrator chooses to configure `call.transferOnConferenceEnd="0"`, then this type of conference will not be possible.

**Caller ID**

Calling Party name and number are supported by the Spectralink 84-Series handsets. Additionally, the 84-Series handsets support the p-asserted identity header which allows the phone to use the PBX supplied messages to update the called and calling party names when re-invites occur.

**Hold and Resume**

Spectralink 84-Series handsets are capable of hold and resume. It may be noted, however, that the Avaya Aura system does not include the **sendrecv** message in the SDP of an initial SIP Invite message. As a result, the 84-Series phone attempts to revert to RFC2543 version of hold (referring held parties to the location 0.0.0.0). However, in our labs, the Aura system ignored this request and referred held parties to the system provided Music On Hold (MOH).
**Music On Hold**

Spectralink 84-Series handsets are capable of hold and resume, and in Spectralink’s lab environment, clients placed on hold by a Spectralink 84-Series handset were able to hear the system supplied (MOH) Music On Hold.

**Call Reject**

Call Reject allows a caller to decline an inbound call. When the 84-Series phone rejects an inbound call, it will by default send a 603 Decline message. In our labs, this resulted in a fast busy tone for any calling party. Use of the following parameter will cause the phone to send a 486 busy message when a call is declined instead, with the result that calling parties should hear the treatment defined by the call coverage location specified in the Communication Manager system (typically the voicemail system.) Spectralink labs recommends the following parameter for the optimal user experience:

```plaintext
voIpProt.SIP.use486forReject="1"
```

**Do Not Disturb**

Do Not Disturb was tested and found to work correctly on the 84-Series Handset using the **Settings > Feature Settings > Do Not Disturb** option available on the handset. However, it should be noted that this is not the equivalent of the Avaya Branded Do Not Disturb feature (which is set on the Callserver itself and typically allows Automatic Callback, Automatic Wakeup and Priority Calling type of calls.) For correct functionality of Do Not Disturb on the 8400-Series telephone the handset must be powered on and in range of the wireless network in order for it to immediately be able to respond to call invites with a 486 – Busy Here message. Otherwise, callers will receive the Communications Manager defined Call Coverage for the handset.

**Call Park**

Call Park and Retrieve were tested in Spectralink’s labs using Feature Access Codes (FACs) to park and retrieve calls. Please consult the **FACs** section of this manual for assistance in programming a directory file that will pre-populate user’s phones with useful System Features and Access codes.

**DTMF via RFC2833**

The Spectralink 84-Series handset utilizes RFC2833 in order to support delivery of DTMF tones. There is no special configuration required in order for the handset to utilize RFC2833, and RFC2833 was verified to function correctly in the course of Spectralink lab testing through the
manipulation of Communication Manager Messaging menus and trunk calls to PSTN IVR services.

Call Forward

The Spectralink 84-Series handset was tested using the Call Forward All functionality. The following two methods for implementing this feature were both tested in Spectralink’s labs:

Call Forward All Calls using the handset

This method of implementing Call forward was tested by navigating to the Home > Settings > Feature Settings > Forward > Always menu and specifying a forward destination, and then enabling the feature. One advantage to this method is that it posts a user friendly indication on the home screen that the phone is in the forwarding state any time you have enabled call forwarding.

The disadvantage to this method of call forward implementation is that the phone must be powered on and connected to the WLAN in order to successfully redirect any offered calls to the Call Forward All destination. So a user that set call forward and then powered the handset off would not, in fact, still be forwarding calls.

Call Forward All Calls using Feature Access Codes (FACs)

This method of implementing call forward was tested by dialing the Call Forward All Calls and Call Forward Cancel FAC’s programmed in the Communications Manager. One advantage to this method of implementing call forward is that the forward remains in effect regardless of whether the handset remains powered on or in range of the wireless network.

The disadvantage to this method of call forward implementation is that the phone does not provide any user friendly indication that the phone is in the forwarded state when call forwarding is set. The call forwarding state is maintained by the Communications Manager itself and the Aura system will simply never offer calls to the phone until the call forward is cancelled using the Call Forward Cancel FAC.

The feature access codes pertaining to our site specific implementation are contained in the sample 0000000-directory.xml file contained in this document. If the site administrator would like to offer this capability, please replace the FAC’s contained in our sample configuration file, with their site specific Feature Access Code.

Call Forward No Answer

Call Forward No Answer was also tested and verified to work, though it is expected that this would typically be programmed in the Endpoints’ configuration settings on Communications Manager rather than implemented on the handset. Otherwise, the Forward After Rings value on
the 84-Series handset must be set to a value that will be less than the number of rings the Communications Manager is configured to use before providing Call Coverage.

