Networking

Topics:

Poly phones support several wireless modes, security options, radio controls, and Quality of Service monitoring.

All phones connect through Ethernet, although some can connect via Wi-Fi as well.

System Names Transmitted with Network Protocols

The phone transmits its system name with network protocols. To customize your network for specific phone models, parse the network packets for these strings.

The phone's system name is the model name with no spaces, followed by an underscore and the last 4 digits of the phone's MAC address.

> For example: CCX700_D1EB

System and Model Names

Model	System Name
CCX 400	CCX400_<*MAC*>
CCX 500	CCX500_<*MAC*>
CCX 600	CCX600_<*MAC*>
CCX 700	CCX700_<*MAC*>

Configuring Internet Protocol Settings

The phone depends on a reliable network connection to perform all of its core functions.

Poly phones place and receive audio/video calls using a network connection. Other features rely on a network connection as well, such as the phone’s ability to sync with a user’s calendar to join meetings.

Configure a Static IPv4 Address

Configure IPv4 mode in the phone’s local interface.

Connect your phone to an Ethernet network connection.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Go to **Settings** > **Advanced** > **Administration Settings** > **Network Configuration** > **Network Interfaces** > **Ethernet Menu**.
2. Select **IP Mode** > **IPv4** .
3. Select **IPv4 Configuration**.
4. Clear the **DHCP** check box.
5. Configure the following settings:
   - **IP Address**
   - **Subnet Mask**
   - **IPv4 Gateway**

6. Back out of the menus. When prompted, select **Save Config**.

   The phone reboots.

Enable IPv4 ICMP Redirects

To ensure your phones communicate using the optimal network route, configure IPv4 to allow Internet Control Message Protocol (ICMP) redirects.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Configure the phone to allow you to enable the ICMP redirect parameter.

   ```
   device.icmp.ipv4IcmpIgnoreRedirect.set="1"
   ```

2. Enable ICMP redirects.

   ```
   device.icmp.ipv4IcmpIgnoreRedirect="0"
   ```

DHCP IP Address

The phone enables DHCP by default.

If the phone can't communicate with the DHCP server on startup, the phone's status bar reports `Network Down`. The phone communicates with the DHCP server every 5 minutes to acquire an IP address or for lease renewal.

Set the DHCP Boot Server Option in IPv4 Mode

Configure the phone based on the DHCP boot server option in IPv4 mode.

Important: Configuring the following parameter(s) causes the phone to reboot. For more information, see the parameter reference topic(s) in the *Poly CCX Parameter Reference Guide*.

Procedure

- **»** Set the phone to get the boot server details from the custom options number provided through DHCP.

  The following values apply:

  - 0 (Default) - The phone gets the boot server address from option 66.
  - 1 - The phone gets the boot server details from the custom option number provided through DHCP.
  - 2 - The phone uses the boot server configured through the **Server** menu.
  - 3 - The phone uses the custom option first or uses option 66 if the custom option isn't present.

  ```
  device.dhcp.bootSrvUseOpt="<value>"
  ```

Enable DHCP IP Address Cache

Enable DHCP IP address cache to retain IP addresses on the phones when the DHCP server becomes unavailable.

When you enable the IP address cache feature, there isn't a service interruption even if the IP address lease time expires and the DHCP server doesn't respond. The phone periodically attempts to resume DHCP service with a new DHCP Discover message for the entire time the cached IP address is in use.

DHCP IP address cache stores the following lease parameters:

- Interface
- IP address
- Subnet mask
- Gateway
- DNS server
- Domain name

DHCP IP address cache has the following limitations:

Important: If a DHCP server restarts and loses lease details, enabling DHCP IP address cache can lead to IP address conflicts on the phones. This results in cascading service outages.

- The phones don't cache DHCP option 99 values for Enhanced 911 location services. A WAN outage may affect IP address cache and emergency calling services.
- If a DHCP server restarts and loses lease details, enabling DHCP IP address cache can lead to IP address conflicts on the phones. This results in cascading service outages.
- DHCP IP address cache supports only IPv4 addresses. DHCP IP address cache doesn't currently support IPv6 addresses.
- DHCP IP address cache doesn't support DHCP VLAN Discovery (DVD).
- If you move a phone from one VLAN to another VLAN where DHCP doesn't respond, the phone continues to use the cached IP address.
- The phones can't update the software using DHCP IP address cache. When the phones attempt to update Poly UC software without DHCP server availability, the phones experience a reboot loop. This continuous reboot loop occurs only when:
  - A cached IP address is in use.
  - The DHCP server is unavailable.
  - A software provisioning server is available.
  - New software is available on the provisioning server.
- You can use DHCP IP address cache only for the UC Software application; you can't use it for the Updater.

Procedure

1. Enable the phone to use a cached IP address if the phone doesn't receive a new IP address from the DHCP user.

   ```
   device.net.cachedIPAddress="1"
   ```

2. If the phone uses a cached IP address, configure how long the phone waits, in seconds, to attempt to get a new IP address from the DHCP server. This parameter is only available when you enable `device.net.cachedIPAddress`.

The default is 3600. The value range is 300 to 7200.

```
device.net.cachedIPAddressRetryTime="<value>"
```

Wi-Fi Network Connectivity

Enabling Wi-Fi automatically disables the Ethernet port. You can't use Wi-Fi and Ethernet simultaneously to connect phones to your network.

Note: CCX 400 and CCX 500 business media phones don't support Wi-Fi.

Note the following when using Wi-Fi:

- The phone still requires power using a power adapter for power when using Wi-Fi.
- When you connect the system to your network over Wi-Fi, you can only place audio-only calls.
- The phone doesn't support Wi-Fi captive portals or Wireless Display (WiDi).

Your phone supports the following wireless modes:

- 2.4 GHz / 5 GHz operation
- IEEE 802.11a radio transmission standard
- IEEE 802.11b radio transmission standard
- IEEE 802.11g radio transmission standard
- IEEE 802.11n radio transmission standard

Note: When you provision via a Wi-Fi connection to the network, the phone looks for files on the provisioning server using the LAN MAC address and not the Wi-Fi MAC address.

Configure Wi-Fi Using a Configuration File

Configure your phone's Wi-Fi settings using a provisioning file.

Connect the phone to your Ethernet network to receive the provisioning file.

Set `device.set="1"`.

Note: CCX 400 and CCX 500 business media phones don't support Wi-Fi.

Procedure

1. Enable Wi-Fi.

   ```
   device.wifi.enabled="1"
   ```

2. Optional: Set a country of operation.

Note: Poly recommends this step to ensure the best performance on a Wi-Fi network. If you don’t set the country of origin, the phone operates in a world safe mode, which restricts Wi-Fi channels and power.

```
device.wifi.country="<two-letter country code>"
```

3. Enable DHCP for Wi-Fi.

```
device.wifi.dhcpEnabled="1"
```

4. Enter the SSID for your Wi-Fi network. The SSID is the network’s name as it appears in a network search.

```
device.wifi.ssid="<SSID>"
```

5. Optional: Specify your Wi-Fi network security mode.

```
device.wifi.securityMode="<wireless security mode type>"
```

   - If your network uses WEP, configure the WEP key.

```
device.wifi.wep.key="<WEP key>"
```

   - If your network uses WPA PSK, WPA2 PSK, or WPA2 PSK Enterprise, configure the security credentials.

```
device.wifi.wpa2Ent.method="<EAP setting>"
device.wifi.wpa2Ent.user="<WPA2 username>"
device.wifi.wpa2Ent.password="<WPA2 password>"
```

Configure Wi-Fi Using the Local Interface

Using the menus available on the phone’s local interface, connect the phone to a Wi-Fi network. This is useful if you don’t have an Ethernet connection available so the phone can send a provisioning file to the server.

Note: CCX 400 and CCX 500 business media phones don’t support Wi-Fi.

Procedure

1. Go to **Settings** > **Advanced** > **Administration Settings** > **Network Configuration** > **Network Interfaces** > **Wi-Fi Menu**
2. Optional: Set a country of operation.

Note: Poly recommends this step to ensure the best performance on a Wi-Fi network. If you don’t set the country of origin, the phone operates in a world safe mode, which restricts Wi-Fi channels and power.

   1. Select **Country of operation**.
   2. Choose your country from the list.
   3. Select the back arrow.

3. Select **Wi-Fi**.
4. Toggle Wi-Fi on and select the back arrow.

   The phone reboots.
5. Go to **Settings** > **Advanced** > **Administration Settings** > **Network Configuration** > **Network Interfaces** > **Wi-Fi Menu** > **Wi-Fi**.
6. Select an available Wi-Fi network.
7. Optional: If required, enter the Wi-Fi network's security password.
8. Select **Connect**.

   The phone connects to the network.

Remove Wi-Fi from the Basic Settings Menu

The default configuration includes a **Wi-Fi** menu item in the **Basic** settings menu.

For increased network security, you can remove the wireless network option from the **Basic** menu. You can restrict phone users from updating wireless network settings from the phone's local interface.

Procedure

» Remove the wireless menu option from the **Basic** menu.

```
homeScreen.wifi.enable="0"
```

Bluetooth Settings

The base configuration disables Bluetooth by default. You can disable Bluetooth entirely, disable certain features, and configure Bluetooth settings.

Limitations with Bluetooth technology may cause voice quality issues when using a Bluetooth headset. You may not experience the highest voice quality using a Bluetooth headset with the 2.4 GHz band enabled. Other Bluetooth devices in the area may also cause interference and quality loss.

Enable Bluetooth

By default, the phone disables Bluetooth and Bluetooth discovery. Enable Bluetooth on the phone and display it on the local interface.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Enable Bluetooth.

   ```
   feature.bluetooth.enabled="true"
   ```
2. Enable Bluetooth radio.

