

E&M Digital CAS 信令的 EM_PARK 问题疑难解答

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景理论](#)

[规则](#)

[问题](#)

[解决方案](#)

[冒充应答](#)

[相关信息](#)

简介

在Cisco2600的数字E&M信令，3600和MC3810路由器平台，一些T1/E1时间间隙能陷在EM_PARK状态。当您发出**show voice call summary**命令时，这可视。本文解释如何发现并解决此问题。

此输出显示一些时间间隙在EM_PARK状态。一个时间间隙在EM_PARK状态没有使用语音呼叫。

```
Router#show voice call summary
PORT      CODEC      VAD      VTSP STATE      VPM STATE
=====      =====      ===      =====      =====
1/0:0.1    -          -          -              EM_ONHOOK
1/0:0.2    -          -          -              EM_PARK
1/0:0.3    -          -          -              EM_PARK
1/0:0.4    -          -          -              EM_ONHOOK
1/0:0.5    -          -          -              EM_ONHOOK
```

先决条件

要求

本文档没有任何特定的要求。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- 硬件— Cisco2600、Cisco3600、Cisco VG200和MC3810路由器

- 软件—所有

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

背景理论

在T1 CAS，例如，WINK启动信令，当PBX去摘机时，路由器/网关侧时间间隙留在空闲(EM_ONHOOK)状态，直到呼叫由远程目的地应答。对EM_OFFHOOK的路由器时间间隙状态变换，当呼叫由远程目的地应答。

如果呼叫不连接，路由器/网关播放带内重排序声音给呼叫方。因为在路由器端的信道状态仍然是EM_ONHOOK，路由器无法挂断信道。在呼叫方挂断后，PBX需要更改其信道状态从摘机到挂机。

有时，PBX不在ABCD帮助下传送挂机信息，过渡了。路由器有此的一应急方案呼叫的冒充应答。没有假应答应急方案，信道在EM_PARK的状态无限地暂停。欲知更多信息，请参阅[Fake Answer部分](#)。

注意：如果语音网关路由器机箱没有适当地，电子被接地呼叫可以在EM_PARK状态被滞留在一些T1信道。参考硬件安装指南关于电子地面的更多信息。

规则

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

问题

有两可能，主要理由时间间隙在EM_PARK状态被滞留：

- 数字信号信号处理器(DSP)是坏的并且有任一个硬件或软件问题。
- PSTN switch/PBX发送一个连续挂机信号到路由器，并且不发布它。

解决方案

这些是解决方案对此问题：

如果在您的系统的时间间隙在EM_PARK状态被滞留，请检查DSP。参考[排除故障在NM-HDV的DSP Cisco 2600/3600系列路由器的](#)为了检查DSP。

如果DSP运行，问题可以在PSTN switch/PBX侧或Cisco IOS (路由器/网关不开始冒充应答步骤)。欲知更多信息，请参阅[Fake Answer部分](#)。

冒充应答

Cisco路由器/网关等待默认值30秒(请使用[超时等待版本](#)和[超时call-disconnect](#)命令为了更改这些值)，在知道后时间间隙需要从PBX设置为挂机，当播放交换机忙音时。

如果这不发生，路由器迁移时间间隙向EM_PARK状态并且启动有10秒的持续时间的另一个计时器。如果PBX不在10秒钟的持续时间之后仍然去挂机，路由器欺骗PBX。路由器发送一一秒钟持续时

间的冒充应答然后去挂机。

在路由器发送假应答信号后，路由器启动五分钟另一个计时器。如果PBX去挂机，计时器终止和路由器转换时间间隙向EM_ONHOOK状态。否则，在五分钟之后它发送一秒期限的另一个假应答信号。路由器重复此进程，直到PBX去挂机。路由器强制PBX清除呼叫。

注意：因为清除，此回答的转换没有更新对任何计费记录实际呼叫。当答案和用户对一秒期限呼叫，很可能收费，但是，PBX了解它。

如果DSP关联与时间间隙在EM_PARK状态活跃和健康的，并且问题持续，请运行[debug vpm all](#)和[debug vtsp all](#)命令为了发现Cisco IOS是否尝试发送冒充应答。

注意：您需要运行调试超过五分钟。

注意：在大多数情况下，如果DSP是坏的，路由器不执行假应答应急方案。参考[排除故障在NM-HDV的DSP Cisco 2600/3600系列路由器的](#)欲知更多信息。