As in Call Forward All Calls, the handset must also be powered on and connected to the WLAN in order to successfully redirect any offered calls to the Call Forward No Answer destination.

**Call Forward Busy**

Spectralink recommends Call Forward Busy destinations be handled by the Communications Manager rather than on the handset. Spectralink 84-Series phones are capable of receiving up to 24 calls per registration so calls to a registered extension will be deemed “busy” by the Communications Manager well before this capacity is reached on the handset.

**Feature Access Codes**

Feature Access codes were utilized in Spectralink's labs for the testing of several features (Call Forward and Call Park to name a few.) First, Spectralink recommends the use of a 000000000000-directory.xml file with useful feature access codes pre-programmed for users’ benefit. A sample of this style of configuration file can be found in Appendix A—the sample Configuration Files section of this manual.

Also worth mentioning is the fact that the default dial plan the Spectralink 84-Series phones utilize prevents phones from dialing numbers that start with either a pound # or an asterisk *, which many sites may utilize in their FAC dial plan. Dial plans may be quite elaborate. In our sample configuration file, we null out the entire dialplan in order to avoid this pitfall. If your deployment is experiencing issues dialing numbers that begin with a * or a # sign, you may wish to try the following two parameters to rule out the dial plan as a possible culprit. If this resolves the issue, a more detailed dialplan analysis may be desirable to optimize your deployment.

```
  dialplan.digitmap=""
```

This parameter “nulls out” the existing dialplan, such that you should be able to dial any number with no dial plan timeouts.

```
  dialplan.impossibleMatchHandling="2"
```

This parameter tells the phone to wait for the user to hit the green send key after dialing before sending any value that is not a part of the current dialplan to the server for immediate processing. Since we have deleted the entire dialplan in the preceding parameter, the phone would otherwise consider every string dialed “impossible”, and send it immediately.

**SIP Using TCP**

There are two methods for accomplishing this and they correspond to the two methods mentioned in the Direct to Avaya Aura SIP Registration section.
Method 1
This method involves adding an additional line to the configuration file that will specify that the phone should utilize the transport type of TCP for SIP messaging. It can be utilized when specifying an ip address or a DNS A-Name record for the reg.1.server.1.address. The configuration parameter that would be utilized to accomplish this is:
  \texttt{reg.1.server.1.transport=TCPOnly}

Method 2
This method utilizes a DNS SRV record to specify the transport type of UDP. See Appendix B for more details regarding this method.

G.711u, G.711a, G.729A, and G.722 Codecs
The Spectralink 84-Series handset was tested using each of the above codecs when deployed against the Avaya Aura system. Some configuration of the Communication Manager’s settings may be required to achieve the G.722 codec if desired as described below.

Default 8400 Advertised Codec List
Typical deployments should not require modifications to the Spectralink 84-Series phones’ default list of advertised codecs, which are shown below in the preconfigured, default order of advertisement:

1) G.722-64K \texttt{voice.codecPref.G722=4}
2) G.722.1-32K \texttt{voice.codecPref.G7221.32kbps=5}
3) G.711Mu \texttt{voice.codecPref.G711_Mu=6}
4) G.711A \texttt{voice.codecPref.G711_A=7}
5) G.729A \texttt{voice.codecPref.G729_AB=8}

Codec selection on the Aura system
If G.722 is desired and is not being established, customers may need to modify the ip-codec-set their phone is utilizing. To do so, they will need to verify which ip-network region their phone is a member of, then determine which codec-set that ip-network region uses. The below two sections detail the steps that would allow you to determine this:

1 Identify the ip-network region
   a Connect (SSH) directly to the Communications Manager
   a From the System Administration Terminal (SAT) enter the command \texttt{change ip-network-map}
2 Identify the Codec Set
   a From the SAT, enter the command `change ip-network-region x` (where x is the ip-network region you identified for your phone in the step above)
   b Look for the field **Media Parameters > Codec Set** (This is the codec set your phone is using)

3 Changing or Viewing the Supported Codec Set
   a From the SAT, enter the command `change ip-codec-set x` (where x is the ip-codec-set you identified for your phone in the step above.)
   b In the form that appears you should see a numbered list that will allow you to enter codecs in the order that you prefer them (i.e. 1 is preferred, 2 is next etc.) If you wish to utilize G.722, specify G.722-64K at the top of the list. (Aura requires high bandwidth codecs to be listed first). If some devices in the network region don’t support the G.722 codec, it should not present an issue. As long as devices don’t advertise G.722 support they will simply revert to one of the G.711 variants or G.729 as long as those codecs are also listed somewhere in the supported list for the ip-codec set.