   ```
   bluetooth.radioOn="1"
   ```

3. For security, you can completely disable Bluetooth or turn it off by default. To disable Bluetooth discovery, set:

```
bluetooth.device.discoverable="0"
```

Update the Bluetooth Device Name

By default, the system uses the model number as the Bluetooth device name. Update the device name to something that better identifies the device.

Procedure

» Update the Bluetooth device name. The maximum length is 20 characters.

```
bluetooth.device.name="<Device name>"
```

Configure Bluetooth Features

Adjust the default Bluetooth values based on your deployment requirements.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Set the max time that the phone attempts to connect with other devices.

   The default value 0 disables the discoverable timeout. The value ranges from 0 to 3600 seconds.

```
bluetooth.discoverableTimeout="x"
```

2. Set the maximum number of devices stored in the phone's memory.

   By default, 10 devices remain in the phone's memory. The value ranges from 0 to 3600 seconds.

```
bluetooth.pairedDeviceMemorySize="x"
```

3. Set the maximum number of devices the phone can pair with. If you don't want the phone to store devices is memory, set this value to 0.

   By default, 10 devices remain in the phone's memory.

```
bluetooth.device.maxPaired="x"
```

4. Set the amount of time, in minutes, that the phone remains paired with a device when you set `bluetooth.device.maxPaired` to 0.

   By default, the phone remains paired for 30 minutes.

```
bluetooth.device.pairedTimeout="x"
```

Setting the Time and Date

Synchronizing the phone to the SNTP server gives you the most accurate time and date. The phone continuously flashes the time and date until it receives a successful SNTP response.

Related Links

Time and Date Display on page 186

Configure Time and Daylight Saving Time

Configure time, time zone, and daylight saving time on the phone.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Configure the SNTP server to automatically configure the time.

```
tcpIpApp.sntp.address="<valid SNTP hostname or IP address>"
tcpIpApp.sntp.resyncPeriod="<# of seconds>"
```

2. Optional: For time zones offset from GMT by fractions of a whole hour, specify the offset (in seconds) up to one hour (+/- 3600 seconds). A value 0 (default) represents GMT.

```
tcpIpApp.sntp.gmtOffset="<positive or negative integer>"
```

3. Optional: Configure Daylight Saving Time.

```
tcpIpApp.sntp.daylightSavings.fixedDayEnable="1"
tcpIpApp.sntp.daylightSavings.start.month="<set month to start DST>"
tcpIpApp.sntp.daylightSavings.start.date="<date of set month to start
DST>"
tcpIpApp.sntp.daylightSavings.start.time="<hour of set date to start
DST>"
tcpIpApp.sntp.daylightSavings.stop.month="<set month to stop DST>"
tcpIpApp.sntp.daylightSavings.stop.date="<date of set stop to start
DST>"
tcpIpApp.sntp.daylightSavings.stop.month="<set month to stop DST>"
```

Set the Time Zone Location

If you configure your time zone with `device.sntp.gmtOffset` or `tcpIpApp.sntp.gmtOffset`, you must set the correct time zone location to display on the phone and in the system web interface.

Your configuration file must include `device.set="1"`.

Procedure

- Set the correct time zone location to display on the local interface and the system web interface.

```
device.sntp.gmtOffsetcityID="<time zone location parameter value>"
```

Use the following parameters to configure the time zone location.

Time Zone Location Parameter Values

Permitted Value	Time Zone Description
0	(GMT -12:00) Eniwetok, Kwajalein
1	(GMT -11:00) Midway Island
2	(GMT -10:00) Hawaii
3	(GMT -9:00) Alaska
4	(GMT -8:00) Pacific Time (US & Canada)
5	(GMT -8:00) Baja California
6	(GMT -7:00) Mountain Time (US & Canada)
7	(GMT -7:00) Chihuahua, La Paz
8	(GMT -7:00) Mazatlan
9	(GMT -7:00) Arizona
10	(GMT -6:00) Central Time (US & Canada)
11	(GMT -6:00) Mexico City
12	(GMT -6:00) Saskatchewan
13	(GMT -6:00) Guadalajara
14	(GMT -6:00) Monterrey
15	(GMT -6:00) Central America
16	(GMT -5:00) Eastern Time (US & Canada)
17	(GMT -5:00) Indiana (East)
18	(GMT -5:00) Bogota, Lima
19	(GMT -5:00) Quito
20	(GMT -4:30) Caracas
21	(GMT -4:00) Atlantic Time (Canada)
22	(GMT -4:00) San Juan
23	(GMT -4:00) Manaus, La Paz
24	(GMT -4:00) Asuncion, Cuiaba
25	(GMT -4:00) Georgetown
26	(GMT -3:30) Newfoundland
27	(GMT -3:00) Brasilia
28	(GMT -3:00) Buenos Aires
29	(GMT -3:00) Greenland
30	(GMT -3:00) Cayenne, Fortaleza

Permitted Value	Time Zone Description
31	(GMT -3:00) Montevideo
32	(GMT -3:00) Salvador
33	(GMT -3:00) Santiago
34	(GMT -2:00) Mid-Atlantic
35	(GMT -1:00) Azores
36	(GMT -1:00) Cape Verde Islands
37	(GMT 0:00) Western Europe Time
38	(GMT 0:00) London, Lisbon
39	(GMT 0:00) Casablanca
40	(GMT 0:00) Dublin
41	(GMT 0:00) Edinburgh
42	(GMT 0:00) Monrovia
43	(GMT 0:00) Reykjavik
44	(GMT +1:00) Belgrade
45	(GMT +1:00) Bratislava
46	(GMT +1:00) Budapest
47	(GMT +1:00) Ljubljana
48	(GMT +1:00) Prague
49	(GMT +1:00) Sarajevo, Skopje
50	(GMT +1:00) Warsaw, Zagreb
51	GMT +1:00) Brussels
52	(GMT +1:00) Copenhagen
53	(GMT +1:00) Madrid, Paris
54	(GMT +1:00) Amsterdam, Berlin
55	(GMT +1:00) Bern, Rome
56	(GMT +1:00) Stockholm, Vienna
57	(GMT +1:00) West Central Africa
58	(GMT +1:00) Windhoek
59	(GMT +2:00) Bucharest, Cairo
60	(GMT +2:00) Amman, Beirut

Permitted Value	Time Zone Description
61	(GMT +2:00) Helsinki, Kyiv
62	(GMT +2:00) Riga, Sofia
63	(GMT +2:00) Tallinn, Vilnius
64	(GMT +2:00) Athens, Istanbul
65	(GMT +2:00) Damascus
66	(GMT +2:00) E.Europe
67	(GMT +2:00) Harare, Pretoria
68	(GMT +2:00) Jerusalem
69	(GMT +2:00) Kaliningrad (RTZ 1)
70	(GMT +2:00) Tripoli
71	(GMT +3:00) Moscow
72	(GMT +3:00) St.Petersburg
73	(GMT +3:00) Volgograd (RTZ 2)
74	(GMT +3:00) Kuwait, Riyadh
75	(GMT +3:00) Nairobi
78	(GMT +3:00) Baghdad
76	(GMT +3:00) Minsk
77	(GMT +3:30) Tehran
79	(GMT +4:00) Abu Dhabi, Muscat
80	(GMT +4:00) Baku, Tbilisi
81	(GMT +4:00) Izhevsk, Samara (RTZ 3)
82	(GMT +4:00) Port Louis
83	(GMT +4:00) Yerevan
84	(GMT +4:30) Kabul
85	(GMT +5:00) Yekaterinburg (RTZ 4)
86	(GMT +5:00) Islamabad
87	(GMT +5:00) Karachi
88	(GMT +5:00) Tashkent
89	(GMT +5:30) Mumbai, Chennai
90	(GMT +5:30) Kolkata, New Delhi

Permitted Value	Time Zone Description
91	(GMT +5:30) Sri Jayawardenepura
92	(GMT +5:45) Kathmandu
93	(GMT +6:00) Astana, Dhaka
94	(GMT +6:00) Almaty
95	(GMT +6:00) Novosibirsk (RTZ 5)
96	(GMT +6:30) Yangon (Rangoon)
97	(GMT +7:00) Bangkok, Hanoi
98	(GMT +7:00) Jakarta
99	(GMT +7:00) Krasnoyarsk (RTZ 6)
100	(GMT +8:00) Beijing, Chongqing
101	(GMT +8:00) Hong Kong, Urumqi
102	(GMT +8:00) Kuala Lumpur
103	(GMT +8:00) Singapore
104	(GMT +8:00) Taipei, Perth
105	(GMT +8:00) Irkutsk (RTZ 7)
106	(GMT +8:00) Ulaanbaatar
107	(GMT +9:00) Tokyo, Seoul, Osaka
108	(GMT +9:00) Sapporo, Yakutsk (RTZ 8)
109	(GMT +9:30) Adelaide, Darwin
110	(GMT +10:00) Canberra
111	(GMT +10:00) Magadan (RTZ 9)
112	(GMT +10:00) Melbourne
113	(GMT +10:00) Sydney, Brisbane
114	(GMT +10:00) Hobart
115	(GMT +10:00) Vladivostok
116	(GMT +10:00) Guam, Port Moresby
117	(GMT +11:00) Solomon Islands
118	(GMT +11:00) New Caledonia
119	(GMT +11:00) Chokurdakh (RTZ 10)
120	(GMT +12:00) Fiji Islands

Permitted Value	Time Zone Description
121	(GMT +12:00) Auckland, Anadyr
122	(GMT +12:00) Petropavlovsk-Kamchatsky (RTZ 11)
123	(GMT +12:00) Wellington
124	(GMT +12:00) Marshall Islands
125	(GMT +13:00) Nuku'alofa
126	(GMT +13:00) Samoa

Configure Olson Time Zone

Configure an Olson time zone on your phone to ensure a more accurate time and date display.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Use the values in this table to configure the Olson time zone.

