



## **Land Mobile Radio over IP Configuration Guide, Cisco IOS XE Release 3S**

<a href="#">LMR over IP Overview</a>	<b>2</b>
<a href="#">Finding Feature Information</a>	<b>2</b>
<a href="#">Information About Land Mobile Radio</a>	<b>2</b>
<a href="#">How to Configure Land Mobile Radio</a>	<b>12</b>
<a href="#">Additional References for Land Mobile Radio</a>	<b>17</b>
<a href="#">Feature Information for Land Mobile Radio</a>	<b>18</b>

Revised: December 21, 2015,

## LMR over IP Overview

A Land Mobile Radio (LMR) system is a collection of portable and stationary radio units designed to communicate with each other over predefined frequencies. They are deployed wherever organizations need to have instant communication between geographically dispersed and mobile personnel. Typical LMR system users include public safety organizations such as police departments, fire departments, and medical personnel. However, LMR systems also find use in the private sector for activities like construction, building maintenance, and site security.

### Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

### Information About Land Mobile Radio

In typical LMR systems, a central dispatch console or base station controls communications to the disparate handheld or mobile units in the field. The systems might also employ repeaters to extend the range of communications for the mobile users. LMR systems can be as simple as two handheld units communicating between themselves and a base station over preset channels. Or, they can be quite complex, consisting of hundreds of remote units, multiple dispatch consoles, dynamic channel allocation, and other elements.

### LMR over IP Service

With the LMR over IP service, standards-based VoIP technology voice gateways are used in combination with additional LMR specific features to address interoperability, extending command and control, and other issues. Base stations, repeaters, and dispatch consoles generally possess a wired interface that can be used to monitor audio received from their air interface, and as input for audio to be transmitted on their air interface. Although this wired interface may contain other control capabilities as well, as long as it has some sort of speaker output and microphone input, it can be connected to a voice port on a router.

The audio received on the voice port is encoded with a standard audio codec, such as G.711 or G.729. Those audio samples are packaged in standards-based Real-Time Transport Protocol (RTP) packets suitable for transport on an IP network. At this point, the communication element is abstracted from the distinctive characteristics of each radio system, thus providing a solution for the interoperability problem. Now, these audio packets can be sent across the network to other LMR gateways with different brands of radio systems either individually (unicast) or as a group (multicast).

The recipient of the audio packets need not be another LMR gateway. It can be any device capable of receiving and decoding the RTP stream, such as an IP telephone or PC with appropriate software. The IP network and IP-enabled devices can be used to allow users to monitor or transmit on a particular radio channel from a desk without issuing another radio. This can be done locally, nationally, or internationally, assuming the IP network has been properly designed.

## Services and Components

The variety of radio systems, desired participants, and operational needs of an organization cannot be satisfied by one Land Mobile Radio (LMR) over IP architecture. So, instead of having an omnibus architecture, LMR over IP is broken down into a series of services. Some of these services are implemented by means of the Cisco gateway routers running Cisco IOS software with the LMR over IP feature set. Some services employ Cisco IP telephony equipment such as Cisco IP Phones and Cisco CallManager.

The following sections outline these LMR over IP services, providing a description of how and when they may be used to achieve a more efficient, elegant, or scalable solution for LMR needs. It is important to understand that these tools are not exclusive of each other. In many cases they can be quite complementary.

### Interoperability—The Radio Interface Adaption Layer

The foundation on which all LMR services are built is the interface joining the LMR radio systems to the IP network. The key to implementing all these other services is to adapt the disparate LMR endpoints to a common standard. The radio systems are connected through available wired connection points and not through their air interface. The radios are connected to the LMR-enabled Cisco gateway routers through either an analog ear and mouth (E&M) interface or a digital T1 interface. LMR-specific enhancements to Cisco IOS software provide greater flexibility in controlling audio levels, tuning voice activity detection (VAD), and improving the ability of the router to interact with those radios that employ physical signaling. For those radio systems that utilize in-band tones for signaling, additional hardware such as third-party tone remote units are required to adapt the radio to the IP environment.

With all participants using the same communications structure, any-to-any communication is now possible. Tone controlled systems can communicate with systems using other tone schemes, or systems using physical signaling. Organizations can merge different radio systems onto one common backbone, and they are freed from being tied down to one particular vendor or solution for their radio needs.

### Transport

Once the radio system is connected to the network, the audio transmission needs to be sent to and from other endpoints. These three mechanisms for accomplishing this task are discussed in the following sections:

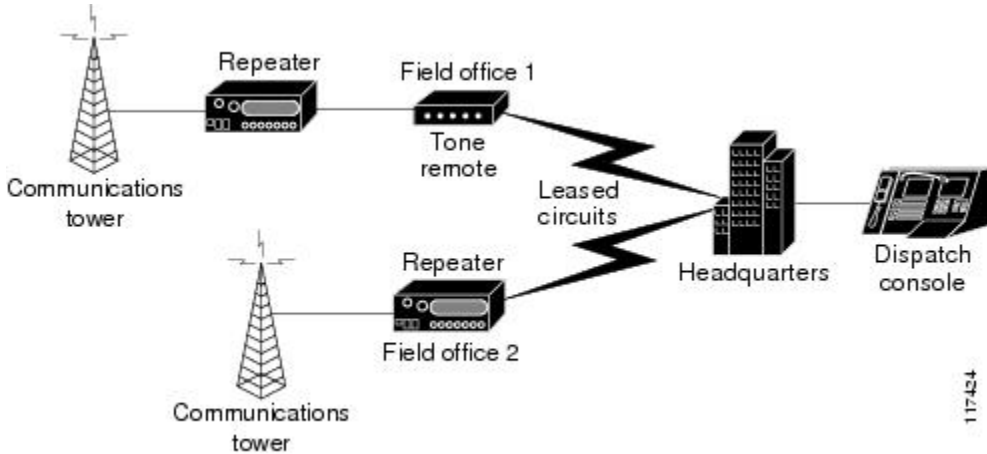
#### Unicast Connection Trunk (Leased Line Replacement)

