Configuring and Troubleshooting Transparent CCS

Document ID: 19087

Contents

Introduction

This document describes how to configure and troubleshoot Transparent Common Channel Signaling (T−CCS).

Prerequisites

Requirements

Readers of this document should have knowledge of these topics:

- How to configure Cisco IOS® Software for Voice functionality.

Components Used

The information in this document is based on these software and hardware versions:

- Cisco IOS Software Release 12.2.7a.
- The Cisco 3640 Router.

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Background Information

T−CCS allows the connection of two PBXs with digital interfaces that use a proprietary or unsupported CCS protocol without the need for interpretation of CCS signaling for call processing.

With T−CCS, the PBX voice channels can be nailed up (made permanent), and compressed between sites. The accompanying signaling channel or channels can be tunneled (transmitted transparently) across the IP/FR/ATM backbone between PBXs. Thus, calls from the PBXs are not routed by Cisco on a call−by−call basis, but follow a preconfigured route to the destination.

There are three configurable ways of applying the feature:

- Frame−forwarding T−CCS
- Clear−channel T−CCS
- Cross−connect T−CCS

Cross−connect T−CCS is only possible on the Cisco 3810, and is not discussed in this document.

T−CCS Compatibility Matrix

This table shows the T−CCS features that can be configured on various platforms.

VoX[1]	Cisco 3810	Cisco 26xx/36xx/72xx
VoIP[2]	Clear−Channel: • Any type of CCS signaling. • Any number of signaling channels.	Clear−Channel: • Any type of CCS signaling. • Any number of signaling channels.
VoFR[3]	Clear−Channel: • Any type of CCS signaling. • Any number of signaling channels. Frame−Forwarding: • HDLC−framed signaling.[4] • Only 1 signaling channel: E1. • E1 = TS16. • T1= TS 24.	Clear−Channel: • Any type of CCS signaling. • Any number of signaling channels. Frame−Forwarding: • HDLC−framed signaling. • Signaling channels = Configurable channel−groups per controller.

	TDM[5] Cross−Connect: • Any type of CCS signaling. • Any number of signaling channels.	
VoATM[6]	Clear−Channel: • Any type of CCS signaling. • Any number of signaling channels. Frame−Forwarding: • HDLC−framed signaling. • Only 1 signaling channel.	Clear−Channel: • Any type of CCS signaling. • Any number of signaling channels. Frame−Forwarding: • HDLC−framed signaling. • Signaling channels = Configurable channel−groups per controller.

1. VoX = Voice over X

2. VoIP = Voice over IP

3. VoFR = Voice over Frame Relay

4. HDLC = High−Level Data Link Control

5. TDM = Time−Division Multiplexing

6. VoATM = Voice over ATM

Frame−Forwarding T−CCS

Frame−forwarding T−CCS can only be used to support PBX proprietary protocols where the signaling channel or channels are HDLC−framed, and the desired VoX technology is VoFR or VoATM. In this solution, the HDLC signaling frames are encapsulated and forwarded through a channel group that is configured for the signaling on the controller, and thus treated as a serial interface. The HDLC framing is interpreted and understood, although the signaling messages are not. Idle frames are suppressed, and only real data is propagated across the signaling channel.

Implement Frame−Forwarding T−CCS

Caveat: CSCdt55871 Limitation

There is a current limit on the number of usable voice channels when configuring frame−forwarding TCCS on E1. The limitation occurs because of a conflict between ds0−group and channnel−group number ranges, as is explained in CSCdt55871 (registered customers only) .

Trying to configure a ds0 group that is +1 of the previously input channel group results in failure, as shown below.

```
!
 controller t1 2/1
 channel-group 0 timeslot 24 speed 64
 ds0-group 1 timeslots 1 type ext-sig
```

The above configuration results in an error message when the ds0 group is defined, claiming that channel 0 is already used, as shown here:

```
%Channel 0 already used by other group
```

The workaround is to miss the conflicting group, and continue with the next group number in range. This reduces the number of configurable groups by one.

