

排除IP电话注销故障 — 案例研究

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[SCCP保持连接和故障切换机制](#)

[保活](#)

[故障转移](#)

[正常故障切换](#)

[延迟故障切换](#)

[缺点](#)

[优势](#)

[SIP保持连接](#)

[到主](#)

[到辅助](#)

[所需日志](#)

[相关链接](#)

[从电话捕获](#)

[从CUCM捕获](#)

[案例研究1.2](#)

[问题说明](#)

[故障排除](#)

[分辨率](#)

[案例研究2.](#)

[问题说明](#)

[故障排除](#)

[分析](#)

[保活丢弃的原因](#)

简介

本文档介绍可用于排除配置故障的信息。

除网络级TCP保活机制外，Cisco IP电话还使用应用级保活机制。瘦呼叫控制协议(SCCP)和会话发起协议(SIP)设备的保持连接机制可确保设备在呼叫控制中保持注册。它们还用于通过呼叫控制重新建立设备连接。

先决条件

要求

本文档没有任何特定的要求。

使用的组件

本文档不限于特定的软件和硬件版本。

SCCP保持连接和故障切换机制

SCCP使用TCP协议进行传输，并使用端口2000和2443（用于安全）连接到Call Manager。SCCP电话在注册到Cisco Unified Communications Manager(CUCM)之前应与其建立TCP连接。随后，在端口2000上将发生TCP三次握手，以建立通信信道。电话通过向CUCM发送SYN（同步）来发起此连接，CUCM使用SYN、ACK（确认）做出响应。电话依次以ACK响应，TCP连接建立。

保活

有两种保持连接方法：应用级(SKINNY keep-alive)和网络级(TCP keep-alive)

故障转移

在理想情况下，SCCP电话会保持与主CUCM和第一个备份CUCM建立的TCP连接。SCCP电话将保持连接发送到它已建立TCP连接的所有CUCM。然后，主服务器响应SCCP保持连接。时间间隔是30秒到主服务器，60秒到备份服务器。

主CUCM以SCCP保持连接ACK（确认SCCP和TCP连接）作出回应。备份CUCM只向电话发送的keep-alive发送TCP ACK。当电话因呼叫管理器服务不可用或主CUCM的TCP连接本身不可用而无法备份CUCM时，它使用两种机制来检测主CM故障，它们是正常和延迟的。

正常故障切换

此方法使用一种算法来计算CUCM确认之前保持连接所花费的平均时间。

例如，如果CUCM对过去10000个保持连接所花费的平均时间是X秒，则电话将等待X秒，然后检测到CUCM故障。然后，它将尝试注册到备用CUCM。

延迟故障切换

在此机制中，电话会等待3个保持连接间隔以检测主CUCM的故障。

缺点

数据包的传输时间波动、延迟故障切换有助于避免不必要的注销。

传输时间波动示例（注意ping响应的延迟）：

```
64 bytes from 10.106.97.150: icmp_seq=1 ttl=63 time=0.100 ms
64 bytes from 10.106.97.150: icmp_seq=2 ttl=63 time=200 ms
64 bytes from 10.106.97.150: icmp_seq=3 ttl=63 time=0.180 ms
64 bytes from 10.106.97.150: icmp_seq=4 ttl=63 time=0.678 ms
64 bytes from 10.106.97.150: icmp_seq=5 ttl=63 time=590 ms
```

```
64 bytes from 10.106.97.150: icmp_seq=6 ttl=63 time=0.100 ms
64 bytes from 10.106.97.150: icmp_seq=7 ttl=63 time=345 ms
64 bytes from 10.106.97.150: icmp_seq=8 ttl=63 time=456 ms
64 bytes from 10.106.97.150: icmp_seq=9 ttl=63 time=0.345 ms
```

优势

此机制可用于延迟敏感网络。

SIP保持连接

SIP电话注册到CUCM，并根据CUCM中的设置每120秒发送一次保持连接。当电话将初始寄存器发送到主CUCM时，它将**Expires** 计时器设置为3600秒（在电话上应用的SIP配置文件中默认设置）。CUCM通过根据Service参数中设置的值将计时器修改为120秒来发送ACK。

因此，电话每120秒发送一次保持连接（实际上是115秒，即120减去SIP配置文件中配置的增量值，默认为5秒）。在这种情况下，电话每115秒发送一次保持连接。

SIP电话将注册消息交换到备份CUCM，**Expires** 字段设置为0。

到主

```
REGISTER sip:10.106.114.161 SIP/2.0

Via: SIP/2.0/TCP 10.106.114.185:53006;branch=z9hG4bKd451a4fa

From: <sip:5678@10.106.114.161>;tag=0024142ddf242c6644b6e5d2-f01c795a

To: <sip:5678@10.106.114.161>

Call-ID: 0024142d-df24000a-44da4e09-0de51424@10.106.114.185

Max-Forwards: 70

Date: Wed, 15 Jul 2015 12:42:56 GMT

CSeq: 11435 REGISTER

User-Agent: Cisco-CP7975G/9.3.1

Contact: <sip:9e9e1ffb-0206-4ea1-6d77-ba04a72017f7@10.106.114.185:53006;transport=tcp>;+sip.instance="<urn:uuid:00000000-0000-0000-0000-0024142ddf24>;+u.sip!devicename.ccm.cisco.com="SEP0024142DDF24";+u.sip!model.ccm.cisco.com="437"

Supported: replaces,join,sdp-anat,norefersub,resource-priority,extended-refer,X-cisco-callinfo,X-cisco-serviceuri,X-cisco-escapecodes,X-cisco-service-control,X-cisco-srtp-fallback,X-cisco-monrec,X-cisco-config,X-cisco-sis-6.0.0,X-cisco-xsi-8.5.1

Content-Length: 0

Expires: 3600
```

SIP/2.0 100 Trying

Via: SIP/2.0/TCP 10.106.114.185:53006;branch=z9hG4bKd451a4fa

From: <sip:5678@10.106.114.161>;tag=0024142ddf242c6644b6e5d2-f01c795a

To: <sip:5678@10.106.114.161>

Date: Wed, 15 Jul 2015 12:42:59 GMT

Call-ID: 0024142d-df24000a-44da4e09-0de51424@10.106.114.185

CSeq: 11435 REGISTER

Content-Length: 0

SIP/2.0 200 OK

Via: SIP/2.0/TCP 10.106.114.185:53006;branch=z9hG4bKd451a4fa

From: <sip:5678@10.106.114.161>;tag=0024142ddf242c6644b6e5d2-f01c795a

To: <sip:5678@10.106.114.161>;tag=1708299782

Date: Wed, 15 Jul 2015 12:42:59 GMT

Call-ID: 0024142d-df24000a-44da4e09-0de51424@10.106.114.185

CSeq: 11435 REGISTER

Expires: 120

Contact: <sip:9e9e1ffb-0206-4ea1-6d77-ba04a72017f7@10.106.114.185:53006;transport=tcp>;+sip.instance="<urn:uuid:00000000-0000-0000-0000-0024142ddf24>;+u.sip!devicename.ccm.cisco.com="SEP0024142DDF24";+u.sip!model.ccm.cisco.com="437"

Supported: X-cisco-srtp-fallback,X-cisco-sis-6.0.0

Content-Length: 0

到辅助

REGISTER sip:10.60.1.12:5060;transport=tcp SIP/2.0

Via: SIP/2.0/TCP 10.60.63.21:3784;rport;branch=z9hG4bKPjdcJ819aZtTctmvr0VBheV6p0uL8aC.pG

Max-Forwards: 70

From: <sip:6836@10.60.1.12>;tag=5oI-ew53.DGjTDu5LB9orkdDpZlccNbv

To: <sip:6836@10.60.1.12>

Call-ID: HxTK.m6BH9qxjstVwexTbhVnUxNeuxle

CSeq: 18800 REGISTER

Expires: 0

Contact: <sip:e2b0f175-feae-d664-befa-b7cd0837fcc6@10.60.63.21:5060;transport=TCP>;+sip.instance="<urn:uuid:00000000-0000-0000-0000-e0d1730aclb1>";+u.sip!devicename.ccm.cisco.com="SEPE0D1730AC1B1";+u.sip!model.ccm.cisco.com="592";expires=0;cisco-keep-alive

Content-Length: 0

所需日志

要确定电话未注册的原因，请收集概述的信息：

- 事件查看器应用和系统日志 — 为电话取消注册提供警报/错误代码，并使用这些代码我们可以继续进行故障排除。
- 同时从电话和CUCM（主要和备份）捕获数据包 — 有助于隔离问题网络视角。
- Cisco Call Manager跟踪。

相关链接

[从CUCM收集数据包捕获](#)

[从IP电话收集捕获](#)

[收集CUCM跟踪](#)

分析日志和数据包捕获

- 事件查看器应用程序日志打印EndPointUnregistered消息以及相关的原因代码。

Example: 31 uc-ucm-01 local7 3 : 41679: uc-ucm-01.pcce.local Jul 02 2015 06:22:31 UTC :

%UC_CALLMANAGER-3-EndPointUnregistered:

[%[DeviceName=SEPE0D1730A8137][IPAddress=10.60.98.210][Protocol=SIP][DeviceType=592][Description=Phone][Reason=13][IPAddrAttributes=0][LastSignalReceived=SIPStationDPrimaryLineTimeout][AppID=Cisco CallManager][ClusterID=StandAloneCluster][NodeID=uc-ucm-01]: An endpoint has unregistered

EndPointUnregistration的原因代码可在系统错误消息文档中找到。

读取Wireshark日志

收集两端的捕获信息时，验证电话发送的保持连接是否实际到达CUCM。

TCP数据包的序列号有助于在嗅探器捕获中轻松跟踪电话和CUCM之间的TCP流量。

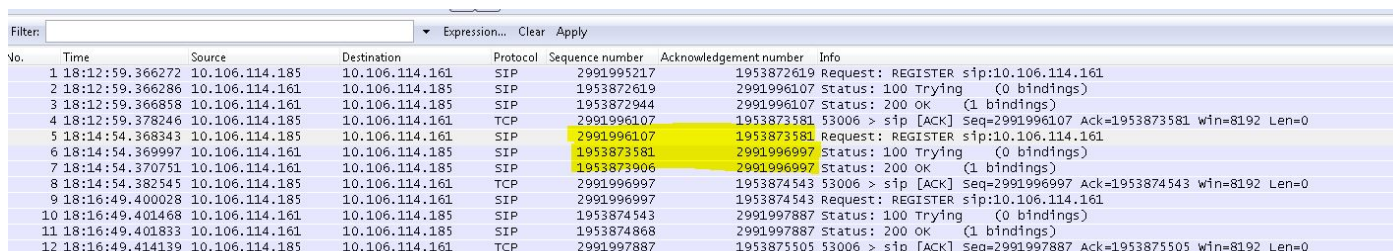
从电话捕获

Filter: ip.addr==10.106.114.185

No.	Time	Source	Destination	Protocol	Sequence Number	Acknowledgement Number	Info
200	18:14:49.051041	10.106.114.185	10.106.114.161	SIP	2991996907	1953873581	Request: REGISTER sip:10.106.114.161 (0 bindings)
201	18:14:49.053199	10.106.114.161	10.106.114.185	SIP	1953873581	2991996907	Status: 100 Trying (0 bindings)
202	18:14:49.053909	10.106.114.161	10.106.114.185	SIP	1953873906	2991996907	Status: 200 OK (1 bindings)
203	18:14:49.065591	10.106.114.185	10.106.114.161	TCP	2991996997	1953874543	53006 > sip [ACK] Seq=2991996997 Ack=1953874543 Win=8192 Len=0
484	18:16:44.077219	10.106.114.185	10.106.114.161	SIP	2991996997	1953874543	Request: REGISTER sip:10.106.114.161
485	18:16:44.079859	10.106.114.161	10.106.114.185	SIP	1953874543	2991997887	Status: 100 Trying (0 bindings)
486	18:16:44.079869	10.106.114.161	10.106.114.185	SIP	1953874868	2991997887	Status: 200 OK (1 bindings)
487	18:16:44.091359	10.106.114.185	10.106.114.161	TCP	2991997887	1953875505	53006 > sip [ACK] Seq=2991997887 Ack=1953875505 Win=8192 Len=0

电话发送序列号为2991996107的数据包，验证此数据包是否到达CUCM。

从CUCM捕获



No.	Time	Source	Destination	Protocol	Sequence number	Acknowledgement number	Info
1	18:12:59.366272	10.106.114.185	10.106.114.161	SIP	2991995217	1953872619	Request: REGISTER sip:10.106.114.161
2	18:12:59.366286	10.106.114.161	10.106.114.185	SIP	1953872619	2991996107	Status: 100 Trying (0 bindings)
3	18:12:59.366858	10.106.114.161	10.106.114.185	SIP	1953872944	2991996107	Status: 200 OK (1 bindings)
4	18:12:59.378246	10.106.114.185	10.106.114.161	TCP	2991996107	1953873581	53006 > sip [ACK] seq=2991996107 Ack=1953873581 win=8192 Len=0
5	18:14:54.368343	10.106.114.185	10.106.114.161	SIP	2991996107	1953873581	Request: REGISTER sip:10.106.114.161
6	18:14:54.369997	10.106.114.161	10.106.114.185	SIP	1953873581	2991996997	Status: 100 Trying (0 bindings)
7	18:14:54.370751	10.106.114.161	10.106.114.185	SIP	1953873906	2991996997	Status: 200 OK (1 bindings)
8	18:14:54.382545	10.106.114.185	10.106.114.161	TCP	2991996997	1953874543	53006 > sip [ACK] seq=2991996997 Ack=1953874543 win=8192 Len=0
9	18:16:49.400028	10.106.114.185	10.106.114.161	SIP	2991996997	1953874543	Request: REGISTER sip:10.106.114.161
10	18:16:49.401468	10.106.114.161	10.106.114.185	SIP	1953874543	2991997887	Status: 100 Trying (0 bindings)
11	18:16:49.401833	10.106.114.161	10.106.114.185	SIP	1953874868	2991997887	Status: 200 OK (1 bindings)
12	18:16:49.414139	10.106.114.185	10.106.114.161	TCP	2991997887	1953875505	53006 > sip [ACK] seq=2991997887 Ack=1953875505 win=8192 Len=0

在CUCM捕获中应看到电话嗅探器捕获中显示的序列号。

案例研究1.2

问题说明

SCCP电话会按固定间隔继续重新启动。

故障排除

事件查看器应用程序日志指示由于缺少保持活动，电话继续重新启动，错误代码为13。

Event Viewer Message.

从IP电话和CUCM收集数据包捕获。 在此场景中，从IP电话发送的最后一个保持连接未到达CUCM。

Image.

由于以下原因，“保活”被丢弃：

当电话发送ARP来获取CUCM的MAC地址时，响应来自具有ASA mac-address的ARP代理。显然，第一个响应并非来自CUCM。但是，由于电话首先收到帧，因此它使用另一台设备的MAC地址将帧发送到交换机。

当ASA上启用ARP代理时，通常会发生这种情况。

The screenshot displays a network traffic analysis interface. At the top, there's a menu bar with options like File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, and Help. Below the menu is a toolbar with various icons. A filter is set to 'eth.addr == 58:0a:20:fb:07:1f'. The main area shows a list of packets with columns for No., Time, Source, Destination, Protocol, Sequence number, Acknowledgement number, and Info. The detailed view below shows a frame of 78 bytes on wire, which is an Ethernet II frame containing an ARP request. The source MAC is 58:0a:20:fb:07:1f and the destination MAC is f4:0f:1b:1e:26:a9. The ARP request is for the IP address 10.10.10.202.

分辨率

在ASA上禁用ARP代理以解决问题。

案例研究2.

问题说明

Cisco IP电话型号8961电话每16分钟重置一次，并注册到辅助CUCM。2分钟后，电话将回退到主CUCM，此循环继续。

故障排除

从电话收集数据包捕获和CUCM跟踪。取消注册是因为IP电话丢失了SIP保持连接。

分析

SIP电话注册到CUCM，并且根据CUCM中的设置每120秒发送一次保持连接。

当电话发送初始注册时，它会将过期计时器设置为3600秒（在电话上应用的SIP配置文件中默认设置）。CUCM通过根据Service参数中设置的值将计时器修改为120秒来确认。

电话每120秒发送一次保持连接（保持连接间隔为115秒，即120减去SIP配置文件中配置的增量值（默认为5秒）。在这种情况下，电话每115秒发送一次keepalive。

在此问题场景中，电话以115秒的速度发送第一个keepalive，并在网络中丢弃。这会导致电话在0.01秒（100毫秒）内重新传输keepalive。它从CUCM获得对REGISTER请求的响应。

现在，电话以115秒的速度发送第二个keepalive，并在网络中丢弃。现在，电话将其REGISTER重试间隔增加到0.02秒（200毫秒）。

每次电话在115后发送keepalive数据包时，它都会被丢弃到网络中，这会使电话重新传输数据包。此外，电话的重试间隔也呈指数级增加。在几次此类保持连接后，电话重试次数将增加到14秒。

电话在14秒后重新传输，并从CUCM获得ACK。

下次电话发送保持连接时，它会丢失，然后电话在28秒后重新传输REGISTER请求。CUCM无法等待28秒，只等待15秒（在115秒后），然后发送注销信号。

保持连接时间和RTO总和为16分钟几秒。

由于来自CUCM的注销信号，16分钟后，电话注册到辅助CUCM，2分钟后，电话重新注册到主CUCM，然后继续。

1930	22:00:17.299388	10.147.230.189	178.215.139.22	SIP	9722	4203 Request: REGISTER sip:178.215.139.22
1931	22:00:17.437220	10.147.230.189	178.215.139.22	SIP	9723	[TCP Retransmission] Request: REGISTER sip:178.215.139.22
1934	22:00:17.471894	178.215.139.22	10.147.230.189	TCP	4563	7623 sip > 50708 [ACK] Seq=4563 Ack=7623 win=22559 Len=0
1935	22:00:17.473022	178.215.139.22	10.147.230.189	SIP	4563	7623 Status: 100 Trying (0 bindings)
1936	22:00:17.473815	178.215.139.22	10.147.230.189	SIP	4900	7623 Status: 200 OK (1 bindings)
1938	22:00:17.507164	10.147.230.189	178.215.139.22	TCP	7623	5548 50708 > sip [ACK] Seq=7623 Ack=5548 win=17940 Len=0
3318	22:00:17.474709	10.147.230.189	178.215.139.22	SIP	7623	5548 Request: REGISTER sip:178.215.139.22
3323	22:00:17.892520	10.147.230.189	178.215.139.22	SIP	7623	5548 [TCP Retransmission] Request: REGISTER sip:178.215.139.22
3324	22:00:17.907067	178.215.139.22	10.147.230.189	TCP	5548	8524 sip > 50708 [ACK] Seq=5548 Ack=8524 win=25319 Len=0
3325	22:00:17.908564	178.215.139.22	10.147.230.189	SIP	5548	8524 Status: 100 Trying (0 bindings)
3326	22:00:17.909810	10.147.230.189	178.215.139.22	TCP	8524	5885 50708 > sip [ACK] Seq=8524 Ack=5885 win=17940 Len=0
3327	22:00:17.909452	178.215.139.22	10.147.230.189	SIP	5885	8524 Status: 200 OK (1 bindings)
3328	22:00:17.909808	10.147.230.189	178.215.139.22	TCP	8524	6534 50708 > sip [ACK] Seq=8524 Ack=6534 win=17940 Len=0
4711	23:00:07.909779	10.147.230.189	178.215.139.22	SIP	8524	6534 Request: REGISTER sip:178.215.139.22
4722	23:00:08.747602	10.147.230.189	178.215.139.22	SIP	8524	6534 [TCP Retransmission] Request: REGISTER sip:178.215.139.22
4723	23:00:08.762120	178.215.139.22	10.147.230.189	TCP	6534	9425 sip > 50708 [ACK] Seq=6534 Ack=9425 win=27030 Len=0
4724	23:00:08.763291	178.215.139.22	10.147.230.189	SIP	6534	9425 Status: 100 Trying (0 bindings)
4725	23:00:08.763658	10.147.230.189	178.215.139.22	TCP	9425	6871 50708 > sip [ACK] Seq=9425 Ack=6871 win=17940 Len=0
4726	23:00:08.764030	178.215.139.22	10.147.230.189	SIP	6871	9425 Status: 200 OK (1 bindings)
4727	23:00:08.764032	10.147.230.189	178.215.139.22	TCP	9425	7519 50708 > sip [ACK] Seq=9425 Ack=7519 win=17940 Len=0
6117	23:02:03.764972	10.147.230.189	178.215.139.22	SIP	9425	7519 Request: REGISTER sip:178.215.139.22
6137	23:02:05.442842	10.147.230.189	178.215.139.22	SIP	9425	7519 [TCP Retransmission] Request: REGISTER sip:178.215.139.22
6138	23:02:05.457251	178.215.139.22	10.147.230.189	TCP	7519	10326 sip > 50708 [ACK] Seq=7519 Ack=10326 win=28832 Len=0
6139	23:02:05.458324	178.215.139.22	10.147.230.189	SIP	7519	10326 Status: 100 Trying (0 bindings)
6140	23:02:05.458692	10.147.230.189	178.215.139.22	TCP	10326	7856 50708 > sip [ACK] Seq=10326 Ack=7856 win=17940 Len=0
6141	23:02:05.459023	178.215.139.22	10.147.230.189	SIP	7856	10326 Status: 200 OK (1 bindings)
6142	23:02:05.459397	10.147.230.189	178.215.139.22	TCP	10326	8505 50708 > sip [ACK] Seq=10326 Ack=8505 win=17940 Len=0
7520	23:04:00.460122	10.147.230.189	178.215.139.22	SIP	10326	8505 Request: REGISTER sip:178.215.139.22
7519	23:04:03.831737	10.147.230.189	178.215.139.22	SIP	10326	8505 [TCP Retransmission] Request: REGISTER sip:178.215.139.22
7560	23:04:03.832323	178.215.139.22	10.147.230.189	TCP	8505	11227 sip > 50708 [ACK] Seq=8505 Ack=11227 win=50634 Len=0
7561	23:04:03.834245	178.215.139.22	10.147.230.189	SIP	8505	11227 Status: 100 Trying (0 bindings)
7562	23:04:03.834726	178.215.139.22	10.147.230.189	SIP	8842	11227 Status: 200 OK (1 bindings)
7563	23:04:03.834728	10.147.230.189	178.215.139.22	TCP	11227	8842 50708 > sip [ACK] Seq=11227 Ack=8842 win=17940 Len=0
7564	23:04:03.835387	10.147.230.189	178.215.139.22	TCP	11227	9489 50708 > sip [ACK] Seq=11227 Ack=9489 win=17940 Len=0
8947	23:05:58.836796	10.147.230.189	178.215.139.22	SIP	11227	9489 Request: REGISTER sip:178.215.139.22
9036	23:06:05.567238	10.147.230.189	178.215.139.22	SIP	11227	9489 [TCP Retransmission] Request: REGISTER sip:178.215.139.22
9030	23:06:05.567350	178.215.139.22	10.147.230.189	TCP	9489	12128 sip > 50708 [ACK] Seq=9489 Ack=12128 win=32436 Len=0
9031	23:06:05.568414	178.215.139.22	10.147.230.189	SIP	9489	12128 Status: 100 Trying (0 bindings)
9032	23:06:05.568832	10.147.230.189	178.215.139.22	TCP	12128	9826 50708 > sip [ACK] Seq=12128 Ack=9826 win=17940 Len=0
9033	23:06:05.568933	178.215.139.22	10.147.230.189	SIP	9826	9826 Status: 200 OK (1 bindings)

保活丢弃的原因

当交换机端口配置了端口安全时，端口老化已配置了非活动计时器。计时器设置为比SIP保活计时器小一分钟。这导致交换机端口每一分钟刷新一次电话MAC。由于SIP保持连接间隔为每2分钟一次，因此数据包会不断被丢弃。