In LMR deployments topography, distance or other environmental factors can limit the coverage of the network. In some situations, leased lines or other dedicated point-to-point transmission facilities are used to connect geographically remote devices. By using a unicast connection trunk configuration on the LMR gateway, organizations can leverage standard IP connectivity over either the public Internet, or a private network to backhaul their LMR traffic and provide data connectivity at these remote sites. In this manner, the organization can achieve cost savings through either reduced facility charges or a reduction in the number of required connections at the site.

The unicast connection trunk service on the LMR gateway provides a permanent point-to-point connection between two voice endpoints. It uses standard H.323 signaling to establish the VoIP circuit between the gateways. This circuit is capable of not only sending audio information using standard Real-Time Transport Protocol (RTP) datagrams, but of physical lead state signaling as well.

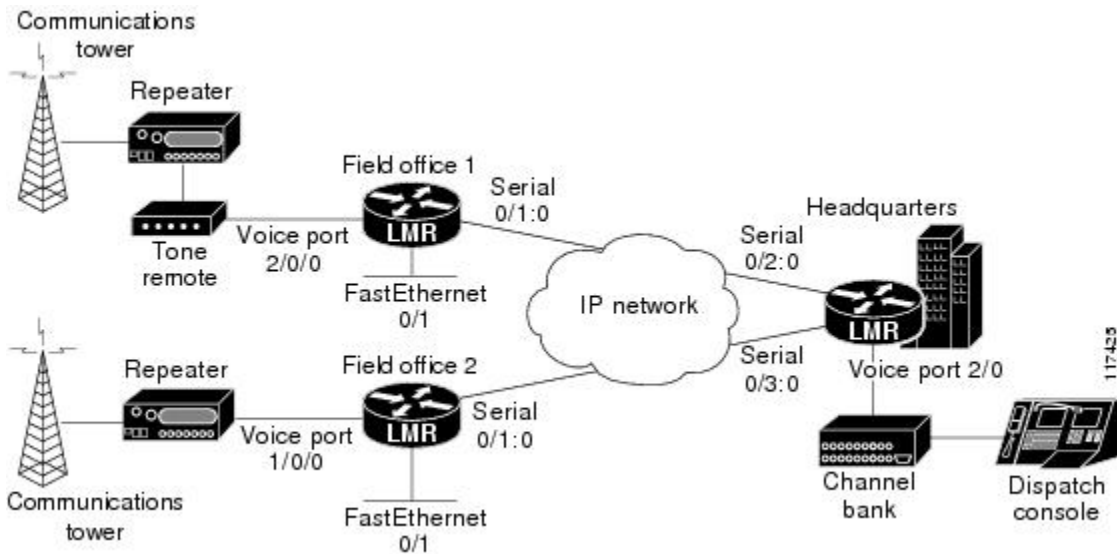
The following figure shows a representative topology for a traditional configuration with a dispatch console in headquarters connecting to repeaters in the field offices over leased circuits.

**Figure 1: Traditional Leased Line Implementation**



The following figure shows connection for each of the LMR components to an LMR gateway and the gateways connected through an IP cloud. The same circuits are used to carry both our LMR audio and any data communications.

**Figure 2: Unicast Connection Trunk Leased Line Replacement**



**Connection PLAR**

The connection private line, automatic ringdown (PLAR) service is a variant of the unicast connection trunk service. PLAR circuits are switched connections between statically configured endpoints. As a switched call, the PLAR connection will be torn down upon certain events such as the calling party going on-hook, or an absence of voice packets for a preconfigured amount of time. Whereas the unicast connection trunk VoIP call is made once at session initialization, with a PLAR connection, we can initiate the VoIP call

based on signaling information from the LMR endpoint. The call will stay active while there is audio on the connection and for a configurable amount of time afterward, and then it will get torn down.

Although some bandwidth savings is achieved by only maintaining the connection while voice is present on the circuit, the main benefit of connection PLAR is its ability to connect with other H.323 capable devices. The unicast connection trunk connection uses special signaling packets to pass the signaling end-to-end between devices. Other devices capable of setting up or receiving H.323 calls will not recognize these packets and thus will not be able to establish a connection. Connection PLAR offers a way to connect to these other H.323 devices.

## Gateway to Gateway Connections: Transport

### Connection Trunk (Unicast)

In general, a trunk, or tie line, is a permanent point-to-point communication link between two voice endpoints. From a Cisco IOS software perspective, the unicast **connection trunk** command creates a permanent VoIP call between two VoIP gateways. It simulates a trunk connection by creating virtual trunk tie lines between two telephony endpoints. To the connected systems, it appears as if a T1 trunk is directly connecting them.

The key features of unicast connection trunk connections are:

- Permanent connection—always up
- Transports signaling end-to-end
- Can be used only with other Cisco IOS software-based devices supporting unicast connection trunk

From an LMR perspective, a typical application for the unicast connection trunk would be as a replacement for leased line connections. Instead of dedicating an entire circuit for LMR traffic, the circuit could be configured for IP transmission with the LMR traffic and other data traffic sharing the same line.

