Understand CUSP Terminology and Routing Logic

Contents

Introduction
Prerequisites
Requirements
Terminology
Definitions
Network Topology
Call Example
Basic Call Routing
Configurations
Key Configuration Elements
Full Configuration
Troubleshoot
Trace Levels Configuration
Trace Collection
Trace Order
Trigger-Condition Trace Sample
Routing Trace Sample
SIP-Wire-Log Trace Sample
Architectural Reference

Introduction

This document explains how Cisco Unified SIP Proxy (CUSP) call routing logic.

Contributed by Joshua Meadows, Cisco TAC Engineer.

Prerequisites

Requirements

Cisco recommends that you have the knowledge of these topics:

- General knowledge of Session Initiation Protocol (SIP)
- Conceptual Understanding of CUSP in voice network deployments

Terminology

Definitions
Term | Definition
---|---
Network | A SIP network is a logical collection of local interfaces that can be treated the same for general routing purposes.


The Network logically defines areas of the network. The network can be defined using interfaces or specific ports can be used to provide segmentation. To achieve this logical segmentation, separate listen ports can be configured.

**Network** (Example: Listen Ports 14.50.245.9:5060, 14.50.245.9:5062, 14.50.245.9:5065 can define three logical networks using a single CUSP layer 3 interface)

Once the Networks are defined logically, they can be used to configure Triggers based on the Network.

**Note:** If you set up a listen port, ensure devices sending traffic to the CUSP use the correct port. If you set up listen port 14.50.245.9:5065 for CUCM traffic, you must ensure CUCM sends traffic to port 5065.

**Triggers**

Triggers can be set to identify incoming messages.

Triggers can identify Inbound Network, Local Port, Remote Network, etc.

**Server Group**

Both Server Group and Route Group can be used as destinations in the Route Table. A server group can be used for redundant devices of the same type. A CUBE stack would be a good example of a Server Group.

A route group allows you to designate the order in which gateways and trunks are selected. It allows for the prioritization of gateways and ports for outgoing trunk selection.

**Route Group**

Both Server Group and Route Group can be used as destinations in the Route Table. A route group allows you to configure weighted group destinations to reach the same device.

A direct SIP trunk to a CUCM and a SIP trunk to a PSTN gateway to reach the CUCM would be a good example of a Route Group. The direct SIP trunk to the CUCM would be the preferred method, and the PSTN route would be a backup.

You configure route tables to direct SIP requests to their appropriate destinations. Each route table consists of keys that are matched based on the lookup policy.

**Route Table**

Route Tables in CUSP are similar to Layer 3 routing tables. CUSP Route Tables consist of keys that are matched based on the lookup policy.

In the CUSP Route Table, keys can be mapped to the following route types to route SIP messages:

- **destination**: a specific host or a locally configured server group can be configured as a destination
- **route-group**: a locally configured route-group with one or more elements
- **route-policy**: route policies can be used to move between Route Tables similar to translation patterns
- **response**: rather than routing a SIP message, CUSP can send a specific response to terminate the call
- **default-sip**: Simple routing following RFC 3263.

**Note:** If mapping a key to a route-policy, be cognizant of logical loops.

A Route Policy points to a Route Table and defines how to use the Key in that route table.

**Example:**

- Route Table Name: "FromCUCM105-RT"
- Lookup Key matches: "Prefix-Longest-Match"
- Lookup Key: "SIP Header: 'To' Phone"

By separating the definition of the Key from the configured value of the Key, the same Route Table can be used in different contexts.
ways. For example, one Route Policy could define the Route Table's **Key** as the prefix for a **TO:** header. A different Route Policy could define the Route Table's **Key** as the prefix for a **FROM:** header.

**Routing** Routing Triggers link a Trigger to a Route Policy. **Triggers** Logically it states if a SIP message matches the Trigger, then use the configured Route Policy.

In summation, a SIP message is tagged with a **Network** based on the SIP listen port. The **Network** can be used to match a **Trigger**. The **Route Policy** then identifies which **Route Table** to use based on the **Trigger** and defines where to look for the **Key**. The **Route Table** will then use the **Key** to find out where to route the SIP message (Route Type). The Route Type (Host, **Server Group**, **Route Group**, etc) will be used to send the SIP message to the configured destination (**element**).