此debug输出显示时间间隙如何变得卡住在EM_PARK，并且如何假应答应急方案工作。

```
Jan 11 17:19:00.767: htsp_dsp_message: SEND/RESP_SIG_STATUS: state=0xC timestamp
=44262 systime=31305235
Jan 11 17:19:00.767: htsp_process_event:
[4/1:1(10), EM_ONHOOK, E_DSP_SIG_1100]em_onhook_offhook htsp_setup_ind
!--- Offhook signal is received from the switch. Jan 11 17:19:00.767: [4/1:1(10)]
get_local_station_id calling num= calling name= calling time=01/11 17:19 Jan 11 17:19:00.767:
vtsp_tsp_call_setup_ind (sdb=0x62BB7B14, tdm_info=0x0, tsp_info=0x62BB4050, calling_number=
calling_oct3 = 0x0, called_number= called_oct3 = 0x81, oct3a=0x0): peer_tag=0 Jan 11
17:19:00.767: : ev.clg.clir is 0 ev.clg.clid_transparent is 0 ev.clg.null_orig_clg is 1
ev.clg.calling_translated is false Jan 11 17:19:00.767: htsp_timer - 3000 msec Jan 11
17:19:00.767: vtsp_do_call_setup_ind Jan 11 17:19:00.767: vtsp_allocate_cdb,cdb 0x62DCEA70 Jan
11 17:19:00.767: vtsp_do_call_setup_ind: Call ID=112722, guid=62DC4230 Jan 11 17:19:00.767:
vtsp_do_call_setup_ind: type=0, under_spec=1640890368, name=, id0=10, id1=1, id2=25038,
calling=, called= subscriber=RegularLine Jan 11 17:19:00.767: vtsp_do_normal_call_setup_ind Jan
11 17:19:00.771: cc_api_call_setup_ind (vdbPtr=0x62BB7FA0, callInfo={called=
,called_oct3=0x81,calling=,calling_oct3=0x0,calling_oct3a=0x0,calling_xlated=fal
se,subscriber_type_str=RegularLine,fdest=0,peer_tag=0, prog_ind=3},callID=0x62DC 40DC) Jan 11
17:19:00.771: cc_api_call_setup_ind type 1 , prot 0 Jan 11 17:19:00.771: vtsp_insert_cdb,cdb
0x62DCEA70 Jan 11 17:19:00.771: vtsp_open_voice_and_set_params Jan 11 17:19:00.771:
dsp_close_voice_channel: [4/1:1:32995] packet_len=8 channel_id=3 packet_id=75 Jan 11
17:19:00.771: dsp_open_voice_channel_20: [4/1:1:32995] packet_len=16 channel_id=3 packet_id=74
alaw_ulaw_select=0 associated_signaling_channel=130 time_slot=2 serial_port=0 Jan 11
17:19:00.771: vtsp_modem_proto_from_cdb: cap_modem_proto 1073741824 Jan 11 17:19:00.771:
vtsp_modem_proto_from_cdb: cap_modem_proto 1073741824 Jan 11 17:19:00.771: dsp_encap_config:
[4/1:1:32995] packet_len=30 channel_id=3 packet_id=92 TransportProtocol 2 t_ssrc=0x0 r_ssrc=0x0
t_vpxcc=0x0 r_vpxcc=0x0 sid_support=1, tse_payload=65535, seq_num=0x0, redundancy=0 Jan 11
17:19:00.771: dsp_set_playout_delay Jan 11 17:19:00.771: dsp_set_playout: [4/1:1:32995]
packet_len=18 channel_id=3 packet_id=76 mode=1 initial=60 min=40 max=200 fax_nom=300
dsp_set_playout_delay_config Jan 11 17:19:00.771: dsp_set_playout_config Jan 11 17:19:00.771:
mode 0, init 60, min 40, max 200 playout default Jan 11 17:19:00.771:
dsp_set_playout_config:mode 0, init 60, min 40, max 200 Jan 11 17:19:00.771:
dsp_set_playout_config: [4/1:1:32995] packet_len=18 channel_id=3 packet_id=76 mode=1 initial=60
min=40 max=200 fax_nom=300 Jan 11 17:19:00.771: dsp_echo_canceler_control: echo_cancel: 1 Jan 11
17:19:00.771: dsp_echo_canceler_control: [4/1:1:32995] echo_cancel 1, disable_hpf 0, flags=0x0,
threshold=-21 Jan 11 17:19:00.771: dsp_echo_canceler_control: [4/1:1:32995] packet_len=12
channel_id=3 packet_id=66 flags=0x0, threshold=-21 Jan 11 17:19:00.771: set_gains: FXx/E&M: msg-
>message.set_codec_gains.out_gain=0 Jan 11 17:19:00.771: dsp_set_gains: [4/1:1:32995]
packet_len=12 channel_id=3 packet_id=91 in_gain=0 out_gain=0 Jan 11 17:19:00.771:
dsp_vad_enable: [4/1:1:32995] enable: packet_len=12 channel_id=3 packet_id=78 thresh=-38 Jan 11
17:19:00.771: cc_process_call_setup_ind (event=0x62E63ACC) Jan 11 17:19:00.771: >>>>CCAPI handed
```