Notice that we continually qualify the connection trunk as either unicast or multicast. To the voice endpoints, both types present the same permanent tie-line style connection. The difference lies in how that connection is implemented across the IP network. The unicast version employs an H.323 connection setup mechanism to establish a channel to another voice gateway and directs the Real-Time Transport Protocol (RTP)/RTP Control Protocol (RTP/RTCP) datastream to that single endpoint. The multicast version directs the RTP/RTCP datastream to an IP multicast address so it can be received by any number of endpoints. Aside from the protocols used for multicast routing, no other telephony connection setup mechanism is used.

How it works

The stages in the life of a unicast trunk connection between two voice interfaces are as follows:

- In this stage, we actually build the permanent connection between the endpoints. The gateway routers:
  - Establish an H.323 connection.
  - Perform H.225.0 session negotiation.
  - Synchronize the endpoints.
- Once the connection is established, the endpoints sit in an inactive state sending connection maintenance frames, until either end decides to send data, which is the active state.
  - Inactive state:
    - RTCP sender and receiver reports are sent.
    - Signaling keepalives are sent.

- Active state:
  - RTCP sender and receiver reports are sent.
  - Signaling transitions are sent.
  - RTP voice frames are sent.
- Because this is a permanent connection, it will stay up as long as the gateways, their respective voice interfaces, and the IP connection between them stay up.

## Connection PLAR

Private line automatic ringdown (PLAR) circuits are switched connections between statically configured voice endpoints. The endpoints have the destination address of the connection preconfigured and so do not require user dialing to connect calls. PLAR connections are those in which a phone goes off-hook and a remote phone rings without digits being dialed. As switched calls, the connections will be torn down on certain events; for example, the calling party goes on-hook, and the resources made available for other connections.

The following are the main similarities and differences between connection PLAR mode and connection trunk (unicast) mode:

- Connection trunk mode is a permanent connection; the VoIP call is always connected independently of the associated voice port being on-hook or off-hook.
- Connection PLAR mode is a switched VoIP call; the call is set up on an as-needed basis. With connection PLAR, no bandwidth is consumed while the device connected to the associated voice port is on-hook. When a device is taken off-hook, the call is automatically connected, and the remote device is signaled to go off-hook.
- Both connection trunk (unicast) and connection PLAR modes have statically configured endpoints and do not require user dialing to connect calls.
- Connection trunk (unicast) mode allows supplemental call signaling to be passed over the IP network between the two telephony devices. Connection PLAR transports only limited types of signaling.
- Connection PLAR does not use a proprietary keepalive mechanism and is thus can establish connections to other H.323 capable endpoints.

From an LMR standpoint, PLAR connections would be employed in those leased line replacement scenarios in which the overhead of a connection trunk (unicast) connection was not needed or desired. PLAR connections would also be useful for connecting to non-Cisco IOS software-based devices that support H.323 call setup mechanisms.

### How it works

The stages in the life of a PLAR connection between two voice interfaces are as follows:

- In this stage, we actually build the permanent connection between the endpoints. The gateway routers:
  - Establish an H.323 connection.
  - Perform H.225.0 session negotiation.
  - Synchronize the endpoints.
- Once the connection is established, the endpoints sit in an inactive state sending connection maintenance frames, until either end decides to send data, which is the active state.
  - Inactive state. RTCP sender and receiver reports are sent.

- Active state:

- RTCP sender and receiver reports are sent.
- RTP voice frames are sent.

- For Cisco IOS software-based LMR gateways, when the party initiating the call goes on-hook, which is the M-lead idle state, it starts the call teardown timer on that gateway. For the LMR gateway receiving the call, as soon as it stops receiving voice frames, it starts its call teardown timer. The gateway whose timer expires first initiates the actual call teardown.

### Connection Teardown

Unlike the unicast connection trunk mode in which the connection stays up as long as the voice ports and underlying IP circuit are up, with connection PLAR, the voice call can be torn down from either side. Each gateway participating in the connection maintains a teardown timer. If there is an active connection, the teardown timer begins running whenever the interface M-lead is in an idle state and whenever no voice frames are received from the network. The length of this timer (in seconds) is set using the following command on the voice port:

```
timeouts teardown lmr {<5-60000> | infinity}
```

We will illustrate the operation of the teardown timer using the output of the **debug vpm signal** command. First, we examine the behavior on the calling router. When the M-lead is raised, we see the following debug messages:

```
lmr-4000#
Jan 29 16:42:34.566 EST: htsp_process_event: [1/0:0(1), LMR_ONHOOK, E_DSP_SIG_1100] lmr_onhook_offhook
Jan 29 16:42:34.566 EST: htsp_timer_stop htsp_setup_ind
Jan 29 16:42:34.566 EST: [1/0:0(1)] get_local_station_id calling num= calling name= calling time=01/29 16:42
orig called=
Jan 29 16:42:34.566 EST: htsp_timer - 3000 msec
Jan 29 16:42:34.570 EST: htsp_process_event: [1/0:0(1), LMR_WAIT_SETUP_ACK, E_HTSP_SETUP_ACK]
lmr_wait_setup_ack_get_ack
Jan 29 16:42:34.570 EST: htsp_timer_stop
Jan 29 16:42:34.570 EST: htsp_process_event: [1/0:0(1), LMR_OFFHOOK, E_HTSP_PROCEEDING]
htsp_progress_notifyhtsp_call_bridged
Jan 29 16:42:34.582 EST: htsp_process_event: [1/0:0(1), LMR_OFFHOOK, E_HTSP_VOICE_CUT_THROUGH]
lmr_offhook_voice_cut
Jan 29 16:42:34.582 EST: htsp_timer_stop
Jan 29 16:42:34.582 EST: htsp_process_event: [1/0:0(1), LMR_OFFHOOK, E_HTSP_CONNECT] lmr_offhook_connect
Jan 29 16:42:34.582 EST: htsp_timer_stop
Jan 29 16:42:34.582 EST: htsp_timer_stop2
```