Olson Time Zone IDs

Olson Time Zone ID	Poly Time Zone ID
Pacific/Midway	(GMT -11:00) Midway Island
Pacific/Honolulu	(GMT -10:00) Hawaii
America/Anchorage	(GMT -9:00) Alaska
Mexico/BajaNorte	(GMT -8:00) Baja California
America/Phoenix	(GMT -7:00) Arizona
America/Chihuahua	(GMT -7:00) Chihuahua,La Paz
America/Denver	(GMT -7:00) Mountain Time (US & Canada)
America/Costa_Rica	(GMT -6:00) Central America
America/Chicago	(GMT -6:00) Central Time (US & Canada)
America/Mexico_City	(GMT -6:00) Mexico City
America/Regina	(GMT -6:00) Saskatchewan
America/Bogota	(GMT -5:00) Bogota,Lima
America/New_York	(GMT -5:00) Eastern Time (US & Canada)
America/Caracas	(GMT -4:30) Caracas
America/Barbados	Atlantic Time (Barbados)
America/Halifax	(GMT -4:00) Atlantic Time (Canada)

Olson Time Zone ID	Poly Time Zone ID
America/Manaus	(GMT -4:00) Manaus,La Paz
America/Santiago	(GMT -3:00) Santiago
America/St_Johns	(GMT -3:30) Newfoundland
America/Sao_Paulo	(GMT -3:00) Brasilia
America/Argentina/Buenos_Aires	(GMT -3:00) Buenos Aires
America/Godthab	(GMT -3:00) Greenland
America/Montevideo	(GMT -3:00) Montevideo
Atlantic/South_Georgia	(GMT -2:00) Mid-Atlantic
Atlantic/Azores	(GMT -1:00) Azores
Atlantic/Cape_Verde	(GMT -1:00) Cape Verde Islands
Africa/Casablanca	(GMT 0:00) Casablanca
Europe/London	(GMT 0:00) London,Lisbon
Europe/Amsterdam	(GMT +1:00) Amsterdam,Berlin
Europe/Belgrade	(GMT +1:00) Bratislava
Europe/Brussels	(GMT +1:00) Brussels
Europe/Sarajevo	(GMT +1:00) Sarajevo,Skopje
Africa/Brazzaville	(GMT +1:00) West Central Africa
Africa/Windhoek	(GMT +1:00) Windhoek
Asia/Amman	Amman
Europe/Athens	(GMT +2:00) Athens
Asia/Beirut	Beirut
Africa/Cairo	(GMT +2:00) Bucharest,Cairo
Europe/Helsinki	(GMT +2:00) Helsinki,Kyiv
Asia/Jerusalem	(GMT +2:00) Jerusalem
Africa/Harare	(GMT +2:00) Harare,Pretoria
Europe/Minsk	(GMT +3:00) Minsk
Asia/Istanbul	(GMT +3:00) Istanbul
Europe/Moscow	(GMT +3:00) Moscow

Olson Time Zone ID	Poly Time Zone ID
Asia/Kuwait	(GMT +3:00) Kuwait,Riyadh
Africa/Nairobi	(GMT +3:00) Nairobi
Asia/Tehran	(GMT +3:30) Tehran
Asia/Baku	(GMT +4:00) Baku,Tbilisi
Asia/Yerevan	(GMT +4:00) Yerevan
Asia/Dubai	Dubai
Asia/Kabul	(GMT +4:30) Kabul
Asia/Karachi	(GMT +5:00) Karachi
Asia/Tashkent	(GMT +5:00) Tashkent
Asia/Yekaterinburg	(GMT +5:00) Yekaterinburg (RTZ 4)
Asia/Calcutta	(GMT +5:30) Kolkata,New Delhi
Asia/Colombo	(GMT +5:30) Sri Jayawardenepura
Asia/Katmandu	(GMT +5:45) Kathmandu
Asia/Dhaka	(GMT +6:00) Astana,Dhaka
Asia/Rangoon	(GMT +6:30) Yangon (Rangoon)
Asia/Krasnoyarsk	(GMT +7:00) Krasnoyarsk (RTZ 6)
Asia/Bangkok	(GMT +7:00) Bangkok,Hanoi
Asia/Jakarta	(GMT +7:00) Jakarta
Asia/Shanghai	(GMT +8:00) Beijing,Chongqing
Asia/Hong_Kong	(GMT +8:00) Hong Kong,Urumqi
Asia/Irkutsk	(GMT +8:00) Irkutsk (RTZ 7)
Asia/Kuala_Lumpur	(GMT +8:00) Kuala Lumpur
Asia/Taipei	(GMT +8:00) Taipei,Perth
Asia/Tokyo	(GMT +9:00) Tokyo,Seoul,Osaka
Asia/Yakutsk	(GMT +9:00) Sapporo,Yakutsk (RTZ 8)
Australia/Adelaide	Adelaide
Australia/Darwin	Darwin
Australia/Brisbane	Brisbane

Olson Time Zone ID	Poly Time Zone ID
Australia/Hobart	(GMT +10:00) Hobart
Australia/Sydney	Sydney,Canberra
Asia/Vladivostok	(GMT +10:00) Vladivostok
Pacific/Guam	(GMT +10:00) Guam,Port Moresby
Asia/Magadan	(GMT +10:00) Magadan (RTZ 9)
Pacific/Auckland	(GMT +12:00) Auckland,Anadyr
Pacific/Fiji	(GMT +12:00) Fiji Islands
Pacific/Majuro	(GMT +12:00) Marshall Islands
Pacific/Tongatapu	(GMT +13:00) Nuku'alofa

Procedure

» Enter an Olson time zone ID. If you set it to an invalid or unrecognized value, the time zone resets to GMT with daylight saving time disabled.

```
tcpIpApp.sntp.olsonTimezoneID="<Olson time zone ID>"
```

IP Multimedia Subsystem Features

Poly CCX business media phones support several IP multimedia subsystem features.

- The call waiting ring-back tone plays to inform users that a call is waiting at the far end.
- The phone supports SIP response code 199 (defined in RFC 6228).
- The **Path** extension header field in the SIP Register request message enables accumulating and transmitting the list of proxies between a user agent and registrar server.
- The caller phone can support the p-early-media SIP header that determines whether the caller phone plays a network-provided media or its own media as a ringback tone.
- The VQMon messages generated by the phone can contain service route information in SIP route headers.
- In a NAT network, a phone may need to send keep-alive messages to maintain the IP addresses mapping in the NAT table.

Enable 3GPP IP Multimedia

Enable the phone to support any IP multimedia (IPM) features.

For an IP multimedia subsystem (IMS) environment, Poly supports a subset of the following 3rd Generation Partnership Project technical specifications (3GPP TS): 24.229, 24.615, and 24.629.

In addition, Poly phones provide partial or complete support for the following RFCs:

- RFC 3327
- RFC 3608

- RFC 3680
- RFC 6665
- RFC 6228
- RFC 3261
- RFC 5009
- RFC 7462
- RFC 7329
- RFC 6026
- RFC 3581
- RFC 6947

Procedure

» Enable support for 3GPP IPM features. This parameter applies to all registered and unregistered SIP lines on the phone.

```
voIpProt.SIP.IMS.enable="1"
```

Create a Custom TCP Keep-Alive Message

Configure a string as the payload for TCP keep-alive messages.

Procedure

» Create a custom string to use as the payload of a TCP keep-alive message. You can't leave the string value blank.

The default string is CRLFCRLFCRLFCRLFCRLFCRLFCRLFCRLF.

```
nat.keepalive.tcp.payload="<string>"
```

Create a Custom UDP Keep-Alive Message

Create a string as the payload of a UDP keep-alive message.

Procedure

» Create a custom string to use as the payload of a UDP keep-alive message. You can leave the string value blank to configure an empty payload.

The default string is CRLFCRLF.

```
parameternat.keepalive.udp.payload="<string>"
```

Enable the P-Early-Media Header

Enable support for the p-early-media header for all lines or for specific registered lines.

Enabling this parameter enables the phone to play network-provided media or its own media as a ringback tone.

Procedure

» Do one of the following:

- Enable the phone to support p-early-media on all outgoing calls.

```
voIpProt.SIP.header.pEarlyMedia.support="1"
```

- Enable the p-early-media header on a registered line. Replace *x* with the registered line number.

```
reg.x.header.pearlymedia.support="1"
```

Remove the Outbound Proxy Address from the Route Header

Prevent the phone from including the outbound proxy address as the topmost route header on a registered line.

Procedure

» Remove the outbound proxy address in the route header. Replace *x* with the registered line number.

```
reg.x.insertOBPAddressInRoute="0"
```

Add Path Extension Header to Request Message

Provide the path extension header field in the Register request message for a specific line registration.

Procedure

» Support and include the path extension header field in the Register request message for a registered line. Replace *x* with the registered line number.

```
reg.x.path="1"
```

Subscribe to Registered Line State Change Notifications

Enable the phone to accept state change notifications for all lines or for specific registered lines.

Procedure

» Do one of the following:

- Subscribe the phone to state change notifications for all lines.

Note: The `reg.x.regevent` parameter overrides this setting for the registered line it's configured for.

```
voIpProt.SIP.regevent="1"
```

- Subscribe the phone to state change notifications for a registered line. Replace *x* with the registered line number.

Note: Setting this parameter overrides the setting in the `voIpProt.SIP.regevent` global parameter for the registered line.

```
reg.x.regevent="1"
```

Reject Calls with Network Determined User Busy Events

The phone can reject incoming calls if it detects a Network Determined User Busy (NDUB) event on all lines or on specific registered lines.