Below is an example of the ip-codec-set configured to prefer G.722-64K
Note: Spectralink labs has also observed that when selecting G.722 audio the Session Manager will start out by forcing G.711 back to the Communications Manager, and then after several seconds send an invitation allowing a peer to peer G.722 conversation between the G.722 enabled endpoints.

Verifying the Codec the handset is using

The codec established by the 84-Series phone may be verified through the 84-Series phone’s menus. Keep in mind that if you wish to achieve g.722 the other endpoint must also be capable of supporting this codec, such as another Spectralink 84-Series phone. To verify the codec established, place a call to another endpoint. While in call, on the 8400 phone, navigate to Settings > Status > Diagnostics > Media Statistics. This should display the codec that is currently in use.

![Media Statistics](image)

**Multiple Line Keys (or Registrations) per Handset**

This is configurable by building each line (or registration) as a separate phone and user in the Aura system. The phone should then be configured to register to both of the Endpoints / User Profiles that were built on the Aura system. From the Aura’s standpoint, the downside to this method is that each line will consume a separate Off-PBX Telephones – OPS license.

Below is a configuration example that would result in the Spectralink 84-Series phone having two registrations (4615 and 4616), and two line appearances shown on the phone. Keep in mind that each line appearance or registration could have multiple calls (by default 3) depending on how many line appearances you configured on the Endpoint in Communications Manager.

```<TelephonyLine1`
reg.1.server.1.address="172.29.102.92"
reg.1.address="4615@engr.local"
reg.1.auth.password="4615"
reg.1.auth.userID="4615"
reg.1.label="4615"
reg.1.displayName="4615"
msg.mwi.1.subscribe="4615"
reg.2.server.1.address="172.29.102.92"
reg.2.address="4616@engr.local"
reg.2.auth.password="4616"
reg.2.auth.userID="4616"
reg.2.label="4616"
reg.2.displayName="4616"
msg.mwi.2.subscribe="4616">
</TelephonyLine1>
Admin Tip: How many devices can share?

Avaya allows up to ten devices. However, the maximum number of calls you may receive on this number is limited by the number of line appearances on the device associated with the User Profile.

For clarification, if you use the system default of three call appearances (call-appr's) per device, but program four phones to “share” this User Profile, you will still only be able to receive two calls to this number and place one additional outbound call from this number, regardless of how you distribute those calls across the four phones. So, if you wish to “share” a User Profile among multiple phones it may be advisable to add additional call-appearance buttons to the device associated with the User Profile in order to allow a larger number of inbound calls to the same number simultaneously. (See the Call Waiting section for a more in-depth discussion of this topic.)

Programming the 84-Series Phone to Support the Shared User Profile

The simplest version of this would be that all phones have only one extension associated with them, and it is the Shared User Profile’s Extension. In this design, callers dialing the Shared User Profile’s extension number would ring all of the Phones registered to that particular extension, and any call initiated from any of the phones registered to this number would show that they originated from the “Shared” number. For this case, the phone does not require any programming differing from that of a normal registration. We would just create multiple phones using the same registration parameters and credentials. The “Shared” portion of the User profile is all handled by the Avaya Aura system itself.

Another method for implementing this functionality would be to give each phone its own, unique number or registration, and then have a second number or registration that is the Shared User Profile number on each phone. The phone side configuration for this would look exactly like the configuration described above in the Multiple Line Keys (or Registrations) portion of this manual. The Aura systems tracks each registration from the phone (reg.1 and reg.2) as a separate device, and the phone maintains two, active registrations with the server in this case. Users can select the second Line, or Shared User Profile number from the Features > Lines menu for outbound call initiation on the second number, and inbound calls on the “Shared” number will simply be offered to the user as a normal call without the need to manually select the second line in order to answer.

Trunk Calling

In and outbound trunk calling were tested utilizing an ISDN PRI circuit connected to a MM710 Interface card in a G450 Gateway. The Spectralink 84-Series handset was able to make and receive calls through this configuration as well as to pass DTMF digits through to IVR style
menus on the PSTN. There is no special configuration on the 84-Series handset required to allow this functionality.