The last line in bold indicates that the teardown timer has stopped running. As long as the calling gateway keeps the M-lead raised, the timer will not start. On the called side, the debug output from the same call appears as follows:

```
Jan 29 16:42:34.576 EST: htsp_timer_stop3 htsp_setup_req
Jan 29 16:42:34.576 EST: htsp_process_event: [1/0/0, LMR_ONHOOK, E_HTSP_SETUP_REQ] lmr_onhook_setup
Jan 29 16:42:34.576 EST: htsp_timer_stop htsp_progress
Jan 29 16:42:34.576 EST: lmr_start_timer: 2000 ms
Jan 29 16:42:34.576 EST: htsp_timer - 2000 msec htsp_call_bridged
Jan 29 16:42:34.580 EST: htsp_process_event: [1/0/0, LMR_WAIT_CUT_THRU, E_HTSP_VOICE_CUT_THROUGH] lmr_cut_thru

Jan 29 16:42:34.580 EST: htsp_timer_stop
Jan 29 16:42:34.580 EST: lmr_pak_suppress_enable FALSE
Jan 29 16:42:34.580 EST: lmr_start_timer2: 25 second
Jan 29 16:42:34.580 EST: htsp_timer2 - 25000 msec
Jan 29 16:42:34.580 EST: htsp_process_event: [1/0/0, LMR_CONNECT, E_DSP_SIG_0000] lmr_conn_onhook
Jan 29 16:42:34.580 EST: htsp_timer_stop
Jan 29 16:42:34.580 EST: lmr_pak_suppress_enable TRUE
Jan 29 16:42:34.580 EST: lmr_start_timer2: 25 second
Jan 29 16:42:34.580 EST: htsp_timer2 - 25000 msec
Jan 29 16:42:34.604 EST: htsp_process_event: [1/0/0, LMR_CONNECT, E_HTSP_V_PAK_RCVD] lmr_conn_pkt_rcvd
Jan 29 16:42:34.604 EST: htsp_timer_stop2 lmr_offhook (0)
Jan 29 16:42:34.604 EST: [1/0/0] set signal state = 0x8 timestamp = 0
Jan 29 16:42:36.888 EST: htsp_process_event: [1/0/0, LMR_CONNECT, E_HTSP_V_PAK_STOP] lmr_conn_pkt_stop lmr_onhook
(0)
```

```

Jan 29 16:42:36.888 EST: [1/0/0] set signal state = 0x0 timestamp = 0
Jan 29 16:42:36.888 EST: lmr_start_timer2: 25 second
Jan 29 16:42:36.888 EST: htsp_timer2 - 25000 msec

```

In the previous lines in bold, we actually see the timer start three times. The timeout value for this voice port is set at 25 seconds. Because we have not raised the M-lead on this side of the connection, we wholly depend on the presence of voice frames to stop the timer from running. We see this timer transition from start to stop and to start again during the two-second voice spurt sent when the call is made. When the called gateway gets another talk spurt a few seconds later, the following debug messages appear:

```

Jan 29 16:42:56.421 EST: htsp_process_event: [1/0/0, LMR_CONNECT, E_HTSP_V_PAK_RCVD] lmr_conn_pkt_rcvd
Jan 29 16:42:56.421 EST: htsp_timer_stop2 lmr_offhook (0)
Jan 29 16:42:56.421 EST: [1/0/0] set signal state = 0x8 timestamp = 0

```

When the called gateway gets the voice frames and goes off-hook, the teardown timer stops counting as shown in the bold line of the previous debug messages. When the talk spurt ends and the called gateway goes on-hook, we see the following:

```

Jan 29 16:43:30.890 EST: htsp_process_event: [1/0/0, LMR_CONNECT, E_HTSP_V_PAK_STOP] lmr_conn_pkt_stoplmr_onhook (0)
Jan 29 16:43:30.890 EST: [1/0/0] set signal state = 0x0 timestamp = 0
Jan 29 16:43:30.890 EST: lmr_start_timer2: 25 second
Jan 29 16:43:30.890 EST: htsp_timer2 - 25000 msec

```

If 25 seconds elapse since the called gateway went on-hook, the timer expires, and the call is torn down. On the called gateway, the following debug messages appear:

```

Jan 29 16:44:18.888 EST: htsp_process_event: [1/0/0, LMR_CONNECT, E_HTSP_V_PAK_STOP] lmr_conn_pkt_stoplmr_onhook (0)
Jan 29 16:44:18.888 EST: [1/0/0] set signal state = 0x0 timestamp = 0
Jan 29 16:44:18.888 EST: lmr_start_timer2: 25 second
Jan 29 16:44:18.888 EST: htsp_timer2 - 25000 msec
lmr-4300#
lmr-4300#
lmr-4300#
Jan 29 16:44:43.889 EST: htsp_process_event: [1/0/0, LMR_CONNECT, E_HTSP_EVENT_TIMER2] lmr_offhook_timer
Jan 29 16:44:43.889 EST: htsp_timer_stop2
Jan 29 16:44:43.889 EST: htsp_timer_stop3
Jan 29 16:44:43.889 EST: htsp_process_event: [1/0/0, LMR_CONNECT, E_HTSP_RELEASE_REQ] lmr_conn_release
Jan 29 16:44:43.889 EST: htsp_timer_stop
Jan 29 16:44:43.889 EST: htsp_timer_stop2 lmr_onhook (0)
Jan 29 16:44:43.889 EST: [1/0/0] set signal state = 0x0 timestamp = 0