If an NDUB event occurs on any registered lines, the phone rejects the call with a `603 Decline` response code.

Procedure

- **»** Do one of the following:
  - Reject calls when the phone detects an NDUB event on all lines.

    ```
    voIpProt.SIP.rejectNDUBInvite="1"
    ```

  - Reject calls when the phone detects an NDUB event on a registered line. Replace *x* with the registered line number.

    ```
    reg.x.rejectNDUBInvite="1"
    ```

Enable Server-Specific Features

Configure the phone to work with server-specific features on registered lines.

The phone supports the following features:

- Standard (default)
- GENBAND
- ALU-CTS
- ocs2007r2
- lcs2005

Procedure

- **»** Enable server-specific features on registered on a registered line. Replace *x* with the desired line key value. Replace *y* with the desired server key value.

  ```
  reg.x.server.y.specialInterop="<feature>"
  ```

Include Service Route Information in VQMon Messages

Include service route information in the voice quality monitoring (VQMon) messages it creates.

Important: Configuring the following parameter(s) causes the phone to reboot. For more information, see the parameter reference topic(s) in the *Poly CCX Parameter Reference Guide*.

Procedure

» Enable the phone to include service route information in VQMon messages.

```
voice.qualityMonitoring.processServiceRoute.enable="1"
```

Enable Support for the 199 Response Code

Enable the phone to support the 199 response code. For information on the 199 response code, see RFC 6228.

Procedure

» Enable support for the 199 response code.

```
voIpProt.SIP.supportFor199="1"
```

Enable Advice of Charge

In an IP multimedia subsystem (IMS) environment, Poly phones support the Advice of Charge (AoC) feature as defined in Technical Specification (TS) 24.647 version 9.1.0 Release 9.

Set:`voIpProt.SIP.IMS.enable="1"`.

Enable Poly phones to display call charges information, which include the following:

- Call setup charge and call tariff information - Displayed at the beginning of a call.
- Cumulative call cost - Displayed on an ongoing call.
- Complete call cost - Displayed after a call ends.

Procedure

1. Display call charge information on the phone.

   ```
   voIpProt.SIP.aoc.enable="1"
   ```

2. Optional: Enable the phone to sound a beep when call charges update on the display.

   ```
   feature.adviceOfCharge.allowAudioNotification="1"
   ```

Enable and Configure TWAMP

UC Software supports Two-Way Active Measurement Protocol (TWAMP), based on RFC 5357 Enable and configure TWAMP to review packet loss and latency between endpoints.

TWAMP defines a control protocol that uses TCP and a test protocol that uses UDP. TWAMP includes the following limitations:

- TWAMP control and test protocols only support unauthenticated mode.
- A maximum of 10 clients can establish a connection with the server.
- The server handles a maximum of 10 sessions per client.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Enable TWAMP.

   ```
   feature.twamp.enabled="1"
   ```

2. Set the TWAMP max port range end. The default is 60000. The value range is 1024 to 65486.

   ```
   twamp.port.udp.PortRangeEnd="<Max port range end>"
   ```

3. Set the TWAMP port range start. The default is 4000. The value range is 1024 to 65485.

   ```
   twamp.port.udp.PortRangeStart="<Port range start>"
   ```

4. Set the maximum TWAMP sessions that can run simultaneously. The default is 1. The value range is 1 to 10.

   ```
   twamp.udp.maxSession="<Max number of simultaneous sessions>"
   ```

Technical Report-069

Technical Report-069 (TR-069) enables you to remotely manage end-user devices.

As a bidirectional SOAP/HTTP-based protocol, TR-069 enables secure communication between auto configuration servers (ACS) and Poly phones. Using TR-069, you can remotely configure and manage Poly phones by provisioning systems that comply with TR-069 technical specification. Configure the TR-069 feature through the system web interface or using configuration parameters on a central provisioning server.

You can configure Poly phones with an ACS server, including username and password, using DHCP option 43 for IPv4. Poly CCX business media phones don’t support IPv6.

Configure TR-069 in the System Web Interface

Configure TR-069 from the system web interface.

Procedure

1. In the system web interface, go to **Settings** > **Provisioning Server** > **TR-069 Menu**.
2. Select **Enable**.
3. Enter the values as needed in the provided fields.
   - ACS URL
   - ACS Username
   - ACS Password
   - CPE Username
   - CPE Password
   - Periodic Inform

- Inform Interval
- Managed Upgrades

4. Select **Save**.

Enable and Configure TR-069 Using a Configuration File

Configure TR-069 using configuration parameters.

Poly provides parameters for the TR-104 and TR-106 data models that support provisioning of TR-069-enabled devices by an auto configuration server (ACS). TR-104 is a parameter data model for VoIP-only devices, and TR-106 is a parameter data model for all TR-069-enabled devices.

Procedure

1. Enable TR-069.

```
device.feature.tr069.enabled="1"
```

2. Enable `device.set` for the TR-069 feature.

```
device.feature.tr069.enabled.set="1"
```

3. Enter the TR-069 ACS server URL.

```
device.tr069.acs.url="<valid URL>"
```

4. Enter the TR-069 username and password to authenticate the phone.

```
device.tr069.acs.username="<username>"
device.tr069.acs.password="<password>"
```

5. Enter the username and password to authenticate a connection request from the ACS server.

```
device.tr069.cpe.username="<username>"
device.tr069.cpe.password="<password>"
```

TR-106 Parameters Mapped to Poly Parameters

The data model TR-106 defines the TR-069 ACS parameter details.

The following tables list the TR-106 parameters and their corresponding Poly parameters.

Note: The parameters listed as Internal Value don’t map directly to a configuration parameter on the phone, and the phone generates these values dynamically to provide to the ACS server.

Device and Device.DeviceInfo

TR-106 ACS parameter names	Poly Parameter	Writable
Manufacturer	Internal Value	No
ManufacturerOUI	Internal Value	No
ModelName	Internal Value	No

TR-106 ACS parameter names	Poly Parameter	Writable
ProductClass	Internal Value	No
SerialNumber	Internal Value	No
HardwareVersion	Internal Value	No
SoftwareVersion	Internal Value	No
UpTime	Internal Value	No

Device.ManagementServer

TR-106 ACS parameter names	Poly Parameter	Writable
URL	`device.tr069.acs.url`	Yes
Username	`device.tr069.acs.username`	Yes
Password	`device.tr069.acs.password`	Yes
PeriodicInformEnable	`device.tr069.periodicInform.enabled`	Yes
PeriodicInformInterval	`device.tr069.periodicInform.interval`	Yes
ConnectionRequestURL	Internal Value	No
ConnectionRequestUsername	`device.tr069.cpe.username`	Yes
ConnectionRequestPassword	`device.tr069.cpe.password`	Yes
UpgradesManaged	`device.tr069.upgradesManaged.enabled`	Yes
STUNServerAddress	`tcpIpApp.ice.stun.server`	Yes
STUNServerPort	`tcpIpApp.ice.stun.udpPort`	Yes
STUNUsername	`tcpIpApp.ice.username`	Yes
STUNPassword	`tcpIpApp.ice.password`	Yes

Device.LAN

TR-106 ACS parameter names	Poly Parameter	Writable
IPAddress	Internal Value	No
SubnetMask	Internal Value	No
DNSServers	Internal Value	No
MACAddress	Internal Value	No
MACAddressOverride	Internal Value	No

TR-104 Parameters Mapped to Poly Parameters

The data model TR-104 defines the TR-069 ACS parameter details.

The following tables list the TR-104 parameters and their corresponding Poly parameters.

Note: The parameters listed as Internal Value don’t map directly to a configuration parameter on the phone, and the phone generates these values dynamically to provide to the ACS server.

VoiceService.{i}.VoiceProfile.{i}

TR-104 ACS parameter names	Poly Parameters	Writable
DigitMap	`dialplan.digitmap`	Yes

VoiceService.{i}.VoiceProfile.{i}.SIP

TR-104 ACS parameter names	Poly Parameters	Writable
RegistrarServer	`voIpProt.server.X.address`	Yes
RegistrarServerPort	`voIpProt.server.X.port`	Yes
OutboundProxy	`voIpProt.SIP.outboundProxy.address`	Yes
OutboundProxyPort	`voIpProt.SIP.outboundProxy.port`	Yes
RegisterExpires	`voIpProt.server.X.expires`	Yes
RegistersMinExpires	`voIpProt.server.X.expires.overlap`	Yes
RegisterRetryInterval	`voIpProt.server.X.retryTimeOut`	Yes

VoiceService.{i}.VoiceProfile.{i}.SIP.EventSubscribe.{i}

TR-104 ACS parameter names	Poly Parameters	Writable
ExpireTime	`voIpProt.server.X.subscribe.expires`	Yes

VoiceService.{i}.VoiceProfile.{i}.RTP

TR-104 ACS parameter names	Poly Parameters	Writable
LocalPortMin	`tcpIpApp.port.rtp.mediaPortRangeStart`	Yes
LocalPortMax	`tcpIpApp.port.rtp.mediaPortRangeEnd`	Yes

VoiceService.{i}.VoiceProfile.{i}.RTP.SRTP

TR-104 ACS parameter names	Poly Parameters	Writable
Enable	`sec.srtp.enable`	Yes

VoiceService.{i}.VoiceProfile.{i}.ButtonMap.Button.{i}

TR-104 ACS parameter names	Poly Parameters	Writable
ButtonName	`softkey.X.label`	Yes
FacilityAction	`softkey.X.action`	Yes
UserAccess	`softkey.X.enable`	Yes

VoiceService.{i}.VoiceProfile.{i}.Line.{i}

TR-104 ACS parameter names	Poly Parameters	Writable
DirectoryNumber	`reg.X.address`	Yes

VoiceService.{i}.VoiceProfile.{i}.Line.{i}.SIP

TR-104 ACS parameter names	Poly Parameters	Writable
AuthUserName	`reg.X.auth.userId`	Yes
AuthPassword	`reg.X.auth.password`	Yes

VoiceService.{i}.VoiceProfile.{i}.Line.{i}.CallingFeatures

TR-104 ACS parameter names	Poly Parameters	Writable
CallForwardUnconditionalEnable	`reg.X.fwdStatus`	Yes
CallForwardUnconditionalNumber	`reg.X.fwdContact`	Yes
CallForwardOnBusyEnable	`reg.X.fwd.busy.status`	Yes
CallForwardOnBusyNumber	`reg.X.fwd.busy.contact`	Yes
CallForwardOnNoAnswerEnable	`reg.X.fwd.noanswer.status`	Yes
CallForwardOnNoAnswerNumber	`reg.X.fwd.noanswer.contact`	Yes
CallForwardOnNoAnswerRingCount	`reg.X.fwd.noanswer.ringCount`	Yes
DoNotDisturbEnable	`divert.dnd.X.enabled`	Yes

Supported TR-069 Remote Procedure Call (RPC) Methods

The following table lists the supported RPC methods.

