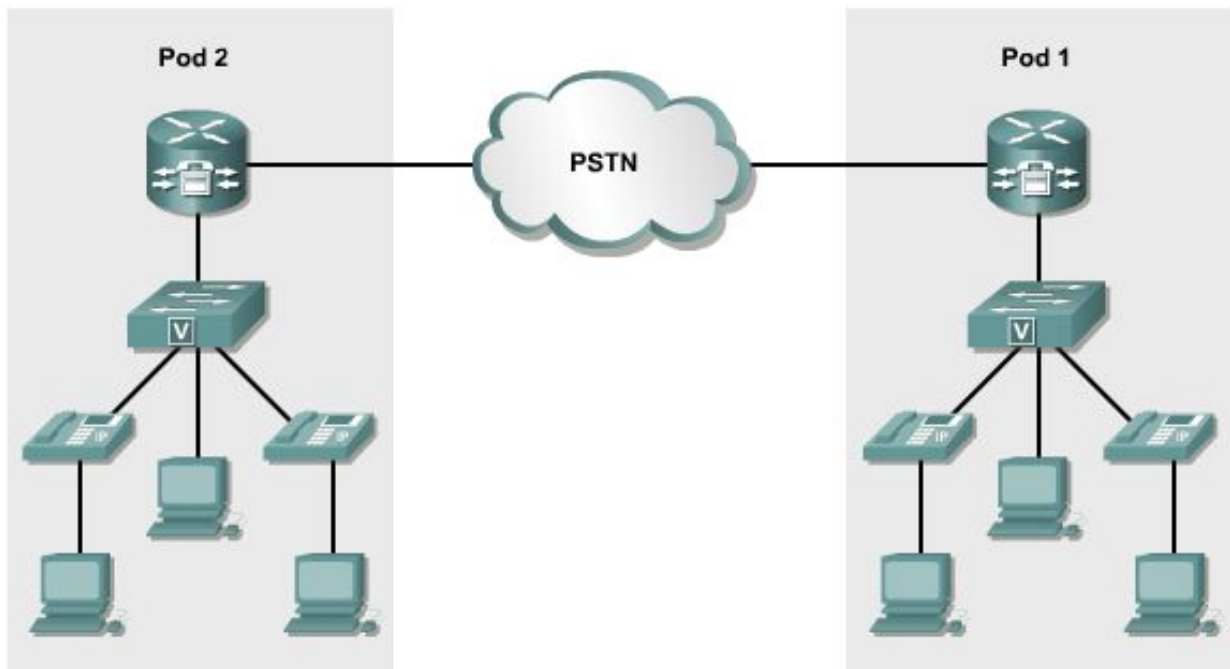


## Lab 4.1.4 Configuring VoIP Dial-Peers Across a WAN Link



### Objective

Configure the VoIP dial peers across a WAN link.

### Equipment Requirements

- Two Cisco CallManager Express (CME) capable routers with a serial port
- Two inline power capable switch or non-inline power switch with power injectors
- Adtran
- Two IP phones
- Two analog phones

This lab relies on labs 2.1.1, 2.1.3, and 3.1.1 being successfully completed and loaded.

In this lab the ACME.com Company has added another site with its own CallManager Express. A WAN connection to the other site will need to be configured and tested.

## Step 1 Configure the Serial Interface

- In this lab, Pod 1 and Pod 2 will partner, and Pod 3 and Pod 4 will partner.
- Ensure that a serial cable connects from the lowest serial interface on *the lowest pod number* router terminating on the other pod's lowest number router serial interface.
- On both pods, use the **show ip interface brief** command to verify that (1) a serial interface is installed and (2) the lowest numbered serial interface.
- What is the port/slot number assigned to the lowest numbered serial interface? \_\_\_\_\_
- On both pods, use the **show controllers serial *mod/port*** for the lowest serial interface and verify if the cable is a DCE or DTE. Examples of this command are **show controllers serial 0/0** or **show controllers serial 0/1/0**.

```
CMERouterX# show controllers serial mod/port
```

- Type the **interface serial *mod/port*** command from global configuration mode to access the serial interface. Examples of this command are **interface serial 0/0** or **interface serial 0/1/0**.

```
CMERouterX(config)# interface serial mod/port
```