Be aware of these points before implementing frame−forwarding T−CCS:

- Frame−forwarding T−CCS must only be configured when the CCS protocol to be transported uses a HDLC type of framing.
- The **mode ccs−frame−forwarding** command defines frame−forwarding CCS.
- The **DSO−group** and the **ext sig** commands determine which voice ports are to be created and used for the trunk with external source signaling.
- The **connection trunk** command establishes permanent voice channels.
- The **channel−group** command defines the frame−forwarding timeslot or timeslots.
- Frame−forwarding T−CCS is not supported for VoIP.
- TS16 on E1 is always reserved for Channel−Associated Signaling (CAS). If you configure another timeslot for CAS (as in the above example), you then have one fewer timeslot for voice.

A Configuration Example for Frame−Forwarding VoFR T−CCS

The configuration and testing reported in this section was performed on a Cisco 3640 Router running Cisco IOS Software Release 12.2.7a. The example shown here represents a situation when the signaling is not applied on the normal timeslot (slot 16). Another timeslot is used here (slot 6) to show the versatility of the feature (not applicable on the Cisco 3810 Router).
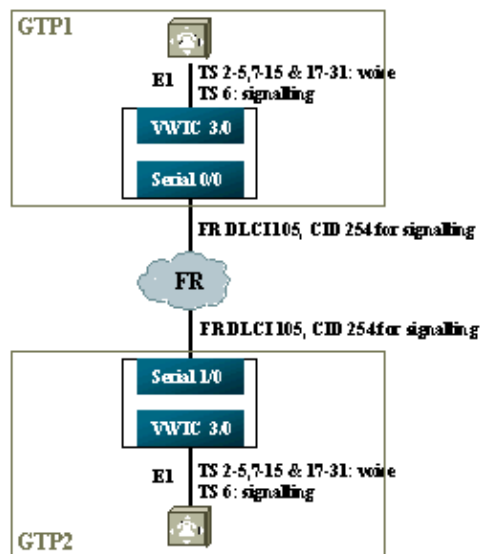
Configuration Steps for the Voice Side

To configure the Voice side, complete these steps:

1. On the T1 or E1 controller:

   a. Add the **mode ccs frame−forwarding** command.
   b. Define the channel−group for each signaling channel (for the Cisco 26xx and 36xx series only; the Cisco 3810 Router automatically creates the D−channel).
   c. Define ds0 groups for each voice channel, using type **ext−sig**.

   GTP1
   ```
   controller E1 3/0
    mode ccs frame-forwarding
    channel-group 0 timeslots 6
    ds0-group 2 timeslots 2 type ext-sig
    ds0-group 3 timeslots 3 type ext-sig
    .
    ds0-group 30 timeslots 30 type ext-sig
   ```

   GTP2
   ```
   controller E1 3/0
    mode ccs frame-forwarding
    channel-group 0 timeslots 6
    ds0-group 2 timeslots 2 type ext-sig
    ds0-group 3 timeslots 3 type ext-sig
    .
    ds0-group 30 timeslots 30 type ext-sig
   ```

2. On the D−channel interface (this serial interface gets created after the **channel−group** command is configured above):

   a. Add the **ccs encap frf11** command.
   b. Point the D−channel to a channel ID on the FR WAN interface by using the **ccs connect Serial x/y DLCI CID** command.

   Note: A separate channel ID must be used for each D−channel if more than one signaling channel is required. Start with channel ID 254, and work backwards.

   GTP1
   ```
   interface Serial3/0:0
    no ip address
    ccs encap frf11
    ccs connect Serial0/0 105 254
   ```

   GTP2
   ```
   interface Serial3/0:0
    no ip address
    ccs encap frf11
    ccs connect Serial0/0 105 254
   ```

3. On the voice ports:

   Add **connection trunk xxx** to each voice port. The number must match the *destination pattern* of the terminating voice port (POTS dial peer) on the other side. Only one side of the connection should specify "answer mode."