```

On the calling gateway, the following debug messages appears:

```

lmr-4000#
Jan 29 16:44:43.888 EST: htsp_timer_stop3
Jan 29 16:44:43.892 EST: htsp_process_event: [1/0:0(1), LMR_CONNECT, E_HTSP_RELEASE_REQ] lmr_conn_release
Jan 29 16:44:43.892 EST: htsp_timer_stop
Jan 29 16:44:43.892 EST: htsp_timer_stop2 lmr_onhook (0)vnm_dsp_set_sig_state:[recEive and transMit1/0:0(1)]
set signal state = 0x0

```

If the called side had raised its M-lead at any time during the call, we would have seen messages similar to the bold lines in the previous debug messages indicating that timer had stopped.

The table below shows the disconnect mechanism from a frame trace perspective.



**Table 1: Connection PLAR Disconnect Frames**

Frame	Time (sec)	Source	Destination	Protocol	Src Port	Dest Port	Length	Information
119	21.02002	10.40.0.2	10.40.0.1	RTP	19508	16454	214	Payload type=ITU-T G.711 PCMU, <del>SSRC=2837536</del> , Seq=6797, Time=105867, Mark
120	21.03996	10.40.0.2	10.40.0.1	RTP	19508	16454	214	Payload type=ITU-T G.711 PCMU, <del>SSRC=2837536</del> , Seq=6798, Time=106027
121	21.0599	10.40.0.2	10.40.0.1	RTP	19508	16454	214	Payload type=ITU-T G.711 PCMU, <del>SSRC=2837536</del> , Seq=6799, Time=106187
122	21.07991	10.40.0.2	10.40.0.1	RTP	19508	16454	214	Payload type=ITU-T G.711 PCMU, <del>SSRC=2837536</del> , Seq=6800, Time=106347
600	30.51938	10.40.0.2	10.40.0.1	RTP	19508	16454	214	Payload type=ITU-T G.711 PCMU, <del>SSRC=2837536</del> , Seq=7272, Time=181867

Frame	Time (sec)	Source	Destination	Protocol	Src Port	Dest Port	Length	Information
601	30.53936	10.40.0.2	10.40.0.1	RTP	19508	16454	214	Payload type=ITU-T G.711 PCMU, SSRC= <del>2837536</del> , Seq=7273, Time=182027
602	30.55936	10.40.0.2	10.40.0.1	RTP	19508	16454	214	Payload type=ITU-T G.711 PCMU, SSRC= <del>2837536</del> , Seq=7274, Time=182187
603	30.57934	10.40.0.2	10.40.0.1	RTP	19508	16454	214	Payload type=ITU-T G.711 PCMU, SSRC= <del>2837536</del> , Seq=7275, Time=182347
604	30.59933	10.40.0.2	10.40.0.1	RTP	19508	16454	214	Payload type=ITU-T G.711 PCMU, SSRC= <del>2837536</del> , Seq=7276, Time=182507
605	30.6093	10.40.0.2	10.40.0.1	RTP	19508	16454	60	Payload type=Unknown (19), SSRC= <del>2837536</del> , Seq=7277, Time=182667
607	34.88555	10.40.0.1	10.40.0.2	RTCP	16455	19509	150	Receiver Report
608	34.95209	10.40.0.2	10.40.0.1	RTCP	19509	16455	146	Sender Report
611	38.12519	10.40.0.2	10.40.0.1	RTCP	19509	16455	126	Receiver Report

Frame	Time (sec)	Source	Destination	Protocol	Src Port	Dest Port	Length	Information
613	40.49343	10.40.0.1	10.40.0.2	RTCP	16455	19509	126	Receiver Report
614	43.53205	10.40.0.2	10.40.0.1	RTCP	19509	16455	126	Receiver Report
615	43.90542	10.40.0.1	10.40.0.2	RTCP	16455	19509	126	Receiver Report
617	46.89014	10.40.0.1	10.40.0.2	RTCP	16455	19509	126	Receiver Report
618	49.03379	10.40.0.2	10.40.0.1	RTCP	19509	16455	126	Receiver Report
619	49.85	10.40.0.1	10.40.0.2	RTCP	16455	19509	126	Receiver Report
621	54.71046	10.40.0.2	10.40.0.1	RTCP	19509	16455	126	Receiver Report
622	55.89928	10.40.0.1	10.40.0.2	H.225.0	1720	11000	104	CS: releaseComplete
623	55.90063	10.40.0.2	10.40.0.1	H.225.0	11000	1720	104	CS: releaseComplete
624	55.90117	10.40.0.2	10.40.0.1	RTCP	19509	16455	86	Receiver Report
625	55.90234	10.40.0.1	10.40.0.2	RTCP	16455	19509	86	Receiver Report
626	55.90255	10.40.0.2	10.40.0.1	ICMP			70	Destination unreachable
627	56.09722	10.40.0.1	10.40.0.2	TCP	1720	11000	60	1720 > 11000 [ACK] Seq=150783408 Ack=4291627661 Win=3771 Len=0

Either side may send voice traffic over the connection. In the table above, note that there are no signaling frames sent advising the other side of lead status changes. In frames 119 through 604, the calling gateway sends out a stream of voice frames. Again, when the stream is finished, we see the RTP type 19 (comfort noise) frame in frame 605. For the next 25 seconds we see the expected handshake of RTCP sender and receiver report frames in frames 607 through 621.

Both sides send final RTCP receiver report messages in frames 624 and 625. It is interesting to note that the calling gateway has closed out its connection by the time it gets the receiver report from the called gateway. Thus, the calling gateway sends an Internet Control Message Protocol (ICMP) destination/port unreachable message to the called gateway. Finally, the TCP stack on both gateways send periodic acknowledgments on the H.323 connection every minute for the next 10 minutes until the TCP connection itself is finally torn down.

## How to Configure Land Mobile Radio

### Configuring Connection Trunk Unicast

Different trunk-conditioning signaling attributes may be required to match the characteristics of the different PBXs to which the router connects. For this reason, trunk-conditioning attributes are configured by creating a voice class for each set of attributes required. The trunk-conditioning attributes are configured for the voice class and the voice class is assigned to one or more dial peers.

A voice class must be configured and assigned to at least one dial peer before the trunk conditioning signaling attributes take effect.



**Note** This configuration supports the North America CAS Protocol and applies only to Cisco private-line or FRF.11 trunk calls. It does not apply to digital T1/E1 trunks using CCS.

To create a voice class and define the trunk-conditioning attributes, use the following commands beginning in global configuration mode:

#### Procedure

	Command or Action	Purpose
<b>Step 1</b>	Router(config)# <b>voice class permanent tag</b>	Creates a voice class. The tag number range is from 1 to 10000, and it must be unique on the router.  <b>Note</b> The <b>voice-class</b> command in dial-peer configuration mode is entered with a hyphen. The <b>voice class</b> command in global configuration mode is entered without the hyphen.
<b>Step 2</b>	Router(config-voice-class)# <b>signal keepalive seconds</b>	(Optional) Defines the keepalive signaling packet interval. The seconds range is from 1 to 65535; the default is 5.
<b>Step 3</b>	Router(config-voice-class)# <b>{no-action   idle-only   oos-only   both}</b>	(Optional) Sets the signaling pattern (when the far-end keepalive message is lost or when AIS is received from the far end). The keywords are as follows: <ul style="list-style-type: none"> <li>• no-action—Sends no signaling pattern.</li> <li>• idle-only or oos-only—Sends only one signaling pattern.</li> <li>• both—Restores the default (both signaling patterns are sent).</li> </ul> <b>Note</b> The no form of the command restores the default also.
<b>Step 4</b>	Router(config-voice-class)# <b>signal pattern {idle receive   idle transmit   oos receive   oos transmit} bit-pattern</b>	(Optional) Overrides the default values for the idle and receive OOS patterns or configures OOS transmit signaling patterns. The keywords and argument are as follows: <ul style="list-style-type: none"> <li>• idle receive—Defines the signaling pattern for an idle message from the network and the signaling pattern to be sent to the PBX if the network trunk is OOS and</li> </ul>

	Command or Action	Purpose
		<p>signal sequence oos idle-only or signal sequence oos are configured. The defaults are:</p> <ul style="list-style-type: none"> <li>◦ For near-end E&amp;M—0000 (for T1) or 0001 (for E1)</li> <li>◦ For near-end FXO loop start—0101</li> <li>◦ For near-end FXO ground start—1111</li> <li>◦ For near-end FXS—0101</li> <li>◦ For near-end MELCAS—1101</li> </ul>
<b>Step 5</b>		<ul style="list-style-type: none"> <li>• idle transmit—Defines the signaling pattern for an idle message from the PBX. The defaults are: <ul style="list-style-type: none"> <li>◦ For near-end E&amp;M—0000</li> <li>◦ For near-end FXO—0101</li> <li>◦ For near-end FXS loop start—0101</li> <li>◦ For near-end FXS ground start—1111</li> <li>◦ For near-end MELCAS—1101</li> </ul> </li> </ul>
<b>Step 6</b>		<ul style="list-style-type: none"> <li>• oos receive—Defines the OOS signaling pattern to be sent to the PBX if the network trunk is OOS and signal sequence oos oos-only or signal sequence oos are configured. The defaults are: <ul style="list-style-type: none"> <li>◦ For near-end E&amp;M—1111</li> <li>◦ For near-end FXO loop start—1111</li> <li>◦ For near-end FXO ground start—0000</li> <li>◦ For near-end FXS loop start—1111</li> <li>◦ For near-end FXS ground start—0101</li> <li>◦ For near-end MELCAS—1111</li> </ul> </li> <li>• oos transmit—Defines the signaling pattern for an OOS message from the PBX. There are no default signaling patterns defined.</li> <li>• bit-pattern—Defines the ABCD bit pattern. Valid values are from 0000 to 1111.</li> </ul> <p>The receive signal pattern comes from the data network side to the PBX. The transmit signal pattern comes from the PBX to the data network side. The range for all options is from 0000 to 1111.</p> <p>Repeat the command entry for each signal pattern required.</p>

	Command or Action	Purpose
<b>Step 7</b>	Router(config-voice-class)# <b>signal timing oos timeout</b> { <i>seconds</i>   <b>disabled</b> }	(Optional) Changes the timeout period for asserting a receive OOS pattern to the PBX when signaling packets are lost. This action changes the delay time before a busyout is sent to the PBX. The keyword and argument are as follows: <ul style="list-style-type: none"> <li>• <b>seconds</b>—Defines the delay duration between the loss of signaling packets and the beginning of the OOS state. The range is from 1 to 65535. The default is 30.</li> <li>• <b>disabled</b>—Deactivates the detection of packet loss. If no signaling packets are received from the network, the router does not send an OOS pattern to the PBX and it continues sending voice packets. Use this option to disable busyout to the PBX.</li> </ul>
<b>Step 8</b>	Router(config-voice-class)# <b>signal timing oos restart</b> <i>seconds</i>	(Optional) Configures permanent voice connections to be restarted after the trunk has been OOS for a specified time. The default is no signal timing OOS pattern parameters are configured. <p><b>Note</b> This command has no effect if <b>signal timing oos timeout</b> is set to disabled.</p>
<b>Step 9</b>	Router(config-voice-class)# <b>signal timing oos slave-standby</b> <i>seconds</i>	(Optional) Configures a slave port to return to its initial standby state after the trunk has been OOS for a specified time. The default is no signal timing OOS pattern parameters are configured. <p><b>Note</b> This command has no effect if <b>signal timing oos timeout</b> is set to disabled.</p>
<b>Step 10</b>	Router(config-voice-class)# <b>signal timing oos {suppress-all</b>   <b>suppress-voice}</b> <i>seconds</i>	(Optional) Configures the router or concentrator to stop sending voice packets or voice and signaling packets to the network if it detects a transmit OOS signaling pattern from the PBX for a specified time. The default is no signal timing OOS pattern parameters are configured. <p><b>Note</b> An OOS transmit signaling pattern must be configured with the <b>signal pattern oos transmit</b> command (see Step 4).</p>
<b>Step 11</b>	Router(config-voice-class)# <b>signal timing idle</b> <b>suppress-voice</b> <i>seconds</i>	(Optional) Configures the router or concentrator to stop sending voice packets after the trunk has been idle for a specified time. The default is no signal timing OOS pattern parameters are configured.
<b>Step 12</b>	Router(config-dial-peer)# <b>voice-class permanent</b> <i>tag</i>	Assigns the voice class to the dial peer. <ul style="list-style-type: none"> <li>• <b>tag</b>—Specifies the unique number. The valid range is from 1 to 10000.</li> </ul> <p><b>Note</b> The <b>voice-class</b> command in dial-peer configuration mode is entered with a hyphen. The <b>voice class</b> command in global configuration mode is entered without the hyphen.</p>

## Configuring Connection PLAR

Different trunk-conditioning signaling attributes may be required to match the characteristics of the different PBXs to which the router connects. For this reason, trunk-conditioning attributes are configured by creating a voice class for each set of attributes required. The trunk-conditioning attributes are configured for the voice class and the voice class is assigned to one or more dial peers.

A voice class must be configured and assigned to at least one dial peer before the trunk conditioning signaling attributes take effect.



**Note** This configuration supports the North America CAS Protocol and applies only to Cisco private-line or FRF.11 trunk calls. It does not apply to digital T1/E1 trunks using CCS.

To create a voice class and define the trunk-conditioning attributes, use the following commands beginning in global configuration mode:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	Router(config)# <b>voice class permanent tag</b>	Creates a voice class. The tag number range is from 1 to 10000, and it must be unique on the router.  <b>Note</b> The <b>voice-class</b> command in dial-peer configuration mode is entered with a hyphen. The <b>voice class</b> command in global configuration mode is entered without the hyphen.
<b>Step 2</b>	Router(config-voice-class)# <b>signal keepalive seconds</b>	(Optional) Defines the keepalive signaling packet interval. The seconds range is from 1 to 65535; the default is 5.
<b>Step 3</b>	Router(config-voice-class)# { <b>no-action</b>   <b>idle-only</b>   <b>oos-only</b>   <b>both</b> }	(Optional) Sets the signaling pattern (when the far-end keepalive message is lost or when AIS is received from the far end). The keywords are as follows:  <ul style="list-style-type: none"> <li>• no-action—Sends no signaling pattern.</li> <li>• idle-only or oos-only—Sends only one signaling pattern.</li> <li>• both—Restores the default (both signaling patterns are sent).</li> </ul> <b>Note</b> The no form of the command restores the default also.
<b>Step 4</b>	Router(config-voice-class)# <b>signal pattern {idle receive   idle transmit   oos receive   oos transmit} bit-pattern</b>	(Optional) Overrides the default values for the idle and receive OOS patterns or configures OOS transmit signaling patterns. The keywords and argument are as follows:  <ul style="list-style-type: none"> <li>• idle receive—Defines the signaling pattern for an idle message from the network and the signaling pattern to be sent to the PBX if the network trunk is OOS and signal sequence oos idle-only or signal sequence oos are configured. The defaults are: <ul style="list-style-type: none"> <li>◦ For near-end E&amp;M—0000 (for T1) or 0001 (for E1)</li> <li>◦ For near-end FXO loop start—0101</li> <li>◦ For near-end FXO ground start—1111</li> <li>◦ For near-end FXS—0101</li> <li>◦ For near-end MELCAS—1101</li> </ul> </li> </ul>
<b>Step 5</b>		<ul style="list-style-type: none"> <li>• idle transmit—Defines the signaling pattern for an idle message from the PBX. The defaults are: <ul style="list-style-type: none"> <li>◦ For near-end E&amp;M—0000</li> </ul> </li> </ul>

