Troubleshoot HTTPS Integration Issues Between the Conductor and the CUCM

Contents

[Introduction](#)
[Problem](#)
[Registration Status Shows *Unregistered*](#)
[Ad hoc Conference Creation Fails](#)
[Solution](#)
[SIP Trunk Configured with FQDN](#)

Introduction

This document describes a problem that is encountered with HTTPS integration between the Cisco Conductor and the Cisco Unified Communications Manager (CUCM).

Problem

The HTTPS integration between the Conductor and the CUCM for ad hoc conferences fails. There are two main symptoms when this problem occurs:

- The registration status for the Conductor Conference Bridge on the CUCM shows as **Unregistered**.

- Attempts to create an ad hoc conference fail.

The sections that follow explain these two symptoms in further detail.

Registration Status Shows *Unregistered*

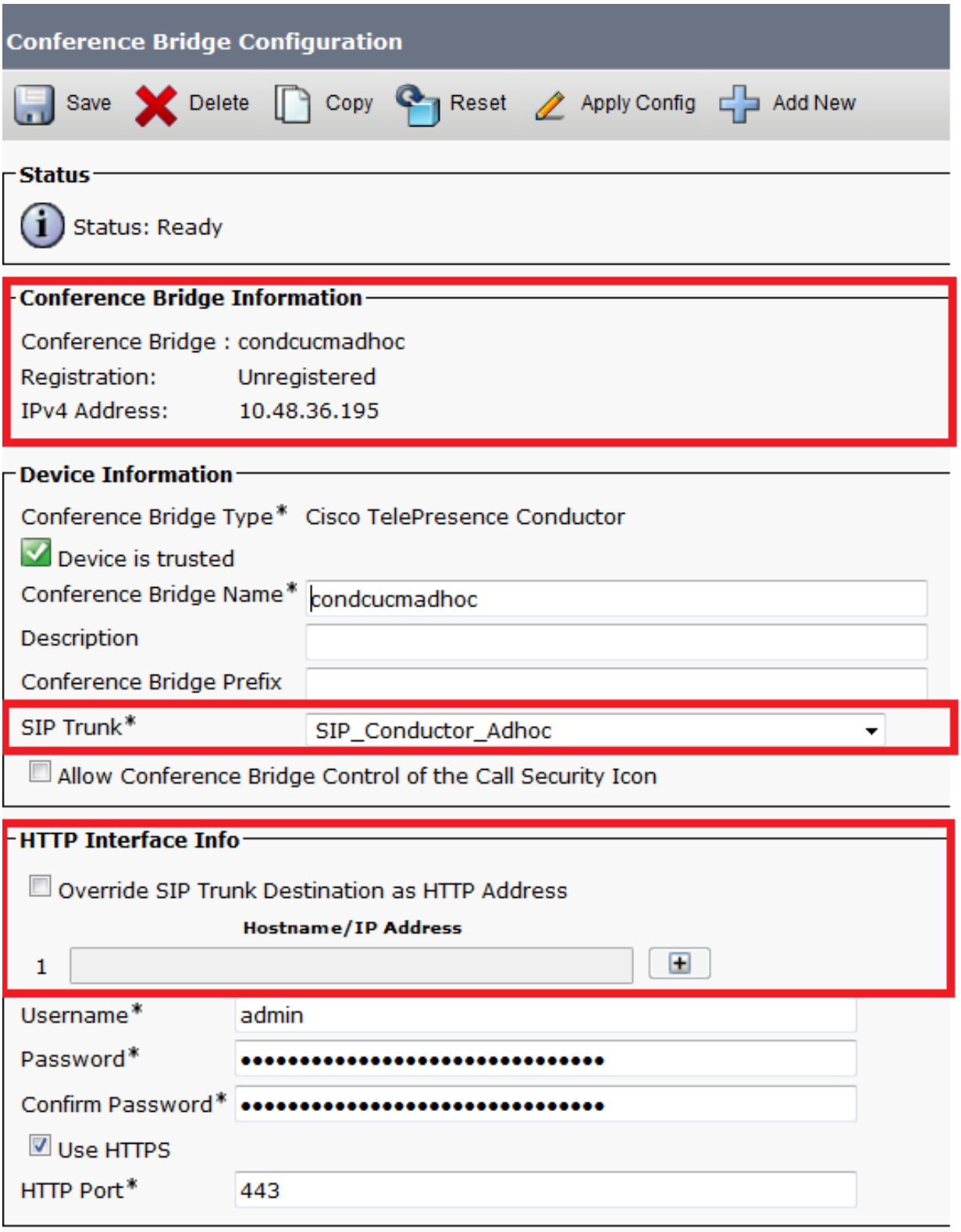
This symptom is observed in these two scenarios:

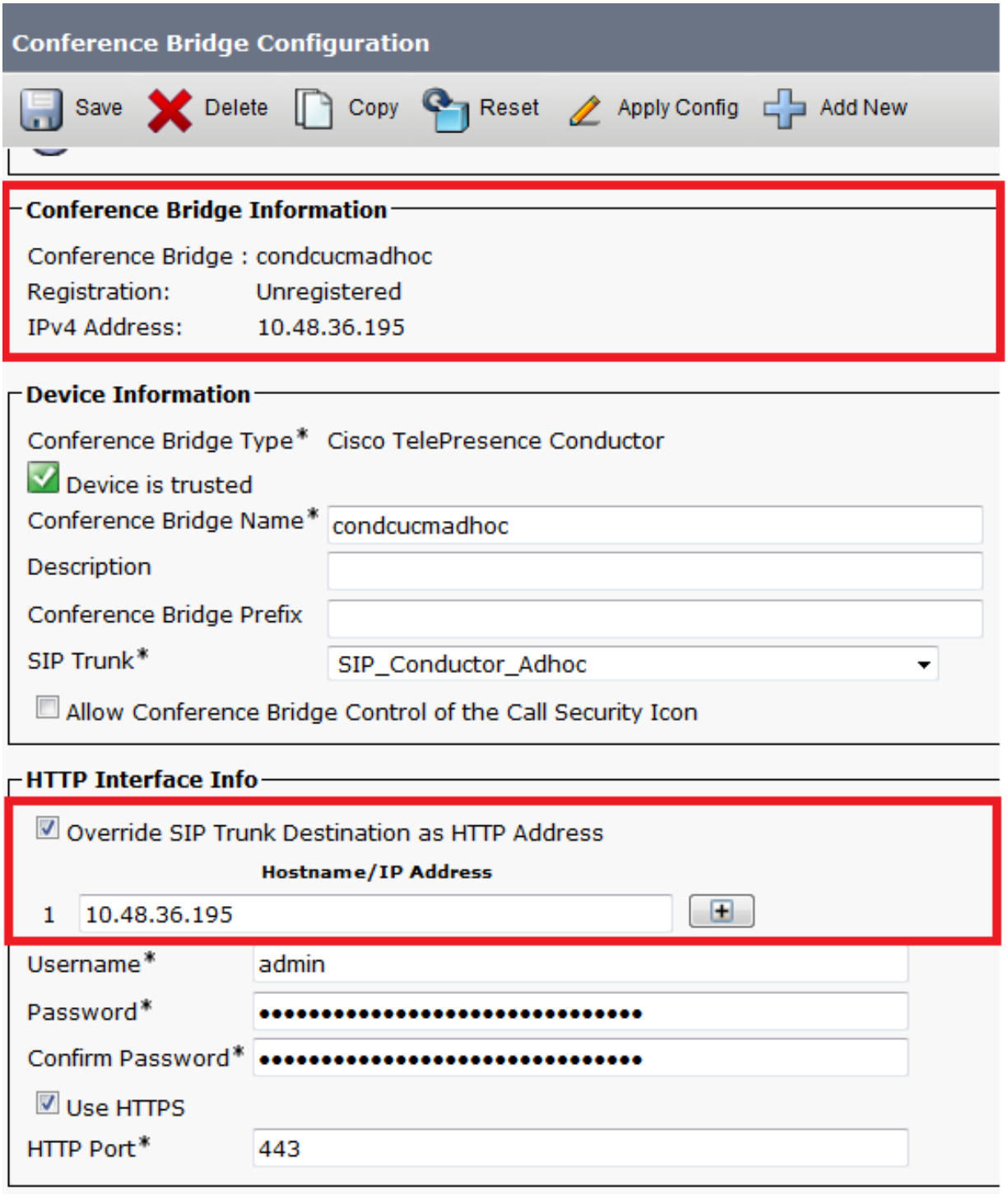
- The *Override SIP trunk destination as HTTP Address* check box is unchecked on the Conductor configuration page, and the associated Session Initiation Protocol (SIP) trunk for the Conductor Conference Bridge has a destination address that is configured as an IP address or a Fully Qualified Domain Name (FQDN).
  Tip: For more information about the FQDN SIP trunk scenario, refer to the SIP Trunk Configured with FQDN section of this document.

- The *Override SIP trunk destination as HTTP Address* check box is checked on the Conductor

configuration page and is configured as an IP address.