GTP1

```
!
voice-port 3/0:2
 timeouts wait-release 3
 connection trunk 6002
!
voice-port 3/0:3
 timeouts wait-release 3
 connection trunk 6003
!
... [channels 4-30 the same] ...
!
voice-port 3/0:30
 timeouts wait-release 3
 connection trunk 6030
!
```

GTP2

```
voice-port 3/0:2
 timeouts wait-release 3
 connection trunk 8002 answer-mode
!
voice-port 3/0:3
 timeouts wait-release 3
 connection trunk 8003 answer-mode
!
... [channels 4-30 the same] ...
!
voice-port 3/0:30
 timeouts wait-release 3
 connection trunk 8030 answer-mode
```

4. On the POTS dial peers:

   a. Add a VoFR dial peer that matches the connection trunk dialed number, and point it to the Frame Relay Data−Link Connection Identifier (DLCI).
   b. Add a POTS dial peer to each voice port that matches the number dialed by the **connection trunk xxx** statements from the other side.

GTP1

```
!
dial-peer voice 8002 pots
 destination-pattern 8002
 port 3/0:2
!
dial-peer voice 8003 pots
 destination-pattern 8003
 port 3/0:3
!
... [channels 4-30 the same] ...

dial-peer voice 6000 vofr
 destination-pattern 6...
 session target Serial0/0 105
!
```

GTP2

```
!
dial-peer voice 6002 pots
 destination-pattern 6002
```

```
 port 3/0:2
!
dial-peer voice 6003 pots
 destination-pattern 6003
 port 3/0:3

... [channels 4-30 the same] ...
!
 dial-peer voice 8000 vofr
 destination-pattern 8...
 session target Serial1/0 105
!
```

Configuration Steps for the WAN Side

To configure the WAN side, complete these steps:

1. Define a Frame Relay serial interface, and a point−to−point subinterface with normal VoFR.
2. Put in **voice−bandwidth** based on the number of channels and the codecs used for voice.
3. Allow additional bandwidth in the Committed Information Rate (CIR) for the signaling channel and other data that share this DLCI.

GTP1

```
interface Serial0/0
 no ip address
 encapsulation frame-relay
 frame-relay traffic-shaping
!
interface Serial0/0.1 point-to-point
 ip address 10.10.105.2 255.255.255.0
 frame-relay class voice-class
 frame-relay interface-dlci 105
  vofr cisco
!
map-class frame-relay voice-class
 no frame-relay adaptive-shaping
 frame-relay cir 512000
 frame-relay bc 5120
 frame-relay be 0
 frame-relay fair-queue
 frame-relay voice bandwidth 512000
 frame-relay fragment 640
!
```

GTP2

```
!
interface Serial1/0
 no ip address
 encapsulation frame-relay
 clock rate 768000
 frame-relay traffic-shaping
 frame-relay intf-type dce
!
interface Serial1/0.1 point-to-point
 ip address 10.10.105.1 255.255.255.0
 frame-relay class voice-class
 frame-relay interface-dlci 105
  vofr cisco
!
!
```

```
map-class frame-relay voice-class
 no frame-relay adaptive-shaping
 frame-relay cir 512000
 frame-relay BC 5120
 frame-relay be 0
 frame-relay fair-queue
 frame-relay voice bandwidth 512000
 frame-relay fragment 640
!
```

Bandwidth

The bandwidth provisioned in the backbone must allow for all the configured voice and signaling channels.
Because these configurations use the connection trunk, all the resulting voice and signaling channels are up all
the time. Voice Activation Detection (VAD) provides savings on the active voice channels (although not on
signaling), but VAD does not become active until the voice channels are established. So the initial bandwidth
needed per voice channel should take into account the codec used, plus header overhead. For VoFR, only the
bandwidth of the voice channels should be accounted for in the **voice bandwidth** and **LLQ** commands. The
bandwidth of the voice and signaling channels should be accounted for on the FR–to–WAN interface.

Troubleshoot and Verify Frame–Forwarding T–CCS

The following steps help verify that frame–forwarding T–CSS is operating as it should.