**Integration with 46xxsettings file**

Spectralink 84-Series handsets do not utilize the 46xxsettings file. Spectralink recommends all handsets use a separate provisioning server for delivery of configuration files and management of phone logs. This server can be configured to utilize TFTP, FTP or FTPS, HTTP, or HTTPS. For more details regarding provisioning server deployment and requirements please reference the *Spectralink 84-Series Deployment Guide*.

**Personal Profile Manager Integration**

Spectralink 84-Series handsets do not support PPM commands. However, some PPM functionality may be implemented in other manners using existing Spectralink 8400 configuration parameters. For example, Spectralink strongly recommends Provisioning Polling be enabled, (prov.polling.enabled="1"), such that Spectralink handsets will look for config file changes and software updates automatically on the central provisioning server at specified time intervals. In this way, configuration changes made in the Spectralink 84-Series’s config files that correspond to configuration changes on the Avaya Aura system will be made automatically by the handsets when they poll the central provisioning server. For more details regarding automatic provision polling please reference the *Spectralink 84-Series Administration Guide*.

**Avaya Presence**

The Spectralink 84-Series handset does not currently support Avaya branded Presence functionality.

**Failover / Fallback / Redundancy / Resiliency**

Although not tested with the Avaya Aura system in Spectralink’s laboratory, Spectralink 84-Series phones do support a number of mechanisms for failover and fallback implementation. Therefore it is likely that handsets may be able to utilize a secondary Session Manager in the event of a Primary failure. This capability could easily be tested in a field situation.

**SRTP**

SRTP on the 84-Series phones was not tested in conjunction with the Avaya Aura system.
**TLS**

TLS on the 84-Series phones was not tested in conjunction with the Avaya Aura system.
Chapter 4: Troubleshooting and Analyzing SIP on the Avaya Aura System

How to gather SIP traces on the Aura system through System Manager

The Maintaining and Troubleshooting Session Manager doc available on Avaya’s support site should contain much more detail about how to manage the integrated SIP Tracer, and how to filter and view any traces it collects. That said, below we’ll provide an example of how you might access and use this tool to gather more information if you are experiencing an issue you think may be SIP related. First we will demonstrate how you might gather typical SIP traffic to or from a given endpoint and second, we’ll show a phone with a failing (dropped due to firewall rules) SIP Invite. We staged this example by implementing a Firewall rule that would not allow the endpoint to send more than three invite packets.

To access the SIP Tracer:

1. Log Into System Manager and Navigate to Elements > Session Manager
2. Select the System Tools > SIP Tracer Configuration menu option
3. On the Tracer Configuration screen, navigate to the bottom and place a checkmark next to the Session Manager of interest under the Session Manager Instances header (in our case, polycom-asm)
4. Select the Read button, this should give you the current Tracer Configuration
In a lab environment, or if there are only a few SIP clients you might be able to use the Trace All Messages configuration option, but on a busy system, you would likely want to implement a filter. For our example, we will implement a filter based on the SIP client's ip address and capture messages both to and from the client. To implement a filter such as this and capture only these messages, ensure the following fields are set:

a. Tracer Enabled needs to be checked

b. Select All Of the other Message types to Trace

c. Under the User Filter heading select the New Button
   i. Place a checkmark next to the empty box that appears
   ii. In the Source column, enter <your phones' ip address>
iii  Change the **Max Message Count** to a value that will be reasonable for your experiment (for a reproducible problem 100 messages is likely enough)

d  Still under the **User Filter** heading select the **New** button again

  i  Place a **checkmark** next to the empty box that appears

  ii  In the **Destination** column, enter `<your phones’ ip address>`

  iii  Change the **Max Message Count** to a value that will be reasonable for your experiment (for a reproducible problem 100 messages is likely enough)

e  Select the **Commit** key

5  Now, run the experiment you are interested in and note the time of day

6  Then, Navigate back to the Sip Tracer Configuration screen and **Disable the Tracer**
Next, using the left hand navigation menus navigate to **Session Manager > System Tools > SIP Trace Viewer**

a. Select the small button to the right of the word Filter and narrow down your timeframe.

b. Then, place a checkmark next to the Session Manager's name: (in our example this is named polycom-asm)

c. Select the **View** button
You should now be able to see the SIP packets coming to and from Session Manager. If you select the radio button next to one and then select the Show button you can also see the details.