These images show the registration status for both of these scenarios:





The root cause for this registration failure is the library that is used for HTTPS/ Transport Layer Security (TLS). The TLS handshake fails with an *Encrypted* alert because the library does not support Uniform Resource Identifiers (URIs) in IP address format for HTTPS/TLS.

At a high level, the TLS handshake occurs similar to this:

1. The CUCM sends a *TLS Client Hello* message to the Conductor.

2. The Conductor sends a *Server Hello* message and certificate information to the CUCM.

3. The Conductor sends the *Server Hello Done* and *Server Key Exchange* messages to the CUCM.

4. The CUCM sends the *Client Key Exchange*, *Change Cipher Spec*, and *Encrypted*

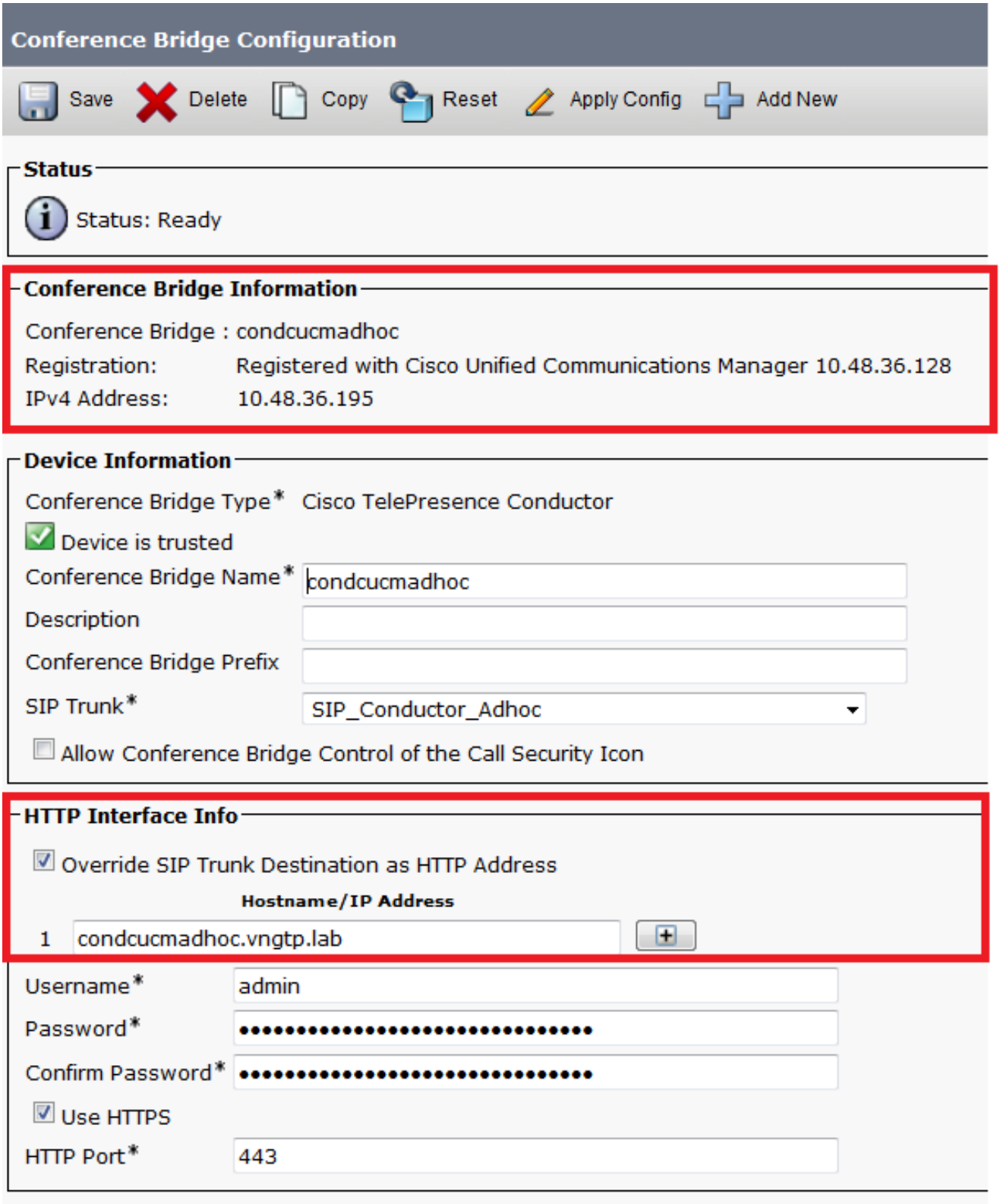
Handshake messages to the Conductor.

5. The Conductor sends the *Change Cipher Spec* and *Encrypted Handshake* messages to the CUCM.

6. The CUCM sends an *Encrypted* alert to the Conductor.

Ad hoc Conference Creation Fails

This symptom is observed when a workaround is applied for the aforementioned symptom, which causes the creation of ad hoc conferences to fail:



The root cause for this symptom is the Conductor, which fails to process the **conference.create** Application Program Interface (API) call from the CUCM when the URI is built with an FQDN.

The Conductor then logs this event:

```
Event="An API request could not be processed." Command="conference.create"
Conference_name="001035060001" Detail="<Fault 201:
'Request received  to a non ad-hoc IP address'>
```

Note: The **Conference_name** value is different for each call.

Solution

In order for the HTTPS integration and the ad hoc conference creation to function properly between the CUCM and the Conductor, a fix is required for Cisco bug ID CSCut22572. This fix should allow the HTTPS destination address to be configured as an FQDN.

> **Note**: The FQDN must resolve to the Virtual IP (VIP) that is associated with the Conductor ad hoc location and must be included as a Subject Alternative Name (SAN) attribute in the Conductor Certificate.

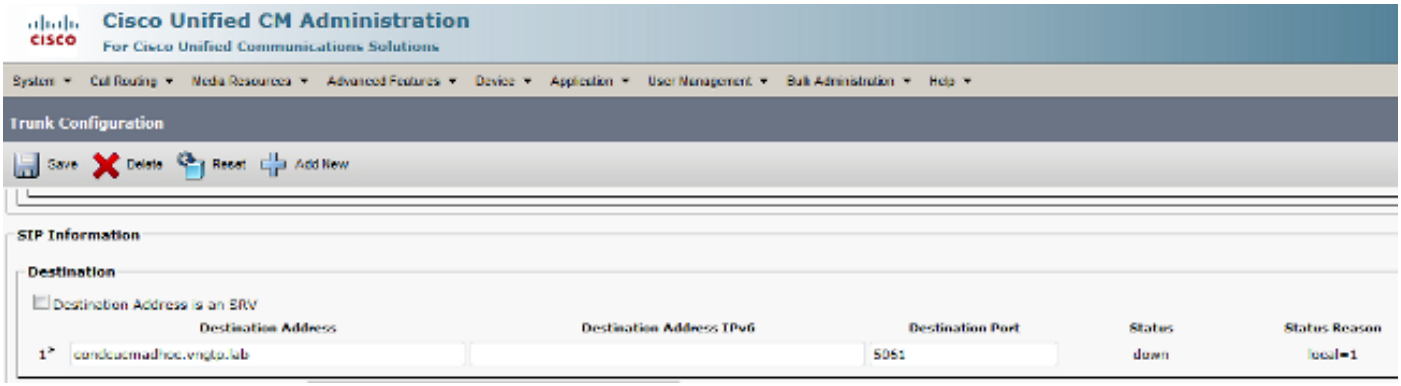
The long term the feature enhancement that is described in Cisco bug ID CSCut10254 will allow the HTTPS destination address to be configured with an IP address, either from a manual/override configuration or from the SIP trunk.

SIP Trunk Configured with FQDN

The SIP trunk service state can sometimes appear as *No Service* or *Down*. This occurs when:

- The destination address in the SIP trunk is configured with an FQDN.
- The FQDN resolves to a VIP that is associated with the ad hoc location that is indicated on the Conductor configuration page.

Here is an example:



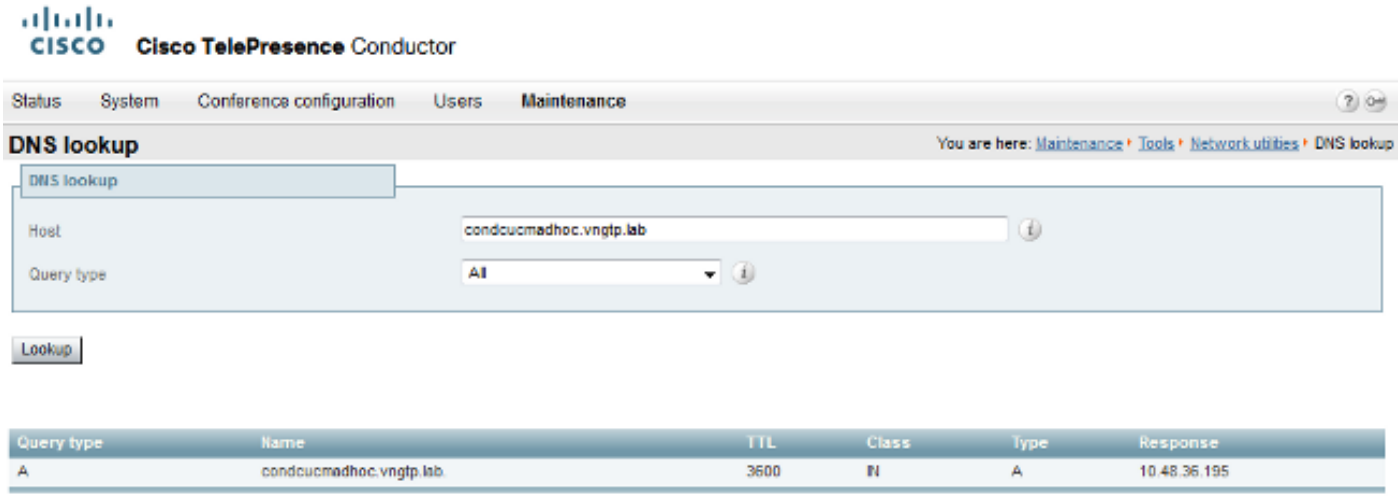
The root cause for this is the Conductor, which does not reply to the *SIP Options* message that is sent from the CUCM. The SIP URI is built based on the destination address, which is an FQDN in this example, and the Conductor expects an IP address notation:

```
2015-03-27T18:00:23+01:00 conductorcucm b2bua[28262]: UTCTime="2015-03-27 17:00:23,269"
Module="network.sip" Level="DEBUG":  Action="Received"  Local-ip="10.48.36.195"
Local-port="5061"  Src-ip="10.48.36.128"  Src-port="40523"
Msg-Hash="17750686918648045057"

 SIPMSG:
 |OPTIONS sip:condcucmadhoc.vngtp.lab:5061 SIP/2.0
 Via: SIP/2.0/TLS 10.48.36.128:5061;branch=z9hG4bK1539977cd7264
 Call-ID: c0a17300-51518ca7-15313-8024300a@10.48.36.128
 CSeq: 101 OPTIONS
 Contact: <sip:10.48.36.128:5061;transport=tls>
 From: <sip:10.48.36.128>;tag=1335522536
```

```
 To: <sip:condcucmadhoc.vngtp.lab>
 Max-Forwards: 0
 User-Agent: Cisco-CUCM10.5
 Date: Fri, 27 Mar 2015 17:00:23 GMT
 Content-Length: 0 2015-03-27T18:00:23+01:00 conductorcucm b2bua[28262]: UTCTime="2015-03-27
17:00:23,322"
Module="developer.applicationmanager.search" Level="INFO"
CodeLocation="ppcmains/ivy/search/SearchFsmState_Idle.cpp(82)"
Method="SearchFsmState_Idle::handleRequest" Thread="0x7feea9888700":
AppId="59" LegId="ASide[1]" CurState="SearchFsmState_Idle"
 Detail="Received search" searchContext="mTarget  : sip:condcucmadhoc.vngtp.lab
 mRouteSet:
 "
2015-03-27T18:00:23+01:00 conductorcucm b2bua[28262]: UTCTime="2015-03-27 17:00:23,325"
Module="developer.applicationmanager.search" Level="INFO"
CodeLocation="ppcmains/ivy/search/SearchFsmState_Idle.cpp(96)"
Method="SearchFsmState_Idle::performSearch" Thread="0x7feea9888700":
AppId="59" LegId="BSide[1]" CurState="SearchFsmState_Idle"
 Detail="Initiating search" searchContext="mTarget  : sip:condcucmadhoc.vngtp.lab
 mRouteSet:
 "
2015-03-27T18:00:23+01:00 conductorcucm b2bua[28262]: UTCTime="2015-03-27 17:00:23,344"
Module="developer.modulefactory.threadeddispatcher" Level="ERROR"
CodeLocation="ppcmains/ivy/threadeddispatcher/ThreadedDispatcher.cpp(106)"
Method="ThreadedDispatcher::run" Thread="0x7feea9888700":  Detail="Caught
std::exception" what="DefaultRouteHeaderStrategy::manipulateOutgoingRouteSet:
Policy routing configured, but no outgoing route found."
```

This occurs even though the conductor can resolve the ad hoc FQDN:



Note: Unless otherwise documented, this issue is also tracked in Cisco bug ID CSCut22572.