

Configuring Optional Re-Registration on Failover Behavior

This engineering advisory describes optional failover behaviors that you can set up to enhance server redundancy technologies your Polycom phones currently use. Specifically, you can configure the phone to perform SIP registration each time the phone performs failover or fallback to pro-actively detect failures and to increase security. In this engineering advisory, you'll learn about the enhanced behaviors, and how to set them up for your phone environment.

The information in this advisory is intended for system administrators. It applies to networks that use failover redundancy where one or both of the following constraints exist:

- Both registration and authentication are required for a phone to access a redundant network element.
- SIP traffic can only flow over one redundant element at a time.

The topics include:

- [Terminology](#)
- [About the Optional Failover Behaviors](#)
- [Setting Up the Optional Failover Behaviors](#)



Web Info: Related Information

Before you read this document, familiarize yourself with [SIP Server Fallback Enhancements](#) (Technical Notification 5844). You may also find it useful to read [Using a Static DNS Cache](#) (Technical Notification 36033), which explains how to set up a failover network using proxy servers. Both documents are available from the [Feature Descriptions & Technical Notifications](#) Support web page.



Note: Compatibility with Microsoft® Lync

The concurrent registration failover/fallback feature is not compatible with Microsoft® Lync.

This engineering advisory applies to Polycom® SoundPoint® IP 321, 331, 320, 330, 335, 430, 450, 550, 560, 650, and 670 phones running SIP software version 3.2.5, and Polycom UC Software 3.3.2 (or later) and 4.0.0 (or later). It also applies to Polycom® VVX® 500 and VVX® 1500 Business Media Phones running UC Software 4.0.1 or later.

Terminology

Before you read this advisory, take a moment to familiarize yourself with the following definitions:

Primary server The primary server is the highest priority server in a group of servers with an active registration. All communications route to the primary server first, unless the phone environment is configured otherwise.

Secondary server A secondary server backs up a primary server when the primary server fails. A secondary server may offer the same, or lesser, functionality than the primary server.

Server redundancy This refers to the practice of employing multiple servers so that when a primary server fails, a secondary server can take over.

Failover A type of server redundancy in which a secondary server takes over all the functions of the primary server when the primary server fails. No phone functionality is lost when the secondary server takes over.

Re-registration on failover A redundancy requirement in which a phone must successfully register with a server before communications can take place. If a server fails and a phone must communicate with another server (for example, a secondary server), the phone must register with the secondary server before communications can take place.

Failback A type of server redundancy in which a secondary server remains operational while communications with a primary server are retried to see if the primary server is functioning again. In certain configurations, the phone attempts to re-register with the primary server during failback.

Register transaction A register transaction associates a phone with a particular location, such as an IP address. A phone sends a message—called a ‘REGISTER’ message—informing the server of its location.

Registrar or Registrar server A registrar server accepts registrations, or location information, from phones and places this information in a database. Every phone must register its current location with a Registrar server before the phone can communicate with a server.

About the Optional Failover Behaviors

Server redundancy is often required in VoIP deployments to ensure continuity of phone service when a server fails or the connection between a phone and server fails.

Polycom phones rely on two server redundancy technologies: failover and fallback. Using these technologies, multiple servers are set up so that when the primary server fails, a secondary server can take over. Re-register on failover only applies to servers that use the failover method.

With failover, all servers (primary and secondary) share the same registration data. In this scenario, secondary servers support all the features that the primary server supports.

With re-register on failover disabled—the default behavior—when a phone’s registration request is diverted to a secondary server, the phone doesn’t have to register with the secondary server.