The trace may also be exported from this window should packet analysis be required. Note that the export is in a text document format (not a Wireshark trace), but relevant SIP details should be included.

The Viewed Trace may also be post processing filtered, for particular packet types by using the Filter: Enable button on this screen.

Dropped packets due to firewall failures

We also intentionally implemented a Firewall rule that would cause the Session Manager's Firewall to drop SIP Invites from our client in an attempt to illustrate what this might look like in the SIP Trace Viewer. Below we'll describe how you might look for these. This assumes that you have already captured a Trace through the Session Manager Tracer as described above and are in the Trace Viewer.
If you used a filter to capture your trace, it will likely be relatively easy to find the dropped packets, but if you did not, you could:

1. Select the Filter: Enable field

2. In the Action Box that appears, enter the term **Dropped**

   The Session Manager Trace Viewer should show you all packets that were Dropped.

   The Messaging inside this packet will show why the packet was dropped (In our below example the packet was dropped by the Session Manager’s Firewall because we triggered the Firewall Rule named BILLC_TEST_DROP)

If the customer is experiencing dropped packets an attempt should be made to determine whether the existing Firewall rules seem reasonable.
Working with the Aura Session Manager’s Integrated SIP Firewall

The Avaya branded *Security Design in Avaya Aura Session Manager* document should provide a much more in depth view of the Session Manager's integrated firewall, however, if the Spectralink phone is not behaving as expected it is probably worth at least a quick look at the Avaya Session Manager’s integrated Firewall to ensure that it is not dropping packets from the Spectralink phone.

A new install of Session Manager would come with default Firewall rules enabled. However, Avaya implemented default Firewall rules that were much more stringent in the Session Manager 6.2 release, and then relaxed the default rule set in later releases of Session Manager 6.3. So, if the customer implemented Session Manager when the defaults were more stringent, those defaults would likely be preserved as part of any upgrade to a later Session Manager release, with the result that a customer might have active rules blocking SIP packets. Typically a default set of rules reflecting what was recommended as a part of later Session Manager releases would also be placed on the system (though not activated) when an upgrade occurs, and you may wish to consult those rules as a comparison to determine if any failures that may exist are indeed reasonable. If in doubt, please consult your Avaya Reseller. You may also wish to reference Avaya PSN004136u.

In order to see whether SIP Packets may be getting dropped due to existing Avaya Session Manager Firewall rules perform the following action:

1. Browse to and login to System Manager
2. Navigate to Elements > Session Manager
3. Navigate to System Status > SIP Firewall Status
4. Select the Session Manager of Interest by placing a checkmark in the box next to it.
5. Then select the arrow next to the Details > Show field

Now, in order to illustrate how the SIP Firewall could potentially block SIP packets from a device, in the below example we created a Firewall rule that would rate limit SIP Invites from a specific endpoint using the endpoints' IP address.

The test rule we created is called BILLC_TEST_DROP, and it was configured to drop any SIP Invites that exceeded the threshold of three Invites in any 20 second period. All we had to do was place a call and then attempt to immediately place another to trigger the rate limiting functionality set by this rule. Below we can see that there were 8 packets matching this rule, and five were dropped using the Rate Limit option.
If your deployment shows packets in the dropped column, you may wish to reset the SIP Firewall counters using the **Reset** button on this page, refresh the browser’s cache (If you do not see the values reset), and reattempt your experiment. Then, refresh the browser again and note any Firewall Rules that show packets were dropped. Dropped packets may result in aborted, incomplete, and unreliable call functionality, and are not necessarily indicative of a problem with the handset. Consult Avaya PSN004136u for additional recommendations as to how to proceed or modify the existing firewall should this occur. The existing firewall rules may need to be modified or relaxed so they are no longer rate limiting. On the other hand, they may help guide you to an interoperability issue.

Firewall Rules can be configured and manipulated by Navigating to **Session Manager** => **Network Configuration** => **SIP Firewall**

The Rule Set that shows in the Assigned Count column is the Active Rule Set and it can be edited by placing a checkmark in the box next to the Rule Set of interest and then selecting the **Edit** button.