RPC Methods

RPC Method	Description
GetRPCMethods	Discovers the set of methods supported by the phone.

RPC Method	Description
SetParameterValues	Modifies the value of one or more phone parameters.
GetParameterValues	Obtains the value of one or more phone parameters.
GetParameterNames	Discovers the parameters accessible on a particular phone.
GetParameterAttributes	Reads the attributes associated with one or more phone parameters.
SetParameterAttributes	Modifies attributes associated with one or more phone parameters.
Reboot	Reboots the phone.
Download	Causes the phone to download a specified file from the designated location. Supported file types for download: ▪ Firmware Image ▪ Configuration File
FactoryReset	Resets the phone to its factory default state.
TransferComplete	Informs the ACS of the completion (either successful or unsuccessful) of a file transfer initiated by an earlier Download or Upload method call.
AddObject	Adds a new instance of an object defined on the phone.
DeleteObject	Removes a particular instance of an object.

Configure Network Signaling Validation

Specify the validation type, method, and the events for validating incoming network signaling

You can choose from the following for validating incoming signaling:

- Source IP address validation - Only accept SIP traffic from trusted IP addresses.
- Digest authentication - Verifies that both parties on a connection (host and endpoint client) know a shared secret (a password). The phone can use this verification method without sending the password in the clear.
- Both source IP address validation and digest authentication

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Set a signaling validation method.
   - Null (default): No validation is made.
   - Source: Ensure request is received from an IP address of a server belonging to the set of target registration servers.

- digest - Challenge requests with digest authentication using the local credentials for the associated registration (line).
- both or all: Apply both of the above methods.

```
voIpProt.SIP.requestValidation.x.method="<value>"
```

2. Set the SIP requests in which validation will be applied.
   - Null (default).
   - INVITE
   - ACK
   - BYE
   - REGISTER
   - CANCEL
   - OPTIONS
   - INFO
   - MESSAGE
   - SUBSCRIBE
   - NOTIFY
   - REFER
   - PRACK
   - UPDATE

   ```
   voIpProt.SIP.requestValidation.x.request="<value>"
   ```

3. Set which events specified with the Event header should be validated
   - Null (default): all events will be validated.
   - A valid string - specified event will be validated

   ```
   voIpProt.SIP.requestValidation.x.request.y.event="<value>"
   ```

   This is applicable only when `voIpProt.SIP.requestValidation.x.request` is SUBSCRIBE or NOTIFY

Jitter Buffer and Packet Error Concealment

Jitter buffer mitigates packet interarrival jitter and out-of-order, lost, or delayed (by the network) packets. You can configure jitter buffer for wired network voice traffic and IP multicast voice traffic.

You can adapt and configure jitter buffer for different network environments. When the audio stream loses packets, a concealment algorithm minimizes negative audio consequences. This feature is enabled by default.

For a list of configurable parameters, see "Voice Jitter Buffer Parameters" in the *Poly CCX Parameter Reference Guide*.

Configure Jitter Buffer for Wired Network Voice Traffic

Configure jitter buffer for wired network voice traffic.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Jitter above the average but below the maximum may result in delayed audio while the jitter buffer adapts. The audio stream doesn't lose packets. Actual jitter above the maximum value always results in packet loss. If you specify legacy `voice.audioProfile.x.jitterBuffer.*` parameters, they configure the jitter buffer and the phone ignores the `voice.rxQoS.*` parameters.

Procedure

1. Enter an average jitter setting in milliseconds. The default setting is 20.The range of values is 0 to 80.

   The average jitter in milliseconds for wired network interface voice traffic.

   ```
   voice.rxQoS.avgJitter="<value>"
   ```

2. Configure the maximum jitter in milliseconds. The default setting is 240.The range of values is 0 to 320.

   The wired interface minimum depth adaptively handles this level of continuous jitter without packet loss.

   ```
   voice.rxQoS.maxJitter="<value>"
   ```

Configure Jitter Buffer for IP Multicast Voice Traffic

Configure jitter buffer for push-to-talk interface voice traffic.

The PTT/paging interface jitter buffer maximum depth is automatically configured to handle this level of intermittent jitter without packet loss.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Actual jitter above the average but below the maximum may result in delayed audio play out while the jitter buffer adapts, but no packets are lost. Actual jitter above the maximum value always results in packet loss.

If legacy `voice.audioProfile.x.jitterBuffer.*` parameters are explicitly specified, they will be used to configure the jitter buffer and these `voice.rxQoS.*`parameters are ignored.

Procedure

1. Enter an average jitter setting in milliseconds. The default setting is 240.

   ```
   voice.rxQoS.ptt.avgJitter="<0 to 320>"
   ```

2. Enter maximum jitter setting in milliseconds. The default setting is 480.

   ```
   voice.rxQoS.ptt.maxJitter="<2 to 500>"
   ```

Set 802.1p/Q Priority

The phone uses IEEE 802.1P and 802.1Q frame tagging protocol for call network traffic. Configure user priority for RTP and call control packets, such as SIP signaling packets, with default settings configurable for all other packets.

The phone tags all Ethernet packets it transmits with an 802.1Q VLAN header when the following occurs:

- The phone's network configuration specifies a valid VLAN ID.
- The phone configuration instructions the phone tag packets through Cisco Discovery Protocol (CDP) running on a connected Ethernet switch.
- DHCP or LLDP obtains a VLAN ID.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Set the user priority for packets without a per-protocol setting. The default is 2. The value range is 0 to 7.

   ```
   qos.ethernet.other.user_priority="<Generic packet priority>"
   ```

2. Set the user priority for video RTP packets. The default is 5. The value range is 0 to 7.

   ```
   qos.ethernet.rtp.video.user_priority="<Video RTP packet priority>"
   ```

3. Set the user priority for voice RTP packets. The default is 5. The value range is 0 to 7.

   ```
   qos.ethernet.rtp.user_priority="<Voice RTP packet priority>"
   ```

4. Set the user priority for call control packets. The default is 5. The value range is 0 to 7.

   ```
   qos.ethernet.callControl.user_priority="<Call control packet
   priority>"
   ```

Provisional Polling of Phones

You can configure phones to poll the server for provisioning updates automatically, and you can set the phone's automatic provisioning behavior to one of the following:

Absolute: The phone polls at the same time every day.

Relative: The phone polls every *x* seconds, where *x* is a number greater than 3600.

Random: The phone polls randomly based on a set time interval.

If the time period is less than or equal to one day, the first poll is at a random time between when the phone starts up and the polling period. Afterward, the phone polls every *x* seconds.

If you set the polling period greater than one day, and rounded up to the nearest day, the phone polls on a random day based on the phone's MAC address. The phones polls within a random time set by the start and end polling time.

Configure Polling for Provisioning Updates

Configure your phones to poll the provisioning server for configuration updates.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Enable automatic polling for updates.

   ```
   prov.polling.enabled="1"
   ```

2. Set the start time for polling the provisioning server.

   The default is 03:00.

   ```
   prov.polling.time="<hh:mm>"
   ```

3. Set the stop time for polling the provisioning server.

   The default is Null.

   ```
   prov.polling.timeRandomEnd="<hh:mm>"
   ```

4. Set the provisioning polling period, in seconds.

   The default is 86400. The integer value must be greater than 3600 seconds.

   Note: The server calculates the polling period in seconds and rounds it up to the nearest number of days in absolute and random mode. If you set this value to a time greater than 86400 (one day), polling occurs on a random day based on the phone's MAC address.

   ```
   prov.polling.period="<polling period>"
   ```

5. Set the provisioning polling mode.

   - abs (default): Absolute; the phone polls every day at the time specified by `prov.polling.time`.
   - rel: Relative; the phone polls after the number of seconds specified by `prov.polling.period`.
   - random: Random; the phone polls at random between a starting time set in `prov.polling.time` and an end time set in `prov.polling.timeRandomEnd`.

   Note: If you set the polling period in `prov.polling.period` to a time greater than 86400 seconds (one day), polling occurs on a random day within that polling period and only between the start and end times. The server calculates the day within the period based upon the phone's MAC address and doesn't change with a reboot. However, the server calculates the time within the start and end times again with every reboot.

   ```
   prov.polling.mode="<polling mode>"
   ```

Configure Provisional Polling for Multiple Phones at Random Times

If you have multiple phones, you can configure polling to happen at different times for each phone.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Configure one or more phones to randomly poll the provisioning server. The phone polls the server between 1:00 AM and 5:00 AM every day. When you configure multiple phones, the phones randomly poll the server based on their IP Address.