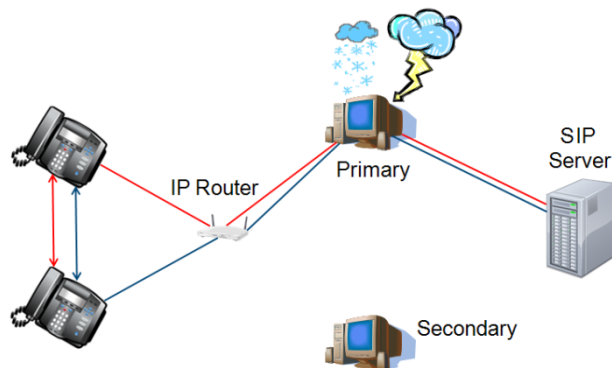
A potential issue with the default failover behavior is that some servers or intermediate SIP-aware devices may limit a phone’s functionality if the server hasn’t successfully processed the phone’s registration request. With re-register on failover enabled, when a phone’s SIP request is diverted to a secondary server, the phone will first register with the secondary server.

Polycom has added optional behaviors to enhance the current redundancy features. These behaviors include re-registration and recovery behaviors, as well as a behavior that controls how existing calls—calls that are established before a server fails—are treated.

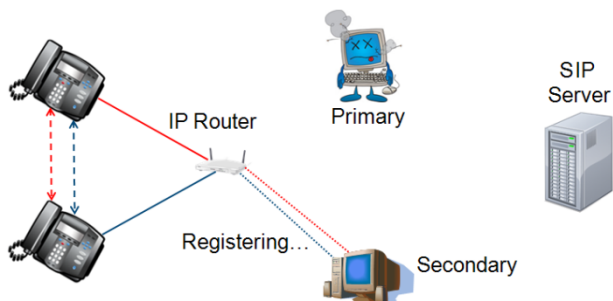
- **Re-registration behavior** The phone must complete a new registration with the failover server before communications can take place between the phone and the failover server.
- **Recovery behavior** This behavior requires phones to communicate with the server that processed the last successful transaction, rather than always with the primary server. If this behavior is configured, you have to set up rules to determine when the primary server is tried again (for example, whenever the phone has a new request, or after a specific period of time). The secondary server will remain operational while the phone is trying to re-register with the primary server (‘failback’).
- **Behavior for existing calls** This behavior controls the handling of calls established through the failed server after failover occurs. When this behavior is enabled, phones won’t communicate with failed servers that recover until failback succeeds. This helps avoid situations in which large numbers of phones toggle rapidly between servers when there is an intermittent failure.

The following diagrams show how a network uses the re-registration on failover behavior. In the diagrams, primary and secondary re-registration on failover-aware Session Border Controllers (RRoFO-aware SBCs) are set up so that if a phone can’t communicate with the primary RRoFO-aware SBC, the phone can attempt to register, and then communicate, with a secondary RRoFO-aware SBC.

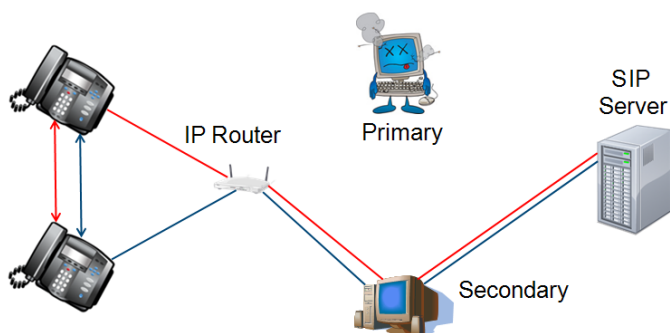
In the following diagram, phones are communicating with a primary RRoFO-aware SBC that is just about to fail.



When the primary RRoFO-aware SBC fails, phones can no longer communicate with it. Phones will attempt to register with a secondary RRoFO-aware SBC, as shown next.



If registration is successful (as shown next), phones can communicate with the secondary RRoFO-aware SBC, and traffic flow will continue without interruption.



Setting Up the Optional Failover Behaviors

To set up the new behaviors on your phones, you'll need to add and define some configuration file attributes, as shown in [Table 1: Configuration File Settings](#).

The parameters you set depend if your implementation uses proxy servers. Also, when you set the parameters, you can choose to set **volpProt** or **reg** parameters:

- **volpProt parameters** These parameters affect all line registrations on the phone and typically represent site-wide defaults.
- **reg parameters** These parameters are set on a per-line basis and control individual line settings. You'll need to define these parameters if different lines require different settings.



Admin Tip: Creating Configuration Files

When you create configuration files with your organization's modifications, Polycom recommends that you use the sample configuration templates—part of the UC Software 4.0.0 deliverables—as a guide. For more information, see the **Configuration File Templates** section in *Simplified Configuration Improvements in Polycom UC Software 3.3.0* (Technical Notification 60519), available from the [Feature Descriptions & Technical Notifications](#) Support web page.



Note: Enabling Proxy Servers

If your implementation uses proxy servers, make sure you enable the use of proxy servers in your configuration file(s). If you've defined **reg** parameters, set **reg.x.server.y.useOutboundProxy** to **1**. If you've defined **voipProt** parameters, set **voipProt.server.y.useOutboundProxy** to **1**. The parameters are in the **sip-interop.cfg** and **site.cfg** configuration files. For more information, see the latest *Polycom UC Software Administrators' Guide*, available from the [Polycom UC Software Support Center](#).



Note: Which Parameters Are Used?

If both **reg** and **voipProt** failover parameters are set, **reg** parameters supersede **voipProt** parameters.
If both **outboundProxy** and **server.x** failover parameters are set, **outboundProxy** parameters supersede **server.x** parameters.

To set up the failover behaviors:

- 1 On the provisioning server, create a configuration file for the phone. In this file, you'll define the parameters for the failover behaviors. (If you don't want to create a new file, you can add the failover parameters to an existing configuration file.)

If you create a new configuration file, you can base the file on a sample configuration template that's available in your software package. To find the sample files, navigate to **<provisioning server location>/Config**.

In the **Config** folder, choose one of the following templates based on the parameters you want to define:

- **sip-interop.cfg** To define voipProt server or outbound proxy server parameters
- **site.cfg** To define reg server parameters
- **reg-advanced.cfg** To define reg outbound proxy server parameters

2 In the file, define the failover parameters.

If you created a file based on one of the templates, the parameters are next to the following attributes:

- In **sip-interop.cfg**, the volpProt parameters are under:
 - » **volpProt.server** for volpProt.server.x attributes.
 - » **volpProt.SIP > volpProt.SIP.outboundProxy > volpProt.SIP.outboundProxy.failOver** and **volpProt.SIP > volpProt.SIP.outboundProxy > volpProt.SIP.outboundProxy.failOver > volpProt.SIP.outboundProxy.failOver.failBack** for proxy server attributes.
- In **site.cfg** and **reg-advanced.cfg**, the reg parameters are under the **reg** attribute.

The following table shows the parameters you need to define.

Table 1: Configuration File Settings

Parameter	Permitted Values	Default
volpProt.server.x.failOver. reRegisterOn	0, 1	0
<i>or</i>		
reg.x.server.y.failOver. reRegisterOn		
If you use proxy servers, set these attributes instead:		
volpProt.SIP.outboundProxy. failOver.reRegisterOn		
<i>or</i>		
reg.x.outboundProxy.failOver. reRegisterOn		

When set to *1*, the phone will attempt to register with (or via, for the outbound proxy scenario), the secondary server. If the registration succeeds (a 200 OK response with valid expires), signaling will proceed with the secondary server.

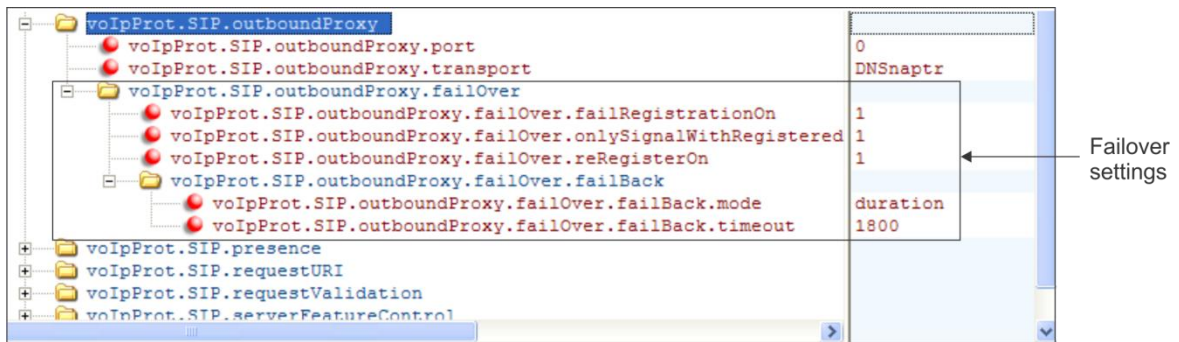
When set to *0*, the phone won't attempt to register with the secondary server, since the phone will assume that the primary and secondary servers share registration information.

Note: When this parameter is enabled, the *authOptimizedInFailover* parameter is automatically enabled. For more information about this parameter, see the latest Polycom UC Software Administrator's Guide, available by navigating to your phone's Support web page from the [Voice Support](#) web page.

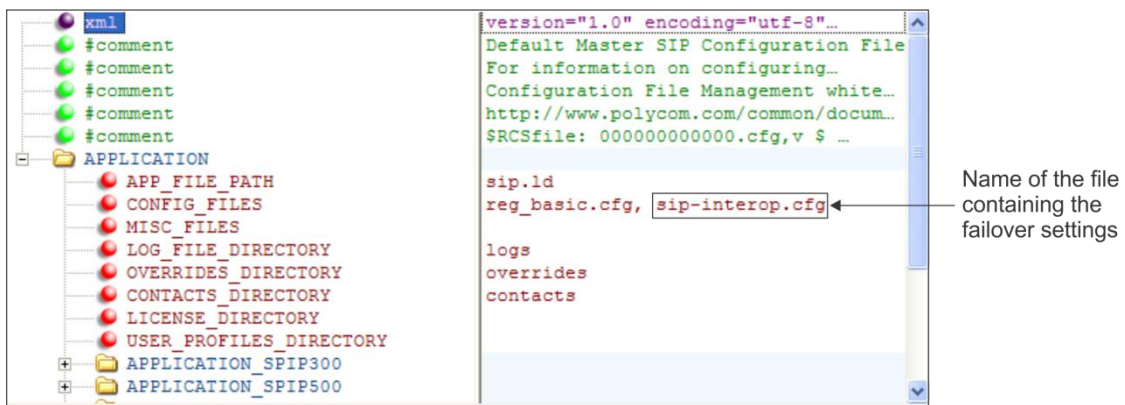
Parameter	Permitted Values	Default
volpProt.server.x.failOver. failRegistrationOn or reg.x.server.y.failOver. failRegistrationOn If you use proxy servers, set these attributes instead: volpProt.SIP.outboundProxy. failOver.failRegistrationOn or reg.x.outboundProxy.failOver. failRegistrationOn	0, 1	0
<p>When set to 1, and the reRegisterOn parameter is enabled, the phone will silently invalidate an existing registration (if it exists), at the point of failing over.</p> <p>When set to 0, and the reRegisterOn parameter is enabled, existing registrations will remain active. This means that the phone will attempt failback without first attempting to register with the primary server to determine if it has recovered.</p>		
volpProt.server.x.failOver. onlySignalWithRegistered or reg.x.server.y.failOver. onlySignalWithRegistered If you use proxy servers, set these attributes instead: volpProt.SIP.outboundProxy. failOver. onlySignalWithRegistered or reg.x.outboundProxy.failOver. onlySignalWithRegistered	0, 1	1
<p>When set to 1, and the 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 will end. No SIP messages will be sent to the unregistered server.</p> <p>When set to 0, and the reRegisterOn and failRegistrationOn parameters are enabled, signaling will be accepted from and sent to a server that has failed (even though failback hasn't been attempted or failover hasn't occurred).</p> <p>Note: This setting primarily affects signaling associated with existing dialogs that are RFC-mandated to communicate with the servers through which they were established. A new dialog's signaling will be sent through the 'current' server.</p>		

Parameter	Permitted Values	Default
volpProt.server.x.failOver. failBack.mode <i>or</i> reg.x.server.y.failOver.failBack. mode If you use proxy servers, set these attributes instead: volpProt.SIP.outboundProxy. failOver.failBack.mode <i>or</i> reg.x.outboundProxy.failOver. failBack.mode	newRequests, DNSTTL, registration, duration	newRequests
<p>When set to <i>newRequests</i>, all dialog-initiating requests are forwarded to the primary server first, regardless of the last server that was used.</p> <p>When set to <i>DNSTTL</i>, the primary server is retried after a timeout equal to the DNSTTL configured for the server the phone is registered to (or via, for the outbound proxy scenario).</p> <p>When set to <i>registration</i>, the primary server is retried when the current working server's registration requires renewal.</p> <p>When set to <i>duration</i>, the primary server is retried after the amount of time defined by the <i>timeout</i> parameter (as shown in the next row).</p> <p>Note: When failback mode is set to <i>DNSTTL</i> or <i>duration</i>, re-registration with the primary server takes place only if the phone is idle (that is, the phone has no calls or active lines). If the timeout period expires and call activity is detected, failback will be retried every second.</p>		
volpProt.server.x.failOver. failBack.timeout <i>or</i> reg.x.server.y.failOver.failBack. timeout If you use proxy servers, set these attributes instead: volpProt.SIP.outboundProxy. failOver.failBack.timeout <i>or</i> reg.x.outboundProxy.failOver. failBack.timeout	0, 60 - 65535	3600
<p>When failBack.mode is set to <i>duration</i>, the time in seconds after failing over to the current working server before the primary server becomes the first server to forward new requests to.</p> <p>If you set a value between 1 and 59, the timeout will be 60 seconds. If you set a value of 0, the primary server won't be selected as the first server to forward new requests to until a failover event occurs with the current working server.</p>		

The following example shows a sample **sip-interop.cfg** file.



- In the phone's master configuration file (for example, **000000000000.cfg** or **<MACaddress>.cfg**), add the name of the file you just created next to the **CONFIG_FILES** attribute, as shown next.



- Reboot your phone.

Trademarks

©2012, Polycom, Inc. All rights reserved.

POLYCOM®, the Polycom “Triangles” logo and the names and marks associated with Polycom’s products are trademarks and/or service marks of Polycom, Inc. and are registered and/or common law marks in the United States and various other countries. All other trademarks are property of their respective owners. No portion hereof may be reproduced or transmitted in any form or by any means, for any purpose other than the recipient’s personal use, without the express written permission of Polycom.

Disclaimer

While Polycom uses reasonable efforts to include accurate and up-to-date information in this document, Polycom makes no warranties or representations as to its accuracy. Polycom assumes no liability or responsibility for any typographical or other errors or omissions in the content of this document.

Limitation of Liability

Polycom and/or its respective suppliers make no representations about the suitability of the information contained in this document for any purpose. Information is provided “as is” without warranty of any kind and is subject to change without notice. The entire risk arising out of its use remains with the recipient. In no event shall Polycom and/or its respective suppliers be liable for any direct, consequential, incidental, special, punitive or other damages whatsoever (including without limitation, damages for loss of business profits, business interruption, or loss of business information), even if Polycom has been advised of the possibility of such damages.

Customer Feedback

We are constantly working to improve the quality of our documentation, and we would appreciate your feedback. Please send email to VoiceDocumentationFeedback@polycom.com.



Visit the [Polycom Voice Support Center](#) for software downloads, product documents, product licenses, troubleshooting tips, service requests, and more.