We may then edit the specific rule of interest by selecting the checkbox next to the rule and selecting the **Edit** key. If the rule is currently causing packets to be dropped we might also deselect the Enabled box, and commit the rule set to determine if this is the cause of a failure or problem we might be experiencing.
Below is the rule of interest in our configuration. We might wish to change the **Action Type** to **None** or relax the threshold if we decide this rule is blocking or impacting functionality. Don’t forget to **Commit** any changes you might make.
If further firewall investigation is desired, obtaining a SIP trace of the endpoint and examining the Session Manager’s Firewall logs will also be helpful. Logs for the firewall can be found in the /var/log/Avaya/asset/asset.log directory on the Session Manager, or you may use the System Manager’s Log Harvester to process these.
Verifying the Registration Status of the 84-Series Handset on the Session Manager

Earlier we discussed how to view the registration status according to the 8400 handset. We may also wish to determine what the Avaya Session Manager shows as the registration status for our device.

To view registration status

1. Login to System Manager and navigate to Session Manager > System Status > User Registrations
2. Now, find your endpoint and select the arrow next to the Show field in the Details column. Experiment with the Views to provide additional detail.

DSCP Values

The default DSCP values for Call Control and Audio are typically aligned with Spectralink recommendations. That said, if wireless analysis determines that packets are not “getting through” to the handset, it may be worth verifying that the Aura system is tagging Audio and SIP Control packets with appropriate DSCP values.
Call Control DSCP

The system default Call Control value for SIP packets sent by the Session Manager appears to be 46. This is typically a voice priority and may be overkill for signaling, but on the other hand, at least we know these packets should be getting delivered.

To modify or check the Call Control DSCP Value,

1. Log into System Manager, then navigate to: Services > Inventory > Manage Elements
2. Place a check mark next to the Element that is the Session Manager and select the Edit key
3. Next, modify the Call Control PHB value to reflect the Decimal DSCP value you would like the call control packets to be tagged with and select the Commit button.
Note that the Communications Manager Network Region Settings screen (where we modify the Audio DSCP values) also contains a field that looks like it would allow you to modify the call control value. However, the Communication Manager does not send SIP messages directly to our phones, the Session Manager does. The Communications Manager does send audio to our phones though, so we can modify the audio DSCP values the PBX sends us in the Communications Manager itself.

The call control setting we see in the Communications Manager Network settings will control messages sent to legacy CCMS phones, but the SIP Call Control DSCP values must be modified separately as described above.

### Audio DSCP

The Aura system default value for audio DSCP seems to be 46. This is typically a good value to achieve wireless prioritization of voice packets for Spectralink wireless phones. However, if you suspect this value has been modified or need to check it, then follow the steps below:

**To check the DSCP value**

1. Identify the Network Region your phone is a member of.
If you are uncertain, you should be able to determine this by logging into the Communications Manager directly and issuing the command `change ip-network-map`. This would provide information allowing you to correlate the network region you are in based on your sets' current IP address. If you don't see your range, then you will typically default to region 1.

2. Now, back in the System Manager, navigate to **Elements > Communications Manager > Network > IP Network Regions**

3. Now, select the Region your phones are a member of and select the Edit key.

4. Next, modify the Audio PHB value to the value you desire the PBX to send. Note that in a SIP peer to SIP peer call, audio will typically be redirected such that it will all flow peer to peer and only use the value delivered by the far end, but, for the first few milliseconds of a call (before the audio is rerouted to flow peer to peer), or on a trunk to SIP, Digital to SIP, Analog to SIP etc. call, the audio will come from the Communications Manager and will utilize the value you specify below:
Sample Configuration Files

The following are the configuration files utilized in our lab experiments. Your configuration and specific values will differ, however these files may be used as a reference and as a starting point for a deployment of the Spectralink 84-Series phones to be integrated in an Avaya Aura Communications Manager and Session Manager environment. For our purposes we modified the template files for a Flat Deployment. Sample templates may be found in the Config => Scenarios => Flat_Deployment folder provided with each release.

000000000000.cfg

This is the file that each handset will use to determine the correct software load and other supporting config files