**Network Topology**

![Network Topology Diagram](image)

**Call Example**

Call from PSTN 1001 to 2003 on CUCM115

**Basic Call Routing**

**Incoming Network:** “PSTN”

**Trigger:** “From-PSTN-Trigger”

Triggers if incoming message matched Network “PSTN”

**Routing Trigger:** “FromPSTN-RPolicy” “From-PSTN-Trigger”

Links “From-PSTN-Trigger” to “FromPSTN-RPolicy”

**Route Policy:** “FromPSTN-RPolicy”

Specifies Routing table “PSTN-RT”
Specifies Lookup Key Matches “Prefix-Longest-Match”

Specifies Lookup Key is “SIP Header: ‘To’ Phone”

**Route Table:** “PSTN-RT”

Contains Key “2” to go to **Route Group** “CUCM115_RG”

**Route Group (or Server Group):** “CUCM115_RG”

Contains Element 14.50.245.20:5065

These configurations combine to make the logical statement:

For a call from the PSTN, where phone number prefix is 2, route to 14.50.245.20:5065

**Configurations**

**PSTN** - 2XXX and 5XXX calls are sent to CUSP via the CUBE and vCUBE

**CUCM 10.5** - 1XXX and 2XXX are sent to CUSP via SIP trunk

**CUCM 11.5** - 1XXX and 5XXX are sent to CUSP via SIP trunk

**Note:** When using the GUI, some configurations must be committed before they are available in other configuration sections. These are marked with ###Commit Configuration

**Key Configuration Elements**

<table>
<thead>
<tr>
<th>CLI Configuration</th>
<th>GUI Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Create a Network</strong></td>
<td><strong>Configure &gt;&gt; Networks &gt;&gt; Add</strong></td>
</tr>
</tbody>
</table>

sip network PSTN standard
Define listening port to identify network 'PSTN'
Configure >> Networks >> [Network Name] >> SIP
Listen Points >> Add

sip listen PSTN udp 14.50.245.9 5060

Trigger for Inbound Network 'PSTN'
Configure >> Triggers >> Add
Configure Trigger Name

trigger condition From-PSTN-Trigger
sequence 1
in-network ^\QPSTN\E$
end sequence
end trigger condition

Configure Trigger Condition and click add
route group CUCM115_RG
  element target-destination 14.50.245.20:5065:udp
  CUCM115 q-value 0.0
  failover-codes 502 - 503
  weight 50
  end element
end route

route table PSTN-RT
  key 2 group CUCM115_RG
  key 5 group CUCM105_RG
end route table
When configuring a Route Group as a destination in a Route Table, do NOT add a port and transport type. By adding a Port and/or Transport type, you are telling CUSP to look for DNS Host entry Cubestack:5060 rather than looking in the locally significant Server Group configurations.

Define the Key for 'FromPSTN-RPolicy'

Configure >> Route Policies >> Add (###Commit Configuration)

Configure a Route Policy name

policy lookup FromPSTN-RPolicy
sequence 100 PSTN-RT header to uri-component phone
rule prefix
end sequence
end policy

Click Add to add a Policy Step
The policy step will define how the Key is used. In this case, the policy looks for the longest Phone number match on the **To** field in the **SIP header**.

**Link the 'From-PSTN-Trigger' to 'FromPSTN-RPolicy'**

**Configure >> Routing Triggers >> Add**

Select a Routing Policy to link to a Trigger.