Procedure

1. Set polling to random.

   ```
   prov.polling.mode="random"
   ```

2. Set the polling period to 604800 seconds (7 days).

   ```
   prov.polling.period="7200"
   ```

3. Set the random polling start time to 01:00..

   ```
   prov.polling.time="01:00"
   ```

4. Set the polling period end time to 05:00.

   ```
   prov.polling.timeRandomEnd="05:00"
   ```

Configure SIP Subscription Timers

To improve the interoperability and performance of devices in the network environment, configure SIP subscription timers. You can configure a subscription expiry independently of the registration expiry.

Note: Per-registration configuration parameters override global parameters. If you don’t configure values for any user features, the phone uses the default values.

You can also configure the following:

- A subscription expiry independently of the registration expiry
- An overlap period for a subscription independently of the overlap period for the registration
- A subscription expiry and subscription overlap for global SIP servers and per-registration SIP servers

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Set the amount of time, in seconds, after which the phone attempts to resubscribe at the beginning of an overlap period. Replace *x* with the desired server key value. The default value is 60 seconds (1 minute). The value range is from 5 to 65535.

   ```
   voIpProt.server.x.expires.overlap="<value>"
   ```

2. Set the number of seconds before the expiration time returned by server *x* after which the phone attempts to resubscribe. If the server value is less than the configured overlap value, the phone tries to resubscribe at half the expiration time returned by the server. Replace *x* with the desired server key value. The default value is 3600 seconds (1 hour). The value range is 10 to 2147483647.

   ```
   voIpProt.server.x.subscribe.expires="<value>"
   ```

3. The phone's requested subscription period, in seconds, after which the phone attempts to resubscribe at the beginning of the overlap period. Replace *x* with the registered line number. Replace *y* with the desired server key value. The default value is 3600 seconds (1 hour). The value range is 10 to 2147483647.

   ```
   reg.x.server.y.subscribe.expires="<value>"
   ```

4. Set the amount of time, in seconds, after which the phone attempts to resubscribe at the beginning of an overlap period. Replace *x* with the registered line number.Replace *y* with the desired server key value. The default value is 60 (1 minute). The value range is 5 to 65535.

   ```
   reg.x.server.y.subscribe.expires.overlap="<value>"
   ```

Configure the SIP Instance Identification Settings

Configure the SIP instance to identify individual phones instead of using IP addresses.

If you register multiple phones using the same address of record (AOR), the server identifies the phones using their IP address. However, firewalls set up in these environments can regularly change the IP addresses of phones for security purposes. Enabling `reg.x.gruu` for a line provides a contact address to a specific user agent (UA) instance, which helps to route the request to the UA instance and is required in cases in which the REFER request must be routed to the correct UA instance.

This feature complies with RFC 3840.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

» Enable the phone to send *sip.instance* in the REGISTER request for line 1.

```
reg.1.gruu="1"
```

Configure SIP Header Warnings

Configure the warning field from a SIP header to display a dialog on the phone, for example, when a call transfer fails due to an invalid extension number.

For a list of supported SIP header warnings, see the Supported SIP Request Headers article in the Poly Online Support Center Knowledge Base.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Replace x in the parameter with the line your configuring

Procedure

1. Enable the phone to display a dialog with any received SIP warnings in the header.

   ```
   voIpProt.SIP.header.warning.enable="1"
   ```

2. Specify a list of accepted SIP warning codes to display. Leave Null to enable the phone to accept all warning codes. Note that only codes between 300 and 399 are supported.

   Separate multiple codes with a comma.

   ```
   voIpProt.SIP.header.warning.codes.accept="<Code1,Code2,Code3>"
   ```

IP Type-of-Service

The type-of-service field in an IP packet header consists of four type-of-service (TOS) bits and a 3-bit precedence field.

Type of Service (ToS) and the Differentiated Services Code Point (DSCP) allows specification of a datagrams desired priority and routing through low-delay, high-throughput, or highly-reliable networks.

You can configure the type of service specifically for RTP packets and call control packets, such as SIP signaling packets.

Enable IP Type-of Service

Type of Service (ToS) and the Differentiated Services Code Point (DSCP) enables specification of a datagram's desired priority and routing through low-delay, high-throughput, or highly-reliable networks.

The IP ToS header consists of four ToS bits and a 3-bit precedence field. DSCP replaces the older ToS specification and uses a 6-bit DSCP in the 8-bit differentiated services field (DS field) in the IP header.

Configure the type of service field RTP and call control packets for Quality of Service (QoS).

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

» Enable ToS.

```
qos.ethernet.tcpQosEnabled="1"
```

Configure IP Type-of-Service for Video

Configure the video-specific IP Type-of-Service parameters.

Ensure that `qos.ip.rtp.video.dscp` is set to NULL. Setting a value in `qos.ip.rtp.video.dscp` overrides other `qos.ip.rtip.video.*` parameters.

When you configure the video ToS parameters, the phone uses the `qos.ip.rtp.*` parameters for audio only.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Enable the reliability bits in the IP ToS field of the IP header used for RTP video.

   ```
   qos.ip.rtp.video.max_reliability="1"
   ```

2. Enable the throughput bits in the IP ToS field of the IP header used for RTP video.

   ```
   qos.ip.rtp.video.max_throughput="1"
   ```

3. Enable the min cost bits in the IP ToS field of the IP header used for RTP video.

   ```
   qos.ip.rtp.video.min_cost="1"
   ```

4. Enable the min delay bits in the IP ToS field of the IP header used for RTP video.

   ```
   qos.ip.rtp.video.min_delay="1"
   ```

5. Enable the precedence bits in the IP ToS field of the IP header used for RTP video.

   ```
   qos.ip.rtp.video.precedence="1"
   ```

SIP Server Registration

After the phone boots up, it registers to all configured servers.

Note: If you disable `reg.x.server.y.register` for a given server *y*, the phone doesn't register to that server. However, the INVITE fails over to that server if all higher priority servers are down.

Server 1 is the primary server and supports greater SIP functionality than other servers. For example, SUBSCRIBE/NOTIFY services used for features such as shared lines, presence, and BLF are established only with Server 1.

When the registration timer of each server registration expires, the phone attempts to reregister. If this is unsuccessful, normal SIP reregistration behavior (typically at intervals of 30 to 60 seconds) proceeds and continues until the registration is successful (for example, when the internet link is again operational).

While the primary server registration is unavailable, the next highest priority server in the list serves as the working server. As soon as the primary server registration succeeds, it returns to being the working server.

Configure VoIP Server DHCP Settings

Configure how the phone reacts to DHCP changes.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Enable the phone to check the DHCP server for an IP address.

   ```
   voIpProt.server.dhcp.available="1"
   ```

2. Set the DHCP option.

   The default is 128. The value ranges are 128 to 254.

   Note: If `reg.x.server.y.address` contains a value, it takes precedence even if a DHCP server is available.

   ```
   voIpProt.server.dhcp.option="<value>"
   ```

3. If you want the phone to request a string, set the type to 1. Otherwise, the phone requests an IP address.
   - 0 (default) - Request IP address
   - 1 - Request string

   ```
   voIpProt.server.dhcp.type="1"
   ```

4. If you want the outbound proxy address to be a string, set the type to 1. Otherwise, the outbound proxy requests an IP address.
   - 0 (default) - IP address
   - 1 - String

   ```
   voIpProt.OBP.dhcpv4.type="1"
   ```

5. Set the outbound proxy option for DHCPv4.

   The default is 120. The value range is 120 to 254.

   ```
   voIpProt.OBP.dhcpv4.option="<value>"
   ```

6. Define the outbound proxy option for DHCPv6.

The default is 21. The value range is 0 to 254.

```
voIpProt.OBP.dhcpv6.option="<value>"
```

SIP Signaling Failure for Outgoing Calls

At the start of a call, SIP signaling failure determines server availability.

Caution: If the phone uses DNS to resolve the address for servers, the phone attempts to contact the DNS server to resolve the address of all servers in its list before initiating a call. This may happen due to the DNS server being unavailable or because the TTL for the DNS records has expired. These attempts time out, but the timeout mechanism can cause long delays (for example, 2 minutes) before the phone call proceeds using the working server. To prevent this issue, use long TTLs. Poly recommends deploying an on-site DNS server as part of the redundancy solution.

SIP signaling failure depends on the SIP protocol you use.

- If the phone uses TCP, then the signaling fails if the connection fails or the Send fails.
- If the phone uses UDP, then the signaling fails if it detects ICMP or if the signal times out.

  If the phone attempts signaling through all servers in the list and this is the last server, then the signaling fails after the complete UDP timeout defined in RFC 3261. If it isn't the last server in the list, the phone uses the maximum number of retries using the configurable retry timeout.
- When the user initiates a call, the phone completes the following steps to connect the call:

  1. The phone tries to call the working server.
  2. If the working server doesn't respond correctly to the INVITE, the phone tries the next server in the list. The phone tries even if there's no current registration with these servers. This can happen if the internet connection goes down but the registration to the working server isn't yet expired.
  3. If the second server is also unavailable, the phone tries all possible servers (even those not currently registered) until it either succeeds in making a call or exhausts the list, at which point the call fails.

Static DNS Cache

Configure a set of static DNS NAPTR SRV or A records in the phone. You can enter a maximum of 12 record entries for DNS-A, DNS-NAPTR, and DNS-SRV records.