```xml
<?xml version="1.0" standalone="yes"?>
<!--
*************************************************************************
**
-->
<!-- * Default Master SIP Configuration File for FLAT DEPLOYMENTS
* -->
<!-- * -->
<!-- * This file is read by every handset at boot to specify where to load its -->
<!-- * software from (APP_FILE_PATH), to specify the sequence in which to read-->
<!-- * configuration files (CONFIG_FILES) and whether to store or retrieve -->
<!-- * information in specific directories (server root by default) -->
<!-- -->
<MASTER_CONFIG>
<!-- You can specify a path with subdirectories to specify the location of the handset software. -->
<!-- See the Deployment Guide for more information about setting up subdirectories. -->
<SOFTWARE
    APP_FILE_PATH="sip.ld"
/>
<!-- Information from files on the left overrules information from files to their right -->
<!-- [PHONE_MAC_ADDRESS] dynamically gets replaced by that handset's MAC address -->
```
<!-- For FLAT DEPLOYMENT, create a <macaddress>-ext.cfg for each handset following the -->
<!-- template provided in this directory -->
<CONFIGURATION>
  CONFIG_FILES="[PHONE_MAC_ADDRESS]-ext.cfg, site.cfg"
</CONFIGURATION>

<!-- DIRECTORIES -->
LOG_FILE_DIRECTORY=""
OVERIDES_DIRECTORY=""
CONTACTS_DIRECTORY=""
CALL_LISTS_DIRECTORY=""

</MASTER_CONFIG>

site.cfg

This is the file intended to provide parameters to all handsets at the site

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<handsetConfig xsi:noNamespaceSchemaLocation="handsetConfig.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <!--site.cfg template for FLAT DEPLOYMENT-->  
  <SystemParameters>
    <!--Unless otherwise specified, all values are recommended settings and do not need to be changed.-->
    <log>
        log.render.file.upload.period="3600"
        log.render.file.size="128"
        log.render.file.upload.append.sizeLimit="10000"
        log.render.level="0"
        log.render.stdout="0" />
    <!--The syslog server name can be an IP address or fqdn (fully qualified domain name)-->
    <prov.polling>
      prov.polling.enabled="1"
      prov.polling.mode="abs"
      prov.polling.period="86400"
      prov.polling.time="03:00"
      prov.polling.timeRandomEnd="/"
    </prov.polling>
  </SystemParameters>
  <TelephonyParameters>
  <openSIP>
    <SIPserver>
      reg.1.server.1.address="172.29.102.92"
      reg.1.server.1.port="5060"
      reg.1.server.1.expires="300" />
  </TelephonyParameters>
</handsetConfig>
Spectralink 84-Series Wireless Telephones and Avaya Aura Communications Manager and Session Manager: Interoperability Notes

dialplan.digitmap=""  
dialplan.impossibleMatchHandling="2" />
</DND>
voIpProt.SIP.use486forReject="1" />
</voicemail>
up.oneTouchVoicemail="1"
msg.mwi.1.callBackMode="contact"
msg.mwi.1.callBack="4999"
np.normal.alert.messageWaiting.tonePattern="silent">
</voicemail>
</openSIP>
</TelephonyParameters>
</handsetConfig>

00000000000-directory.xml

This file can be used to pre-populate useful Feature Access codes

00000000000-directory.xml

Spectralink labs recommends that this file be utilized to pre-populate useful Feature Access codes such that users may quickly find these Features available under either the Favorites or the Features => Lines Flyout menus. This will help alleviate the need for users to memorize Feature Access Codes. The values shown for the Feature Access Codes below would need to be modified to match your deployments’ dial plan.

<?xml version="1.0" standalone="yes"?>
<directory>
  <item_list>
    <item>
      <fn>Call Park</fn>
      <ct>*22</ct>
      <sd>1</sd>
    </item>
    <item>
      <fn>Call Park Retrieve</fn>
      <ct>*23</ct>
      <sd>2</sd>
    </item>
    <item>
      <fn>Call Forward</fn>
      <ct>*71</ct>
      <sd>3</sd>
    </item>
    <item>
      <fn>Call Forward Cancel</fn>
      <ct>*72</ct>
    </item>
  </item_list>
</directory>
This file contained the phone specific registration parameters for the first of our three test phones.

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<handsetConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="handsetConfig.xsd">
<!-- * Sample per-phone Configuration File for FLAT DEPLOYMENTS * -->
<LineRegistration>
    <TelephonyLine1
        reg.1.address="4615@engr.local"
        reg.1.auth.password="4615"
        reg.1.auth.userID="4615"
        reg.1.label="4615"
        reg.1.displayName="4615"
        msg.mwi.1.subscribe="4615">
    </TelephonyLine1>
</LineRegistration>
</handsetConfig>
```