**Full Configuration**

**Note**: show configuration active verbose will show the entire configuration including the Route Tables.

```bash
josmeado-CUSP(cusp)# show configuration active verbose
Building CUSP configuration...
!
server-group sip global-load-balance weight
  server-group sip retry-after 250
  server-group sip element-retries udp 2
  server-group sip element-retries tls 1
  server-group sip element-retries tcp 1
  sip dns-srv
    enable
    no naptr
    end dns
!
  no sip header-compaction
  no sip logging
!
  sip max-forwards 70
  sip network CUCM105 standard
    no non-invite-provisional
```
allow-connections
no tls verify
retransmit-count invite-client-transaction 3
retransmit-count invite-server-transaction 5
retransmit-count non-invite-client-transaction 3
retransmit-timer T1 500
retransmit-timer T2 4000
retransmit-timer T4 5000
retransmit-timer TU1 5000
retransmit-timer TU2 32000
retransmit-timer clientTn 64000
retransmit-timer serverTn 64000
tcp connection-setup-timeout 1000
tls handshake-timeout 3000
udp max-datagram-size 1500
end network
!
sip network CUCM115 standard
no non-invite-provisional
allow-connections
no tls verify
retransmit-count invite-client-transaction 3
retransmit-count invite-server-transaction 5
retransmit-count non-invite-client-transaction 3
retransmit-timer T1 500
retransmit-timer T2 4000
retransmit-timer T4 5000
retransmit-timer TU1 5000
retransmit-timer TU2 32000
retransmit-timer clientTn 64000
retransmit-timer serverTn 64000
tcp connection-setup-timeout 1000
tls handshake-timeout 3000
udp max-datagram-size 1500
end network
!
sip network PSTN standard
no non-invite-provisional
allow-connections
no tls verify
retransmit-count invite-client-transaction 3
retransmit-count invite-server-transaction 5
retransmit-count non-invite-client-transaction 3
retransmit-timer T1 500
retransmit-timer T2 4000
retransmit-timer T4 5000
retransmit-timer TU1 5000
retransmit-timer TU2 32000
retransmit-timer clientTn 64000
retransmit-timer serverTn 64000
tcp connection-setup-timeout 1000
tls handshake-timeout 3000
udp max-datagram-size 1500
end network
!
sip overload reject retry-after 0
!
no sip peg-counting
!
sip privacy service
sip queue message
drop-policy head
low-threshold 80
size 2000
thread-count 20
end queue
!
sip queue radius
drop-policy head
low-threshold 80
size 2000
thread-count 20
end queue
!
sip queue request
drop-policy head
low-threshold 80
size 2000
thread-count 20
end queue
!
sip queue response
drop-policy head
low-threshold 80
size 2000
thread-count 20
end queue
!
sip queue st-callback
drop-policy head
low-threshold 80
size 2000
thread-count 10
end queue
!
sip queue timer
drop-policy none
low-threshold 80
size 2500
thread-count 8
end queue
!
sip queue xcl
drop-policy head
low-threshold 80
size 2000
thread-count 2
end queue
!
routerecursion
!
sip tcp connection-timeout 30
sip tcp max-connections 256
!
no sip tls
!
sip tls connection-setup-timeout 1
!
trigger condition From-CUCM105-Trigger
sequence 1
in-network ^\QCUCM105\ES$
end sequence
end trigger condition
!
trigger condition From-CUCM115-Trigger
sequence 1
in-network ^\QCUCM115\ES$
end sequence
end trigger condition
!
trigger condition From-PSTN-Trigger
  sequence 1
    in-network ^\QPSTN\E$
    end sequence
  end trigger condition
!
trigger condition mid-dialog
  sequence 1
    mid-dialog
    end sequence
  end trigger condition
!
accounting
  no enable
  no client-side
  no server-side
  end accounting
!
server-group sip group Cubestack PSTN
  element ip-address 14.50.245.6 5060 udp q-value 0.0 weight 1
  element ip-address 14.50.245.7 5060 udp q-value 0.0 weight 1
  failover-resp-codes 503
  lbtype weight ping
  end server-group
!
route group CUCM105_RG
  element target-destination 14.50.245.25:5062:udp CUCM105 q-value 0.0
    failover-codes 510
    weight 50
  end element
  end route
!
route group CUCM115_RG
  element target-destination 14.50.245.20:5065:udp CUCM115 q-value 0.0
    failover-codes 502 - 503
    weight 50
  end element
  end route
!
route table FromCUCM105-RT
  key * target-destination Cubestack PSTN
  key 2 group CUCM115_RG
  end route table
!
route table FromCUCM115-RT
  key 1 target-destination Cubestack PSTN
  key 5 group CUCM105_RG
  end route table
!
route table PSTN-RT
  key 2 group CUCM115_RG
  key 5 group CUCM105_RG
  end route table
!
policy lookup FromCUCM105-RPolicy
  sequence 100 FromCUCM105-RT header to uri-component phone
    rule prefix
  end sequence
  end policy
!
policy lookup FromCUCM115-RPolicy
sequence 100 FromCUCM115-RT header to uri-component phone
  rule prefix
  end sequence
end policy
!
policy lookup FromPSTN-RPolicy
sequence 100 PSTN-RT header to uri-component phone
  rule prefix
  end sequence
end policy
!
trigger routing sequence 1 by-pass condition mid-dialog
trigger routing sequence 2 policy FromPSTN-RPolicy condition From-PSTN-Trigger
trigger routing sequence 3 policy FromCUCM115-RPolicy condition From-CUCM115-Trigger
trigger routing sequence 4 policy FromCUCM105-RPolicy condition From-CUCM105-Trigger
!
server-group sip global-ping
!
no server-group sip ping-503
!
sip cac session-timeout 720
sip cac PSTN 14.50.245.6 5060 udp limit -1
sip cac PSTN 14.50.245.7 5060 udp limit -1
!
no sip cac
!
sip listen CUCM105 udp 14.50.245.9 5062
sip listen CUCM115 udp 14.50.245.9 5065
sip listen PSTN udp 14.50.245.9 5060
!
call-rate-limit 100
!
end

Troubleshoot

Trace Levels Configuration

In the CUSP GUI, navigate to Troubleshoot >> Cisco Unified SIP Proxy >> Traces

Trigger-Conditions - Level:debug: This will show which triggers were match to initiate call routing.