- If the pod has the DCE end of the cable, use the command **clock rate 115200** to set the clock rate of the lowest serial interface.

```
CMERouterX(config-if)# clock rate 64000
```

- Use the **encapsulation hdlc** command to configure the serial link encapsulation to HDLC.

```
CMERouterX(config-if)# encapsulation hdlc
```

- Refer to Table 1 for the appropriate IP address to apply to the serial interface.

```
CMERouterX(config-if)# ip address 10.X.0.X 255.255.255.0
```

- Use the command **no shutdown** to enable the serial interface.

```
CMERouterX(config-if)# no shutdown
```

- Wait for your partner pod to complete the previous steps. Verify connectivity by using ping to test the serial connection. Ping the address of your partner pod serial interface. Enter **ping 10.Y.0.X** (refer to Table 1 for the partner pod IP address).
- From an IP phone connected to the pod, attempt to dial a four digit extension numbers of one of the partner pod IP phones.
- What was the result, and why is this result? \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

- What message displayed on the IP phone when the call did not complete? \_\_\_\_\_

## Step 2 Configure the Dial Peer

- From global configuration mode, create a new dial peer with the command **dial-peer voice 6 voip**.

```
CMERouterX(config)# dial-peer voice 6 voip
```

- b. Associate a pattern to the dial peer by using the global configuration mode command **destination-pattern *digits***. Refer to Table 2 for the destination patterns being used. Depending on the pod number, use one of the following commands to configure a destination pattern. Note that the period(s) at the end of the command is part of the command.

Pod 1- **destination-pattern 50[3,4,5].**

Pod 2- **destination-pattern 50[0,1,2].**

Pod 3- **destination-pattern 51[0,1,].**

Pod 4- **destination-pattern 50[6,7,8].**

```
CMERouterX(config-dial-peer)# destination-pattern digits_fm_list
```

- c. Use the **session target ipv4:10.X.0.X** (where the IP address is the partner pod router's serial interface address).

```
CMERouterX(config-dial-peer)# session target ipv4:10.X.0.X
```

- d. The G711 codec is what is used to translate analog voice data to a digital format. G711 uses a 64kbps data rate. Manually configure the codec to be used by entering the command **codec g711ulaw**. This codec will be used by any call that matches this dial peer.

```
CMERouterX(config-dial-peer)# codec g711ulaw
```

### Step 3 Test the Configuration

- a. Dial a four digit extension numbers of one of the phones in your partner's pod and stay connected. The lowest number pod partner should count aloud the numbers one to ten slowly into the phone.
- How is the voice call quality? \_\_\_\_\_
- \_\_\_\_\_
- b. Is the quality of the voice acceptable? \_\_\_\_\_
- c. Coordinating with your partner pod, place a second simultaneous call between the pods using a four digit extension. This will force two calls on the WAN link. The highest number pod partner counts aloud from number one to number 10. After counting, remain connected.
- d. How is the voice quality? Is it the same, better, or worse? \_\_\_\_\_
- \_\_\_\_\_
- e. Verify the codec being used by the call is G.711 by quickly pressing blue *i* or question mark button (depending on the model of phone) on the IP phones twice while the calls are connected.
- f. What does the IP phone display to indicate the codec being used? \_\_\_\_\_
- g. Hang up both calls.

### Step 4 Reconfigure the Codec

- a. From global configuration mode on both pods, access the dial peer with the command **dial-peer voice 6 voip**.

```
CMERouterX(config)# dial-peer voice 6 voip
```

- b. The G729 codec uses an 8kbps data rate. Reconfigure the codec to use the G729 codec by entering the command **codec g729br8**.

```
CMERouterX(config-dial-peer)# codec g729br8
```

- c. Coordinate with the partner pod to place two simultaneous calls across the WAN link by dialing the four digit extensions.
- d. How is the voice quality? Remain connected. \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
- e. On the IP phone, verify the codec is G.729 by quickly pressing the blue *i* or *?* (question mark) button twice (depending on the model of phone) while the calls are still connected.
- f. Save the configuration by using the command **copy running-config startup-config**.