Note the following when configuring the static DNS cache:

- The phone makes an initial attempt to resolve a host name that is within the static DNS cache. For example, the phone makes a query to the DNS if the phone registers to its SIP registrar.
- If the initial DNS query returns no results for the host name or if the phone can't contact it, then the phone uses the values in the static cache for the configured time interval.
- After the configured time interval elapses, a resolution attempt of the host name again results in a query to the DNS.
- If a DNS query for a host name that is in the static cache returns a result, the phone uses the values from the DNS and ignores the statically cached values.

You can’t always configure the DNS cache to take advantage of failover redundancy. Use failover redundancy only when the configured IP server host name resolves (through an SRV or A record) to multiple IP addresses. Support for negative DNS caching enables faster failover when prior DNS queries return no results from the DNS server. For more information, see RFC 2308.

Configure the SIP Server

Configure the SIP server settings to use for the static DNS cache.

Note the following when you configure the static DNS cache:

- The phone makes an initial attempt to resolve a host name that is within the static DNS cache. For example, the phone makes a query to the DNS if the phone registers to its SIP registrar.
- If the initial DNS query returns no results for the host name or if the phone can’t contact it, then the phone uses the values in the static cache for the configured time interval.
- After the configured time interval elapses, a resolution attempt of the host name again results in a query to the DNS.
- If a DNS query for a host name that is in the static cache returns a result, the phone uses the values from the DNS and ignores the statically cached values.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Specify the call server used for this registration. Replace *x* with the desired line key value. Replace *y* with the desired server key value. The default is Null. The maximum string length is 255 characters.

   ```
   reg.x.server.y="<string>"
   ```

2. Specify the user or the user and host part of the registration SIP URI or the H.323 ID/extension. Replace *x* with the desired line key value.

   The default is Null.

   ```
   reg.x.address="<string>"
   ```

3. Specify the SIP server that accepts registrations. Replace *x* with the desired line key value. Replace *y* with the desired server key value.

   The default is Null.

   Note: If you set this parameter, if takes precedence even if the DHCP server is available. All the parameters you configure in this list override the parameters specified in `voIpProt.server.*`.

   ```
   reg.x.server.y.address="<string>"
   ```

4. Set the SIP server port that doesn’t specify registrations.

   The default is Null. The value range is 0 to 65535.

Note: If you set this parameter to 0, the port used depends on the value you set in `reg.x.server.y.transport`.

```
reg.x.server.y.port="<value>"
```

5. Set the transport method the phone uses to communicate with the SIP server.
   - DNSnaptr (Default) - If `reg.x.server.y.address` is a hostname and `reg.x.server.y.port` is 0 or Null, the phone does NAPTR then SRV lookups to try to discover the transport, ports and servers (as per RFC 3263).

     If `reg.x.server.y.address` is an IP address or if you provide a port for `reg.x.server.y.port`, then the phone uses UDP.
   - TCPpreferred - The phone prefers TCP as the transport but uses UDP if TCP fails.
   - UDPOnly - The phone uses only UDP.
   - TLS - If TLS fails, transport fails. Leave the port field empty (defaults to 5061) or set to 5061.
   - TCPOnly - The phone uses only TCP.

```
reg.x.server.y.transport="<value>"
```

Configure the Static DNS Cache with A Record IP Addresses

Configure the static DNS cache with A record IP addresses in the SIP server address fields.

Configure the SIP server information.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Configure the DNS cache IPv4 address. Replace *y* with the desired server key value.

   The default is Null.

```
dns.cache.A.y.address="<string>"
```

2. Configure the DNS cache hostname. Replace *y* with the desired server key value.

   The default is Null.

```
dns.cache.A.y.name="<string>"
```

3. Set the time period, in seconds, the phone uses the static cache record. Replace *y* with the desired server key value.

   The default is 300. The value range is 300 to 536870912.

   If a dynamic network request receives no response, this timer begins on first access of the static record. Once the timer expires, the next lookup for that record retries a dynamic network request before falling back on the static entry, and it resets TTL timer again.

```
dns.cache.A.y.ttl="<value>"
```

Configure the Static DNS Cache with NAPTR and SRV Records

Configure static DNS cache where your DNS provides NAPTR and SRV records.

Configure the SIP server information.

The following configuration causes the phone to build an SRV request based on the address you provide, including all subdomains:

- `voIpProt.SIP.outboundProxy.address="<string>"`
- `voIpProt.SIP.outboundProxy.port="0"`

This SRV request produces a list of servers ordered by weight and priority, enabling you to specify subdomains for separate servers, or you can create partitions of the same system. Note that while making SRV queries and transport is configured as TCP, the phone adds the prefix `<service.proto.>` to the configured address/FQDN but doesn't remove the subdomain prefix. The phone can resolve a single SRV query to many different servers, session border controllers (SBCs), or partitions ordered by weight and priority.

Alternatively, use DNS NAPTR to discover that services that are available at the root domain.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Configure the DNS cache NAPTR parameters. Replace *y* with the desired server key value.

```
dns.cache.NAPTR.y.name="<string>"
dns.cache.NAPTR.y.ttl="<value>"
dns.cache.NAPTR.y.order="<value>"
dns.cache.NAPTR.y.preference="<value>"
dns.cache.NAPTR.y.flag="<value>"
dns.cache.NAPTR.y.service="<value>"
dns.cache.NAPTR.y.regexp="<value>"
dns.cache.NAPTR.y.replacement="<string>"
```

2. Configure the DNS cache parameters. Replace *y* with the desired server key value.

```
dns.cache.SRV.y.name="<string>"
dns.cache.SRV.y.ttl="<value>"
dns.cache.SRV.y.priority="<value>"
dns.cache.SRV.y.weight="<value>"
dns.cache.SRV.y.port="<value>"
dns.cache.SRV.y.target="<string>"
```

DNS SIP Server Name Resolution

If a DNS name is given for a proxy/registrar address, the IP addresses associated with that name is discovered as specified in RFC 3263.

If a port is given, the only lookup is an A record. If no port is given, NAPTR and SRV records are tried before falling back on A records if NAPTR and SRV records return no results. If no port is given, and none is found through DNS, port 5060 is used. If the registration type is TLS, port 5061 is used.

Caution: Failure to resolve a DNS name is treated as signaling failure that causes a failover.

The following configuration causes the phone to build an SRV request based on the address you provide, including all subdomains.

Use the format:

- `voIpProt.SIP.outboundProxy.address`="*sip.example.com*"
- `voIpProt.SIP.outboundProxy.port`="*0*"

This SRV request produces a list of servers ordered by weight and priority, enabling you to specify sub-domains for separate servers, or you can create partitions of the same system. Please note that while making SRV queries and transport is configured as TCP, the phone adds the prefix `< service. proto.>` to the configured address/FQDN but doesn't remove the sub-domain prefix, for example sip.example.com becomes `_sip._tcp.sip.example.com`. A single SRV query can be resolved into many different servers, session border controllers (SBCs), or partitions ordered by weight and priority, for example, `voice.sip.example.com` and `video.sip.example.com`. Alternatively, use DNS NAPTR to discover what services are available at the root domain.

For Outgoing Calls (INVITE Fallback)

At the start of a call, server availability is determined by SIP signaling failure. SIP signaling failure depends on the SIP protocol being used:

- If TCP is used, then the signaling fails if the connection fails or the Send fails.
- If UDP is used, then the signaling fails if ICMP is detected or if the signal times out. If the signaling has been attempted through all servers in the list and this is the last server, then the signaling fails after the complete UDP timeout defined in RFC 3261. If it is not the last server in the list, the maximum number of retries using the configurable retry timeout is used.

  Caution: If DNS is used to resolve the address for Servers, the DNS server is unavailable, and the TTL for the DNS records has expired, the phone attempts to contact the DNS server to resolve the address of all servers in its list before initiating a call. These attempts timeout, but the timeout mechanism can cause long delays (for example, two minutes) before the phone call proceeds using the working server. To prevent this issue, long TTLs should be used. Poly recommends deploying an on-site DNS server as part of the redundancy solution.

When the user initiates a call, the phone completes the following steps to connect the call:

1. The phone tries to call the working server.
2. If the working server does not respond correctly to the INVITE, the phone tries and makes a call using the next server in the list (even if there is no current registration with these servers). This could be the case if the Internet connection has gone down, but the registration to the working server has not yet expired.
3. If the second server is also unavailable, the phone tries all possible servers (even those not currently registered) until it either succeeds in making a call or exhausts the list at which point the call fails.

Customer Phone Configuration

The phones at the customer site are configured as follows:

- Server 1 (the primary server) is configured with the address of the service provider call server. The IP address of the server(s) is provided by the DNS server, for example: `reg.1.server.1.address=voipserver.serviceprovider.com .`
- Server 2 (the fallback server) is configured to the address of the router/gateway that provides the fallback telephony support and is on-site, for example: `reg.1.server.2.address=172.23.0.1 .`

Caution: Be careful when using multiple servers per registration. It is possible to configure the phone for more than two servers per registration but ensure that the phone and network load generated by registration refresh of multiple registrations does not become excessive. This is of particular concern when a phone has multiple registrations with multiple servers per registration and some of these servers are unavailable.

Server Redundancy

VoIP deployments often require server redundancy. Server redundancy ensures phone high availability in the event that the phone loses connection to the server.

Poly phones support failover and fallback server redundancy. In some cases, you can deploy a combination of the two server redundancy types. Consult your SIP server provider for recommended methods of configuring phones and servers for failover configuration.

Note: The default value of the parameters `reg.x.server.y.failOver.concurrentRegistration` and `voIpProt.server.y.failOver.concurrentRegistration` is 0 for Poly devices. Use the *y* variable for redundant failover servers. If you want to register the server concurrently with other servers, set `reg.x.server.y.failOver.concurrentRegistration="1"` or `voIpProt.server.y.failOver.concurrentRegistration="1"`.