	Command or Action	Purpose
		<ul style="list-style-type: none"> <li>◦ For near-end FXO—0101</li> <li>◦ For near-end FXS loop start—0101</li> <li>◦ For near-end FXS ground start—1111</li> <li>◦ For near-end MELCAS—1101</li> </ul>
<b>Step 6</b>		<ul style="list-style-type: none"> <li>• oos receive—Defines the OOS signaling pattern to be sent to the PBX if the network trunk is OOS and signal sequence oos oos-only or signal sequence oos are configured. The defaults are: <ul style="list-style-type: none"> <li>◦ For near-end E&amp;M—1111</li> <li>◦ For near-end FXO loop start—1111</li> <li>◦ For near-end FXO ground start—0000</li> <li>◦ For near-end FXS loop start—1111</li> <li>◦ For near-end FXS ground start—0101</li> <li>◦ For near-end MELCAS—1111</li> </ul> </li> <li>• oos transmit—Defines the signaling pattern for an OOS message from the PBX. There are no default signaling patterns defined.</li> <li>• bit-pattern—Defines the ABCD bit pattern. Valid values are from 0000 to 1111.</li> </ul> <p>The receive signal pattern comes from the data network side to the PBX. The transmit signal pattern comes from the PBX to the data network side. The range for all options is from 0000 to 1111.</p> <p>Repeat the command entry for each signal pattern required.</p>
<b>Step 7</b>	Router(config-voice-class)# <b>signal timing oos timeout</b> { <i>seconds</i>   <b>disabled</b> }	<p>(Optional) Changes the timeout period for asserting a receive OOS pattern to the PBX when signaling packets are lost. This action changes the delay time before a busyout is sent to the PBX. The keyword and argument are as follows:</p> <ul style="list-style-type: none"> <li>• <i>seconds</i>—Defines the delay duration between the loss of signaling packets and the beginning of the OOS state. The range is from 1 to 65535. The default is 30.</li> <li>• <b>disabled</b>—Deactivates the detection of packet loss. If no signaling packets are received from the network, the router does not send an OOS pattern to the PBX and it continues sending voice packets. Use this option to disable busyout to the PBX.</li> </ul>
<b>Step 8</b>	Router(config-voice-class)# <b>signal timing oos restart</b> <i>seconds</i>	<p>(Optional) Configures permanent voice connections to be restarted after the trunk has been OOS for a specified time. The default is no signal timing OOS pattern parameters are configured.</p> <p><b>Note</b> This command has no effect if <b>signal timing oos timeout</b> is set to disabled.</p>



	Command or Action	Purpose
<b>Step 9</b>	Router(config-voice-class)# <b>signal timing oos slave-standby</b> <i>seconds</i>	(Optional) Configures a slave port to return to its initial standby state after the trunk has been OOS for a specified time. The default is no signal timing OOS pattern parameters are configured.  <b>Note</b> This command has no effect if <b>signal timing oos timeout</b> is set to disabled.
<b>Step 10</b>	Router(config-voice-class)# <b>signal timing oos {suppress-all</b> <b>  suppress-voice}</b> <i>seconds</i>	(Optional) Configures the router or concentrator to stop sending voice packets or voice and signaling packets to the network if it detects a transmit OOS signaling pattern from the PBX for a specified time. The default is no signal timing OOS pattern parameters are configured.  <b>Note</b> An OOS transmit signaling pattern must be configured with the <b>signal pattern oos transmit</b> command (see Step 4).
<b>Step 11</b>	Router(config-voice-class)# <b>signal timing idle suppress-voice</b> <i>seconds</i>	(Optional) Configures the router or concentrator to stop sending voice packets after the trunk has been idle for a specified time. The default is no signal timing OOS pattern parameters are configured.
<b>Step 12</b>	Router(config-dial-peer)# <b>voice-class permanent</b> <i>tag</i>	Assigns the voice class to the dial peer.  • <i>tag</i> —Specifies the unique number. The valid range is from 1 to 10000.  <b>Note</b> The <b>voice-class</b> command in dial-peer configuration mode is entered with a hyphen. The <b>voice class</b> command in global configuration mode is entered without the hyphen.
<b>Step 13</b>	Router(config-voice-class)# <b>connection plar</b> <i>string</i>	Specifies a PLAR connection and associates a peer directly with an interface.  • <i>string</i> is a destination telephone number. Valid entries are any series of digits that specify the E.164 standard.

## Additional References for Land Mobile Radio

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Commands List, All Releases</a>

## Technical Assistance

Description	Link
The Cisco Support and Documentation website <a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a> provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## Feature Information for Land Mobile Radio

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [http://www.cisco.com/go/featurenavigator](#). An account on Cisco.com is not required.

**Table 2: Feature Information for Connection Trunk**

Feature Name	Releases	Feature Information
Land Mobile Radio	Cisco IOS XE Release 3.17S	A Land Mobile Radio (LMR) system is a collection of portable and stationary radio units designed to communicate with each other over predefined frequencies.

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2015 Cisco Systems, Inc. All rights reserved.



**Americas Headquarters**  
Cisco Systems, Inc.  
San Jose, CA 95134-1706  
USA

**Asia Pacific Headquarters**  
Cisco Systems (USA) Pte. Ltd.  
Singapore

**Europe Headquarters**  
Cisco Systems International BV  
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).