Routing - Level:debug: This will show what was done during call routing. Which Key’s were matched, what destination was chosen, etc.

SIP-Wire-Log - Level:debug: This will show the SIP messages received and sent.

Trace Collection

Via GUI

In the CUSP GUI, navigate to Troubleshoot >> Cisco Unified SIP Proxy >> Traces

Select Download Log File

You can also Clear Logs
Via FTP Client

By default there is no account with FTP privilages. To enable an account with FTP privilages add the user to a PFS group.

josmeado-CUSP# user platformadmin group ?
Administrators System administrators group
pfs-privusers PFS privileged users group
pfs-readonly PFS read only group
josmeado-CUSP# user platformadmin group pfs

Via FTP Client, connect to CUSP. **File Path:** cusp >> log >> trace >> trace.log

**Trace Order**

1. **SIP-Wire-Log** - Incoming SIP Invite
2. **SIP-Wire-Log** - Return 100 Trying
3. **Trigger-Condition** - Identify Network and Trigger Route Policy
4. **Routing** - See Routing Trace section below for details
5. **SIP-Wire-Log** - Send Invite toward destination
6. **SIP-Wire-Log** - Continue normal SIP transactions until there is a 200 Ok message for each call leg

**Trigger-Condition Trace Sample**

13:24:36:987 08:17:2017 vCUSP,9.1.5,josmeado-CUSP,14.50.245.9,trace.log
action<FromPSTN-RPolicy> actionParameter>
[REQUESTI.7] DEBUG 2017.08.17 13:25:03:006 triggers.ModuleTrigger - ModuleTrigger.eval() got the policy, executing it ...

In the above sample, we see the network is matched as PSTN, which is used in Route Policy "FromPSTN-RPolicy".

**Routing Trace Sample**

[REQUESTI.7] DEBUG 2017.08.17 13:29:33:987 nrs.FieldSelector - getToUri: To header obtained - To: <sip:2003@14.50.245.9>
1. CUSP gets the Key value in the TO: header

2. CUSP identifies the Key as 2003

3. CUSP Looks up the Key in the Routing Table

4. CUSP matches an entry in the Routing Table and identifies destination RouteGroup:CUCM115_RG

5. CUSP applies loadbalancing within the RouteGroup

6. CUSP identifies the specific Element in the RouteGroup to which it will send the SIP message
7. CUSP applies Time Policies if applicable

8. CUSP finalizes the Element to which it will send a SIP Message

SIP-Wire-Log Trace Sample


ShiftAlgorithms.execute()
ShiftAlgorithms.execute()
[REQUESTI.7] DEBUG 2017.08.17 13:29:33:987 nrs.FieldSelector - getToUri: To header obtained -
To: <sip:2003@14.50.245.9>
sip:2003@14.50.245.9 part 1
PSTN-RT with rule prefix and modifiers=none
applyModifiers()
applyModifiers(), returning 2003
[REQUESTI.7] INFO 2017.08.17 13:29:33:987 nrs.XCLPrefix - NRS Routing decision is:
RouteTable: PSTN-RT, RouteKey: 2, RouteGroup: CUCM115_RG
createLoadBalancer()
[REQUESTI.7] DEBUG 2017.08.17 13:29:33:988 nrs.XCLPrefix - Stored NRSAlgResult=isFound=true,
isFailure=false, Response=-1, Routes=[Ruri: 14.50.245.20:5065:udp, Route: null, Network: CUCM115, q-value=0.0radvance=[502, 503]], PolicyAdvance=null
[REQUESTI.7] DEBUG 2017.08.17 13:29:33:988 nrs.NRSAlgResult - set policyAdvance as specified in algorithm=(lookuprule=1, lookupfieldid=52, lookuplength=-1, lookuptable=PSTN-RT, sequence=100, algorithm=1)
ShiftRoutes.execute()
SIP-Wire-Log shows the normal SIP messaging up to the 200 Okay for both call legs.

Architectural Reference