Note: The concurrent failover/fallback feature isn't compatible with Microsoft environments.

For more information, see *Technical Bulletin 5844: SIP Server Fallback Enhancements on Polycom Phones and Technical Bulletin 66546: Configuring Optional Re-Registration on Failover Behavior*.

Configuring Server Redundancy for a Registered Line

Configure a fallback server for a registered line on your phones.

Procedure

1. Set the phone to send a SIP request to the server that sent proxy authentication request in the event of a failover. Replace *x* with the desired line key value.

```
reg.x.auth.optimizedInFailover="1"
```

2. Configure the mode for failover failback. Replace *x* with the desired line key value.

Note: This setting overrides the configuration for `reg.x.server.y.failOver.failBack.mode`.

Set one of the following values:

- duration (default) - The phone tries the primary server again after the time specified by `reg.x.outboundProxy.failOver.failBack.timeout` expires.
- newRequests - All new requests are forwarded first to the primary server regardless of the last used server.
- DNSTTL - The phone tries the primary server again after a timeout equal to the DNS TTL you configured for the server the phone is registered to.

```
reg.x.outboundProxy.failOver.failBack.mode="<value>"
```

3. Configure the time to wait, in seconds before failback occurs. Replace *x* with the desired line key value.

Note: This setting overrides the configuration for `reg.x.server.y.failOver.failBack.timeout`.

The default is 3600. The value range is 0 (no timeout), and 60 to 65535.

```
reg.x.outboundProxy.failOver.failBack.timeout="<value>"
```

4. Enable the global and per-line `reRegisterOn` parameter. The existing registrations remain active. Replace *x* with the desired line key value.

```
reg.x.outboundProxy.failOver.failRegistrationOn="0"
```

5. Enable the global and per-line `reRegisterOn` and `failRegistrationOn` parameters. Signaling is accepted from and sent to a server that has failed. Replace *x* with the desired line key value.

```
reg.x.outboundProxy.failOver.onlySignalWithRegistered="0"
```

6. Configure the phone to attempt to register with (or via, for the outbound proxy scenario), the secondary server. Replace *x* with the desired line key value.

Note: This parameter overrides `reg.x.server.y.failOver.reRegisterOn`.

```
reg.x.outboundProxy.failOver.reRegisterOn="1"
```

7. Configure the SIP server port to which the phone sends all requests. Replace *x* with the desired line key value.

   The default is 0. The value range is 65535

```
reg.x.outboundProxy.port="<value>"
```

8. Configure the transport method the phone uses to communicate with the SIP server. Replace *x* with the desired line key value.
   - DNSnaptr (default)
   - TCPpreferred
   - UDPOnly
   - TLS

- TCPOnly

```
reg.x.outboundProxy.transport="<value>"
```

Configure Server Redundancy for VoIP

Configure a failback server for a VoIP registered line on your phones.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

In the following parameters, *x* refers to the line you're configuring.

Procedure

1. Set the server to register concurrently with other servers.

   ```
   voIpProt.server.y.failOver.concurrentRegistration="1"
   ```

2. Set the failback mode to set a timeout

   ```
   voIpProt.server.x.failOver.failBack.mode="duration"
   ```

3. Enter a time, in seconds, for the server to attempt to connect to the primary servers after a failback.

   The default is 3600. The value range is 0, 60 to 35535.

   ```
   voIpProt.server.x.failOver.failBack.timeout="<60 to 65535>"
   ```

4. Set how the server fails over.
   - 1 (default) - When set to 1, and the global or per-line `reRegisterOn` parameter is enabled, the phone silently invalidates an existing registration (if it exists), at the point of failing over.
   - 0 - When set to 0, and the global or per-line `reRegisterOn` parameter is enabled, existing registrations remain active. This means that the phone attempts failback without first attempting to register with the primary server to determine if it has recovered.

   ```
   voIpProt.server.x.failOver.failRegistrationOn="value"
   ```

5. Set how the server signals a fail over.
   - 1 (default) - When set to 1, and the global or per-line `reRegisterOn` and `failRegistrationOn` parameters are enabled, no signaling is accepted from or sent to a server that has failed until failback is attempted or failover occurs. If the phone attempts to send signaling associated with an existing call via an unregistered server (for example, to resume or hold a call), the call ends. No SIP messages are sent to the unregistered server.
   - 0 - When set to 0, and the global or per-line `reRegisterOn` and `failRegistrationOn` parameters are enabled, signaling is accepted from and sent to a server that has failed (even though failback hasn't been attempted or failover hasn't occurred).

   ```
   voIpProt.server.x.failOver.onlySignalWithRegistered="value"
   ```

6. Set which server the fail over signal is registered on.
   - 0 (default) - When set to 0, the phone won't attempt to register with the second.

- 1 - When set to 1, the phone attempts to register with (or by, for the outbound proxy scenario), the secondary server. If the registration succeeds (a 200 OK response with valid expires), signaling proceeds with the secondary server.

```
voIpProt.server.x.failOver.reRegisterOn="value"
```

Configure NAT

Configure the NAT settings for your phone.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Specify the IP address to advertise within SIP signaling. The IP address must match the external IP address used by the NAT device.

   ```
   nat.ip="<IP-Address>"
   ```

2. Specify the keep-alive interval, in seconds.

   The default is 0. The value range is 0 to 3600.

   ```
   nat.keepalive.interval="<value>
   ```

3. Set the initially allocated RTP port.

   The default is 0. The value range is 0 to 65440.

   Note: This parameter overrides the `tcpIpApp.port.rtp.mediaPortRangeStart` parameter.

   ```
   nat.mediaPortStart="<value>"
   ```

4. Set the port used for SIP signaling.

   The default is 0. The value range is 0 to 65535.

   Note: This parameter overrides the `voIpProt.local.port` parameter.

   ```
   nat.signalPort="<value>"
   ```

Real-Time Transport Protocol

Configure Real-Time Transport Protocol (RTP) for VoIP media on your device.

You can configure RTP ports for your environment in the following ways:

- Filter incoming packets by IP address or port.
- Reject packets arriving from a non-negotiated IP address, an unauthorized source, or non-negotiated port for greater security.

- Enforce symmetric port operation for RTP packets. When you don’t set the source port to the negotiated remote sink port, the phone rejects arriving packets.
- Fix the phone’s destination transport port to a specified value, regardless of the negotiated port.

  This is useful for communicating through firewalls. When you use a fixed transport port, all RTP traffic sends to and arrives on that specified port. Incoming packets are sorted by the source IP address and port, which enables the phone to multiplex multiple RTP streams.
- Specify the phone’s RTP port range. The phone supports conferencing and multiple RTP streams, and it can use several ports concurrently.

  As specified in RFC 1889, RFC 3550, and RFC 3551, the next-highest odd-numbered port sends and receives RTP.

Configure SIP RTP for FECC

Configure the SIP RTP settings for far end camera control (FECC).

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Enable the FECC port range configuration for OpenSIP registrations.

   ```
   tcpIpApp.port.rtp.feccPortRange.enable="1"
   ```

2. Specify the FECC port range start port for OpenSIP registrations.

   The default is 2372. The value range is 1024 to 65486.

   ```
   tcpIpApp.por.rtp.feccPortRangeStart="<value>"
   ```

3. Specify the FECC port range end port for OpenSIP registrations.

   The default is 2419. The value range is 1024 to 65486.

   ```
   tcpIpApp.port.rtp.feccPortRangeEnd="<value>"
   ```

Configure RTP Media Ports

Configure the RTP media ports.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Set the maximum supported end range for the audio ports.

   The default is 2269. The value range is 1024 to 65535.

Important: Each call increments the port number +2 to a maximum of 24 calls after the value resets to the starting point. Because port 5060 is used for SIP signaling, ensure that port 5060 is not within this range when you set this parameter. A call that attempts to use port 5060 has no audio.

```
tcpIpApp.port.rtp.mediaPortRangeEnd="<value>"
```

2. Set the starting port for RTP port range packets.

   The default is 2222. The value range is 1024 to 65436.

```
tcpIpApp.port.rtp.mediaPortRangeStart1="<value>"
```

Configure RTP Video Ports

Select a specific port range for RTP video ports.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Enable RTP video ports.

```
tcpIpApp.port.rtp.videoPortRange.enable="1"
```

2. Set the starting range for RTP video ports.

   The default is 2272. The value range is 1024 to 65486.

```
tcpIpApp.port.rtp.videoPortRangeStart="<value>"
```

3. Set the maximum supported end range for RTP video ports.

   The default is 2319. The value range is 1024 to 65535.

```
tcpIpApp.port.rtp.videoPortRangeEnd="<value>"
```

Configure STUN Settings

Configure the phone to act as a STUN client. The phone sends a request to a STUN server to discover the public IP and port(s). You can also configure the phone to send keep-alive messages to refresh NAT bindings.

Important: Configuring the following parameter(s) causes the phone to reboot. Dependencies and overrides may affect other parameters.

Procedure

1. Enable STUN.

When you enable `voIpProt.SIP.rport`, the phone adds the received IP address and port in the VIA header while generating response.

```
feature .nat.stun.enabled="1"
```

2. Enter the STUN server IP address.

```
nat.stun.server="<STUN server IP address>"
```

3. Optional: Enter a port number.

```
nat.stun.port="<STUN server port>"
```

4. Optional: Enable NAT traversal mode with STUN signaling for a particular line.

   In the parameter, replace *x* with the line number.

```
reg.x.nat.traversal.mode="Auto"
```

Enable GZIP Encoding

To reduce bandwidth consumption, configure the phone to send notifications to the server in GZIP format.

Procedure

- Enable GZIP encoding.

```
voIpProt.SIP.gzipEncoding.enable="1"
```