This file contained the phone specific registration parameters for the second of our three test phones.

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<handsetConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="handsetConfig.xsd">
<!-- * Sample per-phone Configuration File for FLAT DEPLOYMENTS * -->
<LineRegistration>
    <TelephonyLine1
        reg.1.address="4616@engr.local"
        reg.1.auth.password="4616"
        reg.1.auth.userID="4616"
        reg.1.label="4616"
        reg.1.displayName="4616"
        msg.mwi.1.subscribe="4616">
    </TelephonyLine1>
</LineRegistration>
</handsetConfig>
```
This file contained the phone specific registration parameters for the third of our three test phones.

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<handsetConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="handsetConfig.xsd">
  <!-- * Sample per-phone Configuration File for FLAT DEPLOYMENTS * -->
  <LineRegistration>
    <TelephonyLine1
      reg.1.address="4617@engr.local"
      reg.1.auth.password="4617"
      reg.1.auth.userID="4617"
      reg.1.label="4617"
      reg.1.displayName="4617"
      msg.mwi.1.subscribe="4617">
    </TelephonyLine1>
  </LineRegistration>
</handsetConfig>
Appendix B

Creating a DNS SRV record in a Windows Server

While we used a Windows 2003 Server for this example, the steps for a Windows 2008 Server would be very similar. There are MANY different settings and options we might set in a DNS server as we create a new SRV record that corresponds with our SIP Domain name. It is also possible that the DNS Forward Lookup Zone defined for the SIP Domain may already be defined. We'll provide an example of creating a new SIP Domain (Forward Lookup Zone) in the DNS server from scratch, but if the Domain already exists you should be able to simply skip past the step where we create the original SIP Domain and follow the rest of the steps in sequence.

On the DNS Server, navigate to the Forward Lookup Zones and look for a Zone with the name of your SIP Domain, if it already exists, skip down to the “Create a DNS A Name record for the Callserver” section of this document. Otherwise follow the steps below to create a new Forward lookup zone using your SIP Domain name:

Create a New Forward Lookup Zone using the SIP Domain Name (or Verify a Zone Exists with your Domain name)

1. Right click on the Forward Lookup Zones field, and select New Zone
2 Select Next at the Welcome to the New Zone Wizard page

3 We chose to create a Primary Zone, and selected Next, and also selected Next on the following page, using the default of: To all DNS servers in the Active Directory domain “specified Active Directory Domain”.

4 In the Zone Name field, type the name of your domain (In our example mydomain.local), and then select Next again:

5 On the Dynamic Update page, we chose to allow only secure dynamic updates, but you would need to do this in accordance with your local policy. Then select Next again, and finally, select Finish.
Now, you should see the domain name you just created (i.e. mydomain.local) appear as a Forward Lookup Zone:

Create a DNS A Name record for the Callserver (or Verify an A name record already exists for the Callserver)

On the DNS Server, navigate to the Forward Lookup Zone that corresponds to your SIP Domain and highlight it. Now look for an A name record that corresponds to the Callserver the phones will point to (one does not exist in the example shown below). If a host A Name record already exists, skip down to the “Create a SRV record for the Callserver” section of this document. Otherwise follow the steps below to create a new DNS A name record for your Callserver.
1. Right click on the Forward Lookup Zone you just created and select New Host (A)
2  Now enter the Callserver’s name in the name field. If you are not particular about the name it can be named anything that would help to identify it as the SIP PBX. We will simply use this record to help us DNS resolve the Callserver’s IP Address so you should not need to worry about giving it the “wrong” name. For our example, we called the PBX *mycallserver*. You must also provide the PBX’s IP address. This is the address SIP Registrations and Invites should be directed to. Then select *Add Host*.

![New Host dialog box](image)

Note: If you receive an error message about the associated PTR record NOT being created, it simply means the reverse lookup zone is not specified. If the customer is not familiar with how to resolve this error or is not concerned about it, it can be safely ignored.

3  Select *Done* to finish creating A name Host records. You should now see an A name record linking the Callserver’s name to its IP address:
Create a SRV Record for the Callserver

1. Right click on the forward lookup zone you created that corresponds to the SIP Domain and select *Other New Records*
2. In the Resource Record Type window, scroll down to the **Service Location (SRV)** field and highlight it, then select **Create Record**

In the New Resource Record Window, we will create a record that will point to our server, **mycallserver**, and program it to use the udp protocol, on port 5070. To do so, modify the following fields:

**Service:** `_sip` (you will need to type this in.....it is not available in the pull-down)
Protocol: _udp

Priority: 1 (This tells us that this server will be the first item in the list, we may implement support for more than one server / failover in the future)

Port Number: 5070 (Recall we did NOT specify this in the 8700 port field)

Host Offering This Service: mycallserver.mydomain.local

3 Select the OK key

4 Select Done to close the Resource Record Type Window that remains.

Now, we should see a SRV record that points to our Callserver’s name using port 5070 in the Forward Lookup Zone that corresponds to our SIP Domain.
DNS Verification and Troubleshooting

Once you've built the SRV record and A name record on the DNS server, you might want to ensure that you can query them. You can use a regular Windows machine to do this, but remember, it must be pointed to the DNS server (or have access to the DNS Server) where you created the records.

To Verify the A Record for mycallserver.mydomain.local

1. From a Command Prompt type: nslookup mycallserver.mydomain.local
2. To Verify the SRV record for mydomain.local
   a. Type nslookup and press return
      i. Type: set type=srv
      ii. Type: _sip._udp.mydomain.local
   b. The Output from the above tests should look something like the below. If not, the phone won't work either: