Technical Bulletin CS-14-17

84-Series Failover Configuration Using DNS

This technical bulletin explains how to setup your Windows DNS environment to support failover with the 84-Series Handsets.

System Affected

Spectralink 84-Series Handsets

Description

Spectralink 84-Series Handsets support failover between two or more SIP servers natively. However, if your SIP environment uses active-active failover, meaning that your SIP servers are aware of each other and keep track of client states, then you will want to use DNS to configure your failover. The default failover behavior for the 84-Series handset assumes that the SIP servers are setup as active-passive, meaning that when one SIP server is active the other is in a standby state and unaware of the client registration of the other server.

You can use the default failover configuration for an active-active environment without configuring DNS but be aware that doing so means that the handset will attempt to register to both servers at the same time. Some SIP environments do not handle this well and can cause the SIP clients to flip-flop between the two servers constantly. This works fine for the active-passive environment as the passive server will accept the client registration but won't do anything else with it.

By using DNS to handle the failover you are allowing the 84-Series handset to learn about an array of SIP servers that can be used for registration but you are also feeding it the information about which servers are considered primary, secondary, tertiary, etc. This is important because it allows the phone to register to one server at a time instead of all of them at once.

This document will explain how to perform this setup with Windows DNS but it could be done with any DNS server of your choosing. Be aware that NAPTR records are not support in all DNS servers and that may cause issues. With Windows DNS, NAPTR records are only supported when using Windows 2008 R2 Server. Fortunately there are some alternative methods to make this work without NAPTR records which we'll describe. So let's get started!

DNS Configuration

Step 1

There are a couple ways to set up your DNS for failover support of the 84-Series handsets. We'll document both here and you can decide which one you want to use. But here's a little background on how the 84-Series phone handles DNS resolution for this process.

If a DNS name is given for a proxy/registrar address, the IP address(es) associated with that name will be discovered as specified in the RFC 3263. DNS record lookup behavior may additionally be influenced if the transport mechanism or port number are specified in the phone's configuration. If the phone is allowed to operate using the default configuration parameter combination of transport method (NAPTR) and port number (0), the handset will attempt record lookups in the following order until it receives a successful lookup response; NAPTR, then SRV, followed by A record.

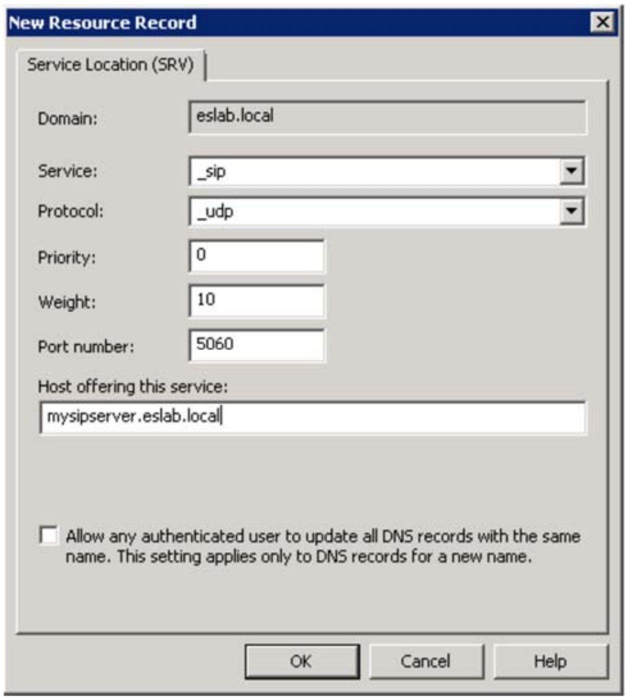
If a transport method is specified to be something other than DNS NAPTR, the phone will skip the attempt to query for a DNS NAPTR record, and will instead attempt a SRV record lookup for the proxy/registrar address using the transport mechanism specified, followed by an A name record lookup if the SRV record lookup fails. If a port number is specified via the configuration parameters, the phone will not attempt NAPTR or SRV record lookups. The only lookup will be an A record. If no port is specified via configuration parameters, and none is found through DNS mechanisms, port 5060 will be used by default for TCP or UDP transport types, and 5061 will be used as the port number for TLS. See RFC 3263 for an example.

Option 1 – Using SRV records and A records only

This first option uses your existing domain as the root for server lookup of the required records. Using this option leverages the already existing NAPTR records that are created as part of an existing domain.

In the domain you wish to use for failover, you will need to open your DNS server management window and right-click to select "Other New Records". In the window that opens, scroll down until you find Service Location (SRV), note that the list is in alphabetical order.

Select the SRV record type and click the Create Record button. This will open another window where you can create the SRV record and enter the details as shown in the image to the right.



Domain: eslab.local

Service: _sip

Protocol: _udp

Priority: 0

Weight: 10

Port number: 5060

Host offering this service: mysipserver.eslab.local

☐ Allow any authenticated user to update all DNS records with the same name. This setting applies only to DNS records for a new name.

You will enter the Service field as _sip which will denote the service type being used is SIP. The Protocol will be either UDP or TCP, depending on what you're deploying. Note that you can create records for both protocol types if you like. Just repeat these steps and enter the appropriate protocol. The Protocol is entered as _udp or _tcp.

Next is the Priority and this one can be a little confusing at first. A lower priority number means the record has higher priority over other entries with the same name. For example, if I have two SIP server, which you should since you're doing this setup, you will need an SRV record for each of the servers. This means that the server you want to be server 1 should have a priority number that is lower than the value you give to server 2.

The Weight is similar to Priority but it uses of higher number to imply importance over another record. If you have two SRV records that have the same priority then the weight will be used to determine the server that gets used.

The port number is required and will typically be 5060 unless you've configured your SIP environment to use a non-default port. An Avaya CS1K SIP environment uses a default of port of 5070, for example.

The Host Offering This Service field is where you will enter the FQDN of the SIP server. This means you must have the domain name appended to the end of the server name you enter. If you haven't created an A record for your SIP server yet, don't worry, we're getting to that bit.

Once you're done, just click OK to save the record. You will need to repeat these steps to add any additional SRV records for other SIP servers. You **MUST** have one SRV record per SIP server.

Now, what you would put into the handset for the SIP server address would just be the domain name. I know this seems a little odd but what happens is the phone will use this information to do a lookup for NAPTR records and it knows how to find the SRV and A records from there.

Here's an example of the XML you would include in your phone configuration file:

```
<voIpProt>
   <voIpProt.server
```

```
        voIpProt.server.1.address="eslab.local"
        voIpProt.server.1.port="0"
        voIpProt.server.1.transport="naptr"
        voIpProt.server.1.expires="3600"/>
    <voIpProt.SIP
        voIpProt.SIP.enable="1"
        voIpProt.SIP.use486forReject="1"/>
</voIpProt>
```

You'll need to replace the server address with your appropriate address and depending on how you've chosen to deploy, using NAPTR records or SRV and A records only, will determine what you'll set the transport type will be. Your options are: DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly.

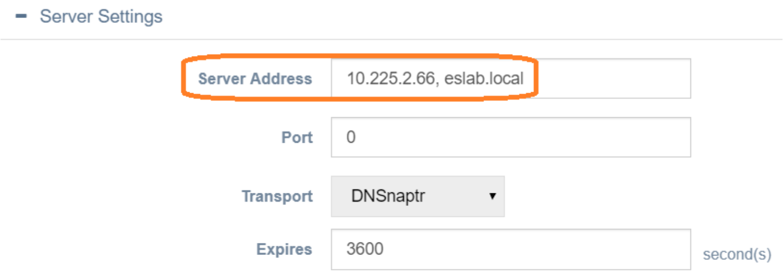
Warning

You **must** use DNSnaptr if you are using SRV or A records and the port **must** be set to “0” in order for the 8400 handset to perform a DNS lookup. If you fail to set these values accordingly the handset will not attempt the DNS lookup.

In addition, you can leverage reg.x parameters instead of the voIpProt parameter. These can be used in the same way as they are describe above but it will give you more granular control over individual line registrations. This is also how it would be done in CMS so it may make more sense to use this approach for the sake of consistency. Here's an example of how that might look.

```
<reg>
    <reg.server
        reg.1.server.1.address="eslab.local"
        reg.1.server.1.port="0"
        reg.1.server.1.transport="naptr"
        reg.1.server.1.expires="3600"/>
</reg>
```

Here's an example of CMS configuration:



Server Address: 10.225.2.66, eslab.local

Port: 0

Transport: DNSnaptr

Expires: 3600 second(s)

Option 2 – Creating a New Sub-Domain

To utilize this option, you will be creating a new sub-domain within your existing DNS server environment. Some customers may wish to use this approach to keep the configuration separate from their primary domain and make it more obvious how things are setup at a glance. The procedure for doing this is pretty simple.

In the domain you want to create your sub-domain, right click and choose New Domain from the list. This will open a new window where you can enter the name you want to use for your new sub-domain. Enter whatever you want this name to be but if you want to make it meaningful or useful for anyone to see what it’s for then enter something like, *sipserver*. Domain add a domain name to the end; we’re creating a domain. Click OK when you’re done.

You should now see a new folder in your DNS view with the name you just entered. This folder if a new sub-domain under whichever domain you chose to use. Click on the folder and you will see a new NAPTR record has been automatically created in this folder. Next we need to create the SRV records for your failover setup. This process if identical to what we’ve described above in Option 1 so please refer back to that section starting at paragraph 2.

When configuring the handset for the SIP Server you will enter the sub-domain. Per the example we used above, this might be *sipserver.mydomain.local*, as an example. The phone would then us this information locate the SRV and A records to perform the server registration.

Here’s an example of the XML you would include in your phone configuration file:

```
<voIpProt>
   <voIpProt.server
       voIpProt.server.1.address="sip.eslab.local"
       voIpProt.server.1.port="0"
       voIpProt.server.1.transport="naptr"
       voIpProt.server.1.expires="3600"/>
   <voIpProt.SIP
       voIpProt.SIP.enable="1"
       voIpProt.SIP.use486forReject="1"/>
</voIpProt>
```

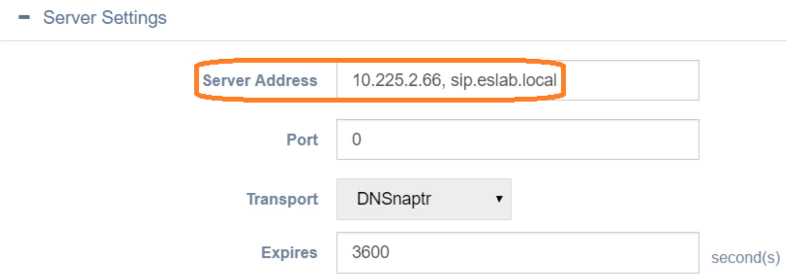
You'll need to replace the server address with your appropriate address and depending on how you've chosen to deploy, using NAPTR records or SRV and A records only, will determine what you'll set the transport type will be. Your options are: DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly.

> **Warning**
>
> You **must** use DNSnaptr if you are using SRV or A records and the port **must** be set to "0" in order for the 8400 handset to perform a DNS lookup. If you fail to set these values accordingly the handset will not attempt the DNS lookup.

In addition, you can leverage reg.x parameters instead of the voIpProt parameter. These can be used in the same way as they are describe above but it will give you more granular control over individual line registrations. This is also how it would be done in CMS so it may make more sense to use this approach for the sake of consistency. Here's an example of how that might look.

```
<reg>
   <reg.server
      reg.1.server.1.address="eslab.local"
      reg.1.server.1.port="0"
      reg.1.server.1.transport="naptr"
      reg.1.server.1.expires="3600"/>
</reg>
```

Here's an example of CMS configuration:



Server Settings

Server Address: 10.225.2.66, sip.eslab.local

Port: 0

Transport: DNSnaptr

Expires: 3600 second(s)

Step 2

Next we need to make sure the A records exist for our SIP servers. Without the A records the handsets will not be able to locate the appropriate IP address to be able to register. An A record is just a hostname entry in your DNS that points to an IP address. Regardless of which option you chose in Step 1 above, you will enter A records the same way.

To create an A record simply right click in the domain you wish to create the record and choose New Host (A or AAAA) from the list. You will enter only the hostname for the SIP server in the

Name field and then the server's IP address in the IP Address field. Be sure you click the check box next to Create Associated Pointer (PTR) Record as well so that the reverse lookup works. This allows you to lookup the IP address and get the hostname.

Perform this step for both, or all, of your SIP servers so there is a relevant hostname for each of them. What you call each of them is entirely up to you but you might consider something descriptive such as, avayasipprimary for an Avaya SIP server. Or something like SIPPrimary or SIPSecondary.

Troubleshooting

As with anything, there are sometimes problems. Fortunately, the 84-Series Handsets contain some very helpful tools to help you understand and identify what's wrong should something go wrong. This section will describe the procedure for troubleshooting your SIP Server Failover configuration when using DNS.

The first step will be to ensure you have your handset logging levels set appropriately. It is recommended that you do this via your configuration files but it can also be done via the handset's web interface. It is highly recommended that you setup a provisioning server for all troubleshooting as accessing logs will be much easier this way. Setup of a provisioning server is beyond the scope of this document but we have lots of useful information on how to do this in other bulletins and in the Admin and Deployment Guides for the 84-Series Handsets.

The first parameter we need to set is the Global Log Level Limit to allow the apprioriate level of logging to be output to log files. This is done via a single parameter, log.render.level="0", which will be set in your site.cfg file or it can be added to a handset specific file if you prefer.

Next, we will need to set the logging level for the module that will output the most useful information for us. This is also a single parameter, log.level.change.dns="1", which will output logging for the DNS logging module at Event 1. This parameter can also be in your site.cfg file or in a handset specific configuration file if you prefer.

Once you've gotten these two parameters onto your handset, however you choose to do that, we will want to restart the handset. Note that we're assuming you've already configured your handset to use the new SIP server name information we setup above. By restarting the handset we're ensuring that the logging information is being captured anew and it gives us a delineation point in the log file when we start parsing it. Once your handset boots up and displays the registration error, press and hold each of the following digits 1-5-9. Hold all three digits until the phone makes a tweedle sound. This indicates the function has completed and you can exit back to the standby screen. This function forces the phone to upload its current log files to the provisioning server. There will be a *macaddress*-now-app.log which is a snapshot of the last 300 or so lines of log but what we want to look at is the *macaddress*-app.log file as this will be a compilation of the phone's entire log.

Parsing the Log

Here are example logs from the Spectralink support lab. We'll go through this line by line to explain what we see. I am removing the timestamp information as it is not relevant.

What you'll see now is the logging module the message was generated from followed by the logging level that message outputs at and then the message itself. For the sake of readability I've also removed the 00 from the lines as it is not relevant. I've also added line numbers to make parsing easier to understand.

Failed Registration Log

This log is from a non-working setup as the domain specified doesn't exist, i.e. the NAPTR record doesn't exist. This approach allows you to look for typos and to see how the phone is interpreting the record responses it gets from DNS, if it gets a response at all.

```
1  |dns  |1|  |doDnsLookupForList for record NAPTR: hostname 'testserver.eslab.local' attempting..
2  |dns  |2|  |Miss in All DNS cache for testserver.eslab.local(NAPTR)
3  |dns  |2|  |Miss in All DNS cache for testserver(NAPTR)
4  |dns  |1|  |doDnsLookupForList for record A: hostname '10.225.2.35' attempting..
5  |dns  |1|  |doDnsLookupForList(A): returning passed in ipAddress '10.225.2.35'
6  |dns  |3|  |DNS lookup for (NAPTR) 'testserver.eslab.local' Server has no Answer
7  |dns  |3|  |DNS lookup for (NAPTR) 'testserver.eslab.local' found no match (err code: 4)
8  |dns  |2|  |doDnsLookupForList for record NAPTR: no match found for 'testserver.eslab.local'
9  |dns  |2|  |Added Negative DNS cache entry for testserver.eslab.local(NAPTR) expires in 36   seconds
10 |dns  |2|  |Found DNS cache entry for testserver.eslab.local(NAPTR)
11 |sip  |4|  |NAPTR query for host 'testserver.eslab.local' returned no results
12 |dns  |1|  |doDnsLookupForList for record SRV: hostname '_sip._udp.testserver.eslab.local' attempting..
13 |dns  |2|  |Miss in All DNS cache for _sip._udp.testserver.eslab.local(SRV)
14 |dns  |2|  |Miss in All DNS cache for _sip._udp.testserver(SRV)
15 |dns  |3|  |DNS lookup for (SRV) '_sip._udp.testserver.eslab.local' Server has no Answer
16 |dns  |3|  |DNS lookup for (SRV) '_sip._udp.testserver.eslab.local' found no match (err code: 4)
17|dns  |2|  |doDnsLookupForList for record SRV: no match found for '_sip._udp.testserver.eslab.local'
18 |dns  |2|  |Added Negative DNS cache entry for _sip._udp.testserver.eslab.local(SRV) expires in 36   seconds
19 |dns  |2|  |Found DNS cache entry for _sip._udp.testserver.eslab.local(SRV)
20 |dns  |1|  |doDnsSrvLookupForARecordList(udp): SRV lookup returned nothing
21 |dns  |1|  |doDnsSrvLookupForARecordList(udp): Doing DNS A lookup (port 0)
22 |dns  |1|  |doDnsLookupForList for record A: hostname 'testserver.eslab.local' attempting..
23 |dns  |2|  |Miss in All DNS cache for testserver.eslab.local(A)
24 |dns  |2|  |Miss in All DNS cache for testserver(A)
```

```
25 |dns  |3|  |DNS lookup for (A) 'testserver.eslab.local' Server has no Answer
26 |dns  |3|  |DNS lookup for (A) 'testserver.eslab.local' found no match (err code: 4)
27 |dns  |2|  |doDnsLookupForList for record A: no match found for 'testserver.eslab.local'
28 |dns  |2|  |Added Negative DNS cache entry for testserver.eslab.local(A) expires in 36   seconds
29 |dns  |2|  |Found DNS cache entry for testserver.eslab.local(A)
30 |sip  |4|  |doDnsListLookup(udp): doDnsSrvLookupForARecordList 'testserver.eslab.local' found no records.
31 |sip  |4|  |Registration failed User: 1512, Error Code:480 Temporarily not available
```

Line 6 above indicates that the DNS server did not have a valid response for the address requested. The handset will cache any DNS lookups it does to help speed up future queries and to allow the handset to looks for changes. This is evident in line 9.

Because the handset was setup to perform NAPTR lookup, e.g. the Transport type was set to DNSNAPTR for the SIP Server, the handset also attempts to find SRV records. Note that the SRV lookup, starting on line 15, has a domain name that includes the full SIP Server name entered into the phone. In this case it is looking for testserver.eslab.local as the domain. This too fails because the path is not valid in the DNS server. You'll also notice that the SRV record includes the default service and protocol information, in this case _sip and _udp.

The last line of the log shows that SIP registration has failed. The SIP 480 message can be a little misleading here because the phone is essentially generating this itself since we never did find the SIP server's IP address. So the resolution to this log is that we're missing the domain or path to the server information. You'll need to create the subdomain specified, in this case, *testserver* in the domain *eslab.local*. The SRV records then need to be created in this new domain.

Successful Registration Log

Now let's look at a log from after we made the changes to the DNS server. This log is from the same environment with the proper DNS configuration made to allow for registration to occur. All setup was performed per the steps in this guide.

As with the previous log, I've removed the time stamps and the 00 then added the line numbers.

```
1  |sip  |*|  |Sip Register Usr:1512 Dsp:1512 Auth:'1512' Inx:0
2  |dns  |1|  |doDnsLookupForList for record NAPTR: hostname 'testserver.eslab.local' attempting..
3  |dns  |2|  |Miss in All DNS cache for testserver.eslab.local(NAPTR)
4  |dns  |2|  |Miss in All DNS cache for testserver(NAPTR)
5  |dns  |1|  |doDnsLookupForList for record A: hostname '10.225.2.35' attempting..
6  |dns  |1|  |doDnsLookupForList(A): returning passed in ipAddress '10.225.2.35'
7  |dns  |3|  |DNS lookup for (NAPTR) 'testserver.eslab.local'() TTL=36
8  |dns  |1|  |Current List:
9  |dns  |1|  |order      '2'
```

```
10 |dns  |1|  |preference  '10'
11 |dns  |1|  |flags       'S'
12 |dns  |1|  |service     ''
13 |dns  |1|  |regexp      ''
14 |dns  |1|  |replacement '_sip._udp.es'
15 |dns  |1|  |naptr: Added to tail
16 |dns  |2|  |Added Dynamic DNS cache entry for testserver.eslab.local(NAPTR) expires in 36   seconds
17 |dns  |1|  |doDnsLookupForList for record SRV: hostname '_sip._udp.testserver.eslab.local' attempting..
18 |dns  |2|  |Miss in All DNS cache for _sip._udp.testserver.eslab.local(SRV)
19 |dns  |2|  |Miss in All DNS cache for _sip._udp.testserver(SRV)
20 |dns  |3|  |DNS lookup for (SRV) '_sip._udp.testserver.eslab.local'(,) TTL=36
21 |dns  |1|  |srv: Added to tail
22 |dns  |1|  |target:port 'piper.eslab.local.:5060'
23 |dns  |1|  |priority    '1'
24 |dns  |1|  |weight      '10'
25 |dns  |1|  |srv: Added to tail
26 |dns  |1|  |target:port 'piper2.eslab.local.:5060'
27 |dns  |1|  |priority    '2'
28 |dns  |1|  |weight      '20'
29 |dns  |2|  |Added Dynamic DNS cache entry for _sip._udp.testserver.eslab.local(SRV) expires in 36   seconds
30 |dns  |1|  |dnsSrv2A calling LookupForList(A) for 'piper.eslab.local.' port 5060
31 |dns  |1|  |doDnsLookupForList for record A: hostname 'piper.eslab.local.' attempting..
32 |dns  |2|  |Miss in All DNS cache for piper.eslab.local.(A)
33 |dns  |2|  |Miss in All DNS cache for piper(A)
34 |dns  |3|  |DNS lookup for (A) 'piper.eslab.local.'(10.225.2.31) TTL=36
35 |dns  |2|  |Added Dynamic DNS cache entry for piper.eslab.local.(A) expires in 36   seconds
36 |dns  |1|  |dnsSrv2A calling LookupForList(A) for 'piper2.eslab.local.' port 5060
37 |dns  |1|  |doDnsLookupForList for record A: hostname 'piper2.eslab.local.' attempting..
38 |dns  |2|  |Miss in All DNS cache for piper2.eslab.local.(A)
39 |dns  |2|  |Miss in All DNS cache for piper2(A)
40 |dns  |3|  |DNS lookup for (A) 'piper2.eslab.local.'(10.225.2.32) TTL=36
41 |dns  |2|  |Retaining Dynamic DNS cache for piper.eslab.local.(A)
42 |dns  |2|  |Added Dynamic DNS cache entry for piper2.eslab.local.(A) expires in 36   seconds
43 |dns  |1|  |dnsSrv2A for host '_sip._udp.testserver.eslab.local' returned OK
44 |dns  |1|  |doDnsLookupForList for record A: hostname '10.225.2.31' attempting..
```

```
45 |dns  |1|  |doDnsLookupForList(A): returning passed in ipAddress '10.225.2.31'
46 |sip  |*|  |Fast Boot Measurement Point: Ready for Call, uptime: 491.052 sec.
47 |dns  |1|  |doDnsLookupForList for record NAPTR: hostname 'testserver.eslab.local' attempting..
48 |dns  |2|  |Hit in Dynamic DNS cache for testserver.eslab.local(NAPTR) expires in 3420 seconds
49 |dns  |1|  |doDnsLookupForList for record SRV: hostname '_sip._udp.testserver.eslab.local' attempting..
50 |dns  |2|  |Hit in Dynamic DNS cache for _sip._udp.testserver.eslab.local(SRV) expires in 36   seconds
51 |dns  |1|  |dnsSrv2A calling LookupForList(A) for 'piper.eslab.local.' port 5060
52 |dns  |1|  |doDnsLookupForList for record A: hostname 'piper.eslab.local.' attempting..
53 |dns  |2|  |Hit in Dynamic DNS cache for piper.eslab.local.(A) expires in 36   seconds
54 |dns  |1|  |dnsSrv2A calling LookupForList(A) for 'piper2.eslab.local.' port 5060
55 |dns  |1|  |doDnsLookupForList for record A: hostname 'piper2.eslab.local.' attempting..
56 |dns  |2|  |Hit in Dynamic DNS cache for piper2.eslab.local.(A) expires in 36   seconds
57 |dns  |1|  |dnsSrv2A for host '_sip._udp.testserver.eslab.local' returned OK
58 |dns  |1|  |doDnsLookupForList for record A: hostname '10.225.2.31' attempting..
59 |dns  |1|  |doDnsLookupForList(A): returning passed in ipAddress '10.225.2.31'
60 |dns  |1|  |doDnsSrvLookupForARecordList(udp): Doing DNS A lookup (port 5060)
61 |dns  |1|  |doDnsLookupForList for record A: hostname '10.225.2.31' attempting..
62 |dns  |1|  |doDnsLookupForList(A): returning passed in ipAddress '10.225.2.31'
63 |dns  |1|  |doDnsSrvLookupForARecordList(udp): kept port at 5060
64 |dns  |1|  |doDnsLookupForList for record A: hostname '10.225.2.31' attempting..
65 |dns  |1|  |doDnsLookupForList(A): returning passed in ipAddress '10.225.2.31'
```

The first thing you'll probably notice is how much longer this log is than the previous one. This is largely because of the additional information that was output to the phone from the DNS server that is captured here. Starting at line 7 you'll see the lookup for the *testserver.eslab.local* address we gave to the phone. Lines 9 through 14 are all details the phone gets from the DNS server. The important bit is the service information it gets from the replacement string.

On line 16 you can see that the phone has added this information to its DNS Cache for future referencing. Lines 18 and 19 are the phone looking at the same internal cache for the SRV information. The Miss indicates that this information is not yet available in cache. This causes the phone to perform the query in line 20.

Line 22 through 28 is the response to the SRV server lookup information. This is probably the most important information as you can see now that the phone has received SIP Server information along with the priority and weight of each server. In this case, *piper.eslab.local* will be the primary and *piper2.eslab.local* will be the secondary based on the priority for each record.

Now the phone must perform a lookup for the SIP server hostnames it just got from the SRV records. You see the phone first attempt to find the information in its cache on line 30 and 31 with the discovery that it is not available on line 32 and 33. The query to the DNS server on line 34 shows we got a response of *10.225.2.31* which will then be committed to cache per line 35.

The phone then does the same thing for the second SIP server hostname it received. The final portion shows the phone then repeating the entire process but this time showing that it is using the information in cache. At this point the phone then registers to the primary SIP server and everything is working as expected.

There are potentially other failure scenarios not covered here but when it comes to performing troubleshooting of this particular feature, your best friend will be the logs. You can also set your SIP logging level to 1 or 0 if you’re having issues with registering to the SIP server or you’re not sure if it is registering the way you expected. Pay close attention to the DNS information you receive and make sure that’s what you really want it to do.

Lastly, be sure to really read and understand the failover section of the Admin Guide as it has a lot of possible configuration options available to you that will help you manage and handle failover. You may want to tweak how the phone fails back to the primary or how it handles failover timing. All of this is detailed in the Admin Guide. If you need help then please contact your Spectralink Authorized Reseller or Distribution Partner. They will be quite capable of helping you set this up and further troubleshoot your environment.

Copyright Notice

© 2020 Spectralink Corporation All rights reserved. Spectralink^{TM}, 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.

Warranty

The *Product Warranty and Software License and Warranty* and other support documents are available at http://support.spectralink.com.

Contact Information

US Location
+1 800-775-5330

Spectralink Corporation
2560 55th Street
Boulder, CO 80301
USA

info@spectralink.com

Denmark Location
+45 7560 2850

Spectralink Europe ApS
Bygholm Soepark 21 E Stuen
8700 Horsens
Denmark

infoemea@spectralink.com

UK Location
+44 (0) 20 3284 1536

Spectralink Europe UK
329 Bracknell, Doncastle Road
Bracknell, Berkshire, RG12 8PE
United Kingdom

infoemea@spectralink.com