1. E1 controller must be up for voice ports to go off–hook and trunked.
2. Check whether the call is in place, and whether the correct Digital Signal Processors (DSPs) are
   allocated on timeslots.
3. If calls fail to connect, check the Permanent Virtual Circuit (PVC) status configuration or
   connectivity, and dial–peer provision.
4. If the **show voice port** command shows "idle" and "on hook" for any timeslot, check whether the
   related timeslot has the correct DSP version assigned, and is working correctly with the **show voice
   dsp** command.
5. Debug with the **debug TCCS signaling** command in logging buffered mode (this is very CPU
   intensive).

```
gtp2#show controllers e1 3/0
E1 3/0 is up.
  Applique type is Channelized E1 – balanced
  No alarms detected.
  alarm-trigger is not set
  Version info Firmware: 20011015, FPGA: 15
  Framing is CRC4, Line Code is HDB3, Clock Source is Line.
  Data in current interval (276 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs

gtp2#show voice dsp

DSP  DSP                 DSPWARE CURR  BOOT           VOICE        PAK     TX/RX
TYPE NUM CH CODEC        VERSION STATE STATE RST AI PORT   TS ABORT PACK COUNT
==== === == =======      ======= ===== ===== === == ====== == ===== ============
C549 000 01 g729ar8      3.4.49  busy  idle       0 3/0:18 18     0 119229/70248
C549 000 00 g729ar8      3.4.49  busy  idle     0 0 3/0:2  02     0  41913/45414
C549 001 01 g729ar8      3.4.49  busy  idle       0 3/0:19 19     0 119963/70535
C549 001 00 g729ar8      3.4.49  busy  idle     0 0 3/0:3  03     0  42865/47341
C549 002 01 g729ar8      3.4.49  busy  idle       0 3/0:20 20     0  77746/69876
```

```
gtp2#show voice call summary

PORT          CODEC    VAD VTSP STATE        VPM STATE
=========  ========  === ============    ==============
3/0:2.2      g729ar8  y  S_CONNECT         S_TRUNKED
3/0:3.3      g729ar8  y  S_CONNECT         S_TRUNKED
3/0:4.4      g729ar8  y  S_CONNECT         S_TRUNKED
3/0:5.5      g729ar8  y  S_CONNECT         S_TRUNKED
3/0:6.31     g729ar8  y  S_CONNECT         S_TRUNKED
```

```
gtp2#show frame-relay pvc

PVC Statistics for interface Serial1/0 (Frame Relay DCE)

             Active      Inactive      Deleted      Static
  Local        1            0            0            0
  Switched     0            0            0            0
  Unused       0            0            0            0

DLCI = 105, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE,
INTERFACE = Serial1/0.1

  input pkts 1201908     output pkts 2177352     in bytes 37341051
  out bytes 71856239     dropped pkts 0          in FECN pkts 0
  in BECN pkts 0         out FECN pkts 0         out BECN pkts 0
  in DE pkts 0           out DE pkts 0
  out bcast pkts 167     out bcast bytes 48597
  PVC create time 08:37:30, last time PVC status changed 02:47:05
  Service type VoFR-cisco
```

```
gtp2#show frame-relay fragment
interface    dlci  frag-type   frag-size  in-frag  out-frag  dropped-frag
Serial1/0.1 105   VoFR-cisco  640        172      169       0


debug tccs signaling

Log Buffer (8096 bytes):

08:55:47:  282 tccs packets received from the port.
08:55:47:  282 tccs packets received from the nework.
08:55:47: RX from Serial3/0:0:
08:55:47: tccs_db->vcd = 105, tccs_db->cid = 254
08:55:47: pak->datagramsize=20
BE C0 C0 00 FF 03 C0 21 09 48 00 0C 01 49 F3 69 00 0C 42 00
08:55:47:  282 tccs packets received from the port.
08:55:47:  283 tccs packets received from the nework.
08:55:47: RX from Serial1/0: dlci=105, cid=254, payld-type =0,
              payld-length=188, cid_type=424
08:55:47: datagramsize=20
BE C0 C0 00 FF 03 C0 21 0A 48 00 0C 03 EA DF 0D 00 0C 42 00
08:55:50:  282 tccs packets received from the port.
08:55:50:  284 tccs packets received from the nework.
```

```
08:55:50: RX from Serial1/0: dlci=105, cid=254, payld-type =0,
            payld-length=188, cid_type=424
08:55:50: datagramsize=20
BE C0 C0 00 FF 03 C0 21 09 48 00 0C 03 EA DF 0D 00 62 05 00
08:55:50:  283 tccs packets received from the port.
08:55:50:  284 tccs packets received from the nework.
       08:55:50: RX from Serial3/0:0:
08:55:50: tccs_db->vcd = 105, tccs_db->cid = 254
08:55:50: pak->datagramsize=20
BE C0 C0 00 FF 03 C0 21 0A 48 00 0C 01 49 F3 69 00 62 05 00
gtp2# wr t


!--- This shows packet forwarding and receiving.
```

Clear−Channel Codec T−CCS

Clear−channel T−CCS is used to support PBX proprietary protocols where the signaling channel(s) are ABCD−bit based or HDLC, or where the voice transport technology is VoIP. In this solution, the signaling channel and voice channels are configured as ds0groups, and all are treated as voice calls.

The real voice calls are permanently connected trunk connections using the voice codec of your choice. The signaling channel(s) are also permanently connected trunks using the clear−channel codec, which is similar to G.711 in sample and packet sizes, but automatically excludes echo cancellation and VAD. There is no intelligence in the software to know which channels are voice channels, and which are signaling channels. You must configure the timeslots that you know carry signaling traffic to match a dial peer that assigns the clear−channel codec, while the voice channels must match a dial peer that encodes voice (G.729, and others).

Implement Clear−Channel Codec T−CCS

Be aware of these points before you implement clear−channel T−CCS:

- Clear−channel T−CCS an be used for any type of digital E1 or T1 signaling (including HDLC−based framing).
- Any number of signaling channels can be supported.
- Clear−channel T−CCS can be used in VoIP, VoFR or VoATM environments
- The clear−channel codec is used for signaling channel or channels in clear−channel T−CCS.
- VoIP Signaling and voice bandwidth must be accounted for in IP RTP Priority or Low−Latency Queuing (LLQ).
- VoIPovFR/VoFR Signaling and voice can be on the same or separate DLCIs.
- VoFR Signlling bandwidth is counted as part of VoFR "voice bandwidth."
- With clear−channel T−CCS, signaling takes 64K of dedicated bandwidth (not including packet overhead).
- The **DSO−group** command configures voice and signaling channels.
- Cisco IOS Software is not aware of the which signaling channel is in use.
- Thirty−one DSPs are required for a PBX using signaling on timeslot 16 with 30 voice ports, so two trunks on E1 2MFT would exhaust the quantity of DSPs on NMV2 (62 are required).

When using clear−channel codecs to transport data traffic, it is important that network clocking is synchronized. This is because the DSP algorithm drops packets when buffer overruns occur, and uses its auto−fill algorithm when buffer underruns occur (fine for voice traffic, but not good for data traffic). Both of these situations are likely to cause the D−channel to fail and restart.

Configuration Example for Clear−Channel VoIP T−CCS

Configuration and testing of clear−channel VoIP T−CCS was performed on a Cisco 3640 router running Cisco IOS Software Release 12.2.7a. In the example shown here, the signaling is not applied on the normal timeslot (16). Another timeslot is used here (timeslot 6) to show the versatility of the feature.
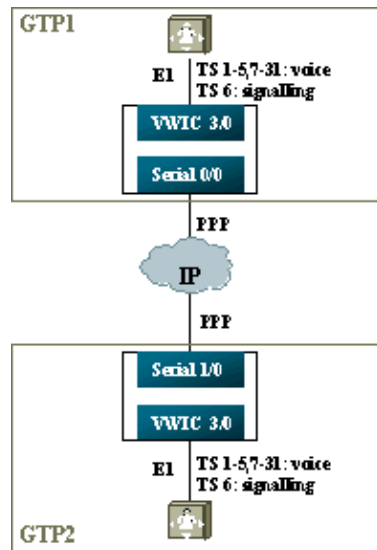
1. On the T1 or E1 controller:

   Define ds0 groups for each voice channel and signaling channel.

GTP1
   | ```
controller E1 3/0
 ds0-group 0 timeslots 6 type ext-sig
 ds0-group 1 timeslots 1 type ext-sig
 ds0-group 2 timeslots 2 type ext-sig
 ds0-group 3 timeslots 3 type ext-sig
 ds0-group 4 timeslots 4 type ext-sig
 ds0-group 5 timeslots 5 type ext-sig
 ds0-group 6 timeslots 31 type ext-sig
 ds0-group 7 timeslots 7 type ext-sig
 ....
 ds0-group 30 timeslots 30 type ext-sig
``` |

GTP2
   | ```
controller E1 3/0
 ds0-group 0 timeslots 6 type ext-sig

 ds0-group 1 timeslots 1 type ext-sig
 ds0-group 2 timeslots 2 type ext-sig
 ds0-group 3 timeslots 3 type ext-sig
 ds0-group 4 timeslots 4 type ext-sig
 ds0-group 5 timeslots 5 type ext-sig
 ds0-group 6 timeslots 31 type ext-sig
 ds0-group 7 timeslots 7 type ext-sig
 ....
 ds0-group 30 timeslots 30 type ext-sig
``` |

2. On the voice ports:

   a. Add a **connection trunk xxx** command to each voice port configuration. The number must match the destination pattern of the terminating voice port (POTS dial peer) on the other side.

b. Add a **connection trunk xxx** command to each signaling voice port configuration the number must match the destination pattern of the terminating voice port (POTS dial peer) on the other side.

c. Only one side of the connection should specify **answer mode**.

GTP1

```
voice-port 3/0:0
 timeouts wait-release 3
 connection trunk 3001
!
voice-port 3/0:1
 timeouts wait-release 3
 connection trunk 6001
!
... [channels 2-30 the same] ...
!
voice-port 3/0:30
 timeouts wait-release 3
 connection trunk 6030
```

GTP2

```
!
 voice-port 3/0:0
 timeouts wait-release 3
 connection trunk 5001 answer-mode
!
 voice-port 3/0:1
 timeouts wait-release 3
 connection trunk 8001 answer-mode
!
... [channels 2-30 the same] ...

voice-port 3/0:30
 timeouts wait-release 3
 connection trunk 8030 answer-mode
```

3. On the dial peers:

a. Add a VoIP dial peer that matches the **connection trunk** dialed number of the voice channels. Point it to the IP address of the remote side; assign the desired (or default) voice codec on this dial peer.

b. Add a VoIP dial peer that matches the **connection trunk** dialed number of the signaling channels. Point it to the IP address of the remote side; assign the clear-channel codec on this dial peer.

c. Add POTS dial peers to each voice port that match the number dialed by the **connection trunk** statements from the other side.

GTP1

```
dial-peer voice 8001 pots
 destination-pattern 8001
 port 3/0:1
!


!--- Pots dial peers 8001 --- 8030 are
!--- configured similarly with exclusive POTS,
!--- destination patterns and ports. These are
!--- associated with the voice channels.
```

```
dial-peer voice 8030 pots
 destination-pattern 8030
 port 3/0:30
!
dial-peer voice 5001 pots
 destination-pattern 5001
 port 3/0:0
```

```
!
dial-peer voice 6000 voip
 destination-pattern 6...
 session target ipv4:10.10.105.1
!
dial-peer voice 3001 voip
 answer-address 5001
 destination-pattern 3001
 session target ipv4:10.10.105.1
 codec clear-channel
```

GTP2

```
!
dial-peer voice 6001 pots
 destination-pattern 6001
 port 3/0:1
!
```

```
dial-peer voice 6030 pots
 destination-pattern 6030
 port 3/0:30
!
dial-peer voice 3001 pots
 destination-pattern 3001
 port 3/0:0
```

```
!
dial-peer voice 8000 voip
 destination-pattern 8...
```

```
 session target ipv4:10.10.105.2
!
dial-peer voice 5001 voip
 answer-address 3001
 destination-pattern 5001
 session target ipv4:10.10.105.2
 codec clear-channel


!--- This is the VoIP dial peer associated with
!--- the destination pattern that connects
!--- to the remote PBX signaling channel port.
```

Configuration Steps for the WAN Side

To configure the WAN side, complete these steps:

Put in an **IP RTP Priority** command or LLQ bandwidth based on the following:

- The number of voice channels, and the codecs used for voice signals.
- The number of signaling channels multiplied by 80K (treated as you would treat G.711).

GTP1
| ```
interface Multilink1
 bandwidth 512
 ip address 10.10.105.2 255.255.255.0
 ip tcp header-compression iphc-format
 no cdp enable
 ppp multilink
 ppp multilink fragment-delay 20
 ppp multilink interleave
 multilink-group 1
 ip rtp header-compression iphc-format
 ip rtp priority 16384 16383 384
!
interface Serial0/0
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 multilink-group 1
``` |

GTP2
| ```
interface Multilink1
 bandwidth 512
 ip address 10.10.105.1 255.255.255.0
 ip tcp header-compression iphc-format
 no cdp enable
 ppp multilink
 ppp multilink fragment-delay 20
 ppp multilink interleave
 multilink-group 1
 ip rtp header-compression iphc-format
 ip rtp priority 16384 16383 384
!!
interface Serial1/0
 no ip address
 encapsulation ppp
 no fair-queue
``` |

```
clock rate 512000
ppp multilink
multilink-group 1
```

Troubleshoot and Verify Clear−Channel T−CCS

These steps help to verify that clear−channel T−CSS is operating as it should:

1. The E1 controller must be up for voice ports to go off−hook and trunked.
2. Ensure that check calls are in place, and the correct DSPs are allocated on timeslots.
3. If calls fail to connect, check the IP configuration and connectivity, and dial peer provision.
4. If the IP is restored after an interface or link failure, the controller must have the **shut/no shut** command issued on its interface, or the router must be reloaded to bring trunk connections back up.
5. If the **show voice port** command shows `idle` and `on hook` for any timeslot, check that the related timeslot has the correct DSP version assigned, and that it is working correctly with the **show voice dsp** command, as shown below.

```
gtp#show voice dsp

DSP DSP                   DSPWARE CURR  BOOT             VOICE     PAK        TX/RX
TYPE NUM CH CODEC         VERSION STATE STATE RST AI PORT     TS ABORT   PACK COUNT
==== === == ======= ======= ===== ===== === == ====== == =====   ============
C549 000 02 g729r8   3.4.49 busy  idle       0 3/0:25    25       0      264/2771
C549 000 01 g729r8   3.4.49 busy  idle       0 3/0:12    12       0      264/2825
C549 000 00 clear-ch 3.4.49 busy  idle   0   0 3/0:0     06       0 158036/16069


!--- The above identifies that the clear codec is used for timeslot 6.
!--- Ensure that clear codec is applied correctly against the correct timeslot.




gtp1#show voice port sum

PORT    CH SIG-TYPE    ADMIN OPER STATUS   STATUS   EC
====== == ========== ===== ==== ======== ======== ==
3/0:0  6    ext       up    up   trunked  trunked  y
3/0:1  1    ext       up    up   trunked  trunked  y
3/0:2  2    ext       up    up   trunked  trunked  y
3/0:3  3    ext       up    up   trunked  trunked  y


!--- This shows that the voice port used for signaling is off-hook and trunked.




gtp1#show voice call sum
PORT         CODEC    VAD VTSP STATE          VPM STATE
============ ======== === ============    =============
3/0:0.6      clear-ch y   S_CONNECT          S_TRUNKED
3/0:1.1      g729r8   y   S_CONNECT          S_TRUNKED
3/0:2.2      g729r8   y   S_CONNECT          S_TRUNKED
3/0:3.3      g729r8   y   S_CONNECT          S_TRUNKED
3/0:4.4      g729r8   y   S_CONNECT          S_TRUNKED
3/0:5.5      g729r8   y   S_CONNECT          S_TRUNKED
3/0:6.31     g729r8   y   S_CONNECT          S_TRUNKED
3/0:7.7      g729r8   y   S_CONNECT          S_TRUNKED


!--- This shows a signaling call in progress.
```

Enable RTP Signaling on AS5350 and AS5400

In order to prevent errors caused by RTP packets of payload type á23 on Cisco AS5350 and AS5400 series platforms, RTP signal processing is disabled by default. Under some circumstances, packets of this type can cause an invalid memory address error in AS5350 and AS5400 series platforms, potentially crashing the devices.

On these models, you can enable RTP signal processing using the **voice−fastpath voice−rtp−signalling enable** hidden configuration command. However, before you enable RTP signal processing, prepare the platform to handle RTP packets of payload type á23 by enabling T−CCS.

After you prepare the platform, you can use these commands in order to enable or disable RTP signal processing.

- In order to enable RTP signal processing, use this command:

```
Router(config)#voice-fastpath voice-rtp-signalling enable
```
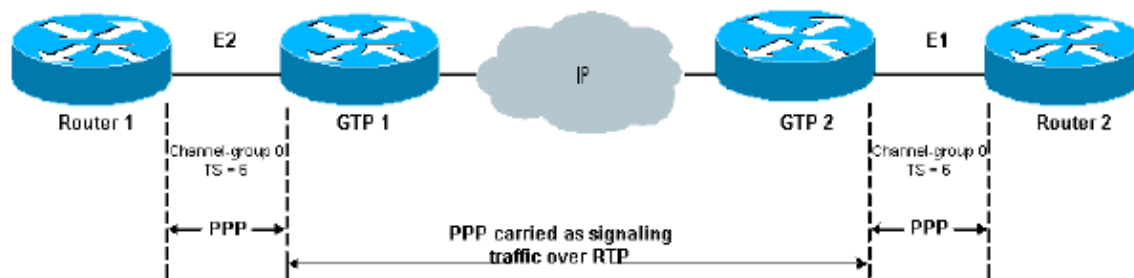- In order to disable RTP signal processing, use this command:

```
Router(config)#no voice-fastpath voice-rtp-signalling enable
```

How to Test T−CCS (Frame−Forwarding and Clear−Channel) Without PBXs

In certain situations it may be impractical to verify the configuration of T−CCS with PBXs. This section describes a method that involves the substitution of the PBXs with routers, to test that signaling can be transported. Because the frame structure used in PPP is similar to that used by message−based signaling (such as CCS), you can use routers configured for PPP to test that the signaling channel is working. This can be useful in situations where the deployment of T−CCS has failed, and further proof is needed that the signaling channel is working. (In frame−forwarding T−CCS there is debug information available showing the transmission and reception of frames. In clear−channel T−CCS, no real−time debug information is available.)

Configure the E1 controller of the routers for the signaling channel of choice. This example uses timeslot 6, to tie−in with the above tests. Configure PPP on the resultant serial interface to represent signaling traffic.

Router 1
| ```
controller E1 0
 clock source internal
 channel-group 0 timeslots 6
!
interface Serial0:0
 ip address 1.1.1.2 255.255.255.0
 encapsulation ppp
``` |

Router 2

```
controller E1 0
 clock source internal
 channel-group 0 timeslots 6
!
interface Serial0:0
 ip address 1.1.1.1 255.255.255.0
 encapsulation ppp
```

Typical Output with `debug ppp packets`

```
1d00h: Se0:0 LCP: Received id 1, sent id 1, line up
1d00h: Se0:0 PPP: I pkt type 0xC021, datagramsize 16
1d00h: Se0:0 LCP: I ECHOREQ [Open] id 2 len 12 magic 0x0676C553
1d00h: Se0:0 LCP: O ECHOREP [Open] id 2 len 12 magic 0x0917B6ED
1d00h: Se0:0 PPP: I pkt type 0x0207, datagramsize 305
1d00h: Se0:0 LCP: O ECHOREQ [Open] id 2 len 12 magic 0x0917B6ED
1d00h: Se0:0 PPP: I pkt type 0xC021, datagramsize 16
1d00h: Se0:0 LCP: I ECHOREP [Open] id 2 len 12 magic 0x0676C553
1d00h: Se0:0 LCP: Received id 2, sent id 2, line up
```

Related Information

- **Voice Hardware: C542 and C549 Digital Signal Processors (DSPs)**
- **Troubleshooting the DSP on NM–HDV for Cisco 2600/3600/VG200 Series Routers**
- **Understanding High Density Voice Network Modules**
- **Voice Technology Support**
- **Voice and Unified Communications Product Support**
- **Troubleshooting Cisco IP Telephony**
- **Technical Support – Cisco Systems**