cid 32995 with tag 0 to app "DEFAULT" Jan 11 17:19:00.771: sess_appl:
ev(24=CC_EV_CALL_SETUP_IND), cid(32995), disp(0) Jan 11 17:19:00.771: sess_appl:
ev(SSA_EV_CALL_SETUP_IND), cid(32995), disp(0) Jan 11 17:19:00.771: ssaCallSetupInd Jan 11
17:19:00.771: ccCallSetContext (callID=0x80E3, context=0x62DFBCF0) Jan 11 17:19:00.771:
ssaCallSetupInd cid(32995), st(SSA_CS_MAPPING),oldst(0), ev (24)ev-
>e.evCallSetupInd.nCallInfo.finalDestFlag = 0 Jan 11 17:19:00.771: ccCallSetupAck
(callID=0x80E3) Jan 11 17:19:00.771: ccGenerateTone (callID=0x80E3 tone=8) Jan 11 17:19:00.771:
ccCallReportDigits (callID=0x80E3, enable=0x1) Jan 11 17:19:00.771: vtsp_report_digit_control:
enable=1: digit reporting enabled Jan 11 17:19:00.771: cc_api_call_report_digits_done
(vdbPtr=0x62BB7FA0, callID=0x80E3, disp=0) Jan 11 17:19:00.771: : vtsp_get_digit_timeouts Jan 11
17:19:00.771: sess_appl: ev(52=CC_EV_CALL_REPORT_DIGITS_DONE), cid(32995), disp(0) Jan 11
17:19:00.771: cid(32995)st(SSA_CS_MAPPING)ev (SSA_EV_CALL_REPORT_DIGITS_DONE)
oldst(SSA_CS_MAPPING)cfid(-1)csz(0)in(1)fDest(0) Jan 11 17:19:00.771: ssaReportDigitsDone
cid(32995) peer list: (empty) Jan 11 17:19:00.771: ssaReportDigitsDone callid=32995 Enable
succeeded Jan 11 17:19:00.771: ccGenerateTone (callID=0x80E3 tone=8) Jan 11 17:19:00.771:
vtsp:[4/1:1:32995, S_SETUP_INDICATED, E_CC_SETUP_ACK] Jan 11 17:19:00.775: act_setup_ind_ack Jan
11 17:19:00.775: vtsp_modem_proto_from_cdb: cap_modem_proto 0 Jan 11 17:19:00.775:
vtsp_modem_proto_from_cdb: cap_modem_proto 0 Jan 11 17:19:00.775: dsp_encap_config:
[4/1:1:32995] packet_len=30 channel_id=3 packet_id=92 TransportProtocol 2 t_ssrc=0x0 r_ssrc=0x0
t_vpxcc=0x0 r_vpxcc=0x0 sid_support=1, tse_payload=65535, seq_num=0x0, redundancy=0 Jan 11
17:19:00.775: dsp_voice_mode: [4/1:1:32995] cdb 62DCEA70, cdb->codec_params.modem 2,
inband_detect flags 0x21 Jan 11 17:19:00.775: map_dtmf_relay_type--digit relay mode: 2 Jan 11
17:19:00.775: dsp_voice_mode: [4/1:1:32995] packet_len=24 channel_id=3 packet_id=73
coding_type=1 voice_field_size=160 VAD_flag=0 echo_length=256 comfort_noise=1 inband_detect=33
digit_relay_mode=2 AGC_flag=0act_setup_ind_ack: modem_mode = 0, fax_relay_on = 1 Jan 11
17:19:00.775: act_setup_ind_ack(): dsp_dtmf_mode() dsp_dtmf_mode(VTSP_TONE_DTMF_MODE) Jan 11
17:19:00.775: dsp_dtmf_mode: [4/1:1:32995] packet_len=10 channel_id=3 packet_id=65 dtmf_or_mf=0
Jan 11 17:19:00.775: vtsp_timer: 31305236 Jan 11 17:19:00.775: vtsp:[4/1:1:32995,
S_DIGIT_COLLECT, E_CC_GEN_TONE] Jan 11 17:19:00.775: act_gen_tone Jan 11 17:19:00.775:
dsp_cp_tone_off: [4/1:1:32995] packet_len=8 channel_id=3 packet_id=71 Jan 11 17:19:00.775:
dsp_cp_tone_on: [4/1:1:32995] packet_len=38 channel_id=3 packet_id=72 tone_id=4 n_freq=2
freq_of_first=350 freq_of_second=440 amp_of_first=5514 amp_of_second=5514 direction=1
on_time_first=65535 off_time_first=0 on_time_second=0 off_time_second=0 Jan 11 17:19:00.775:
vtsp:[4/1:1:32995, S_DIGIT_COLLECT, E_CC_GEN_TONE] Jan 11 17:19:00.775: act_gen_tone Jan 11
17:19:00.775: dsp_cp_tone_off: [4/1:1:32995] packet_len=8 channel_id=3 packet_id=71 Jan 11
17:19:00.775: dsp_cp_tone_on: [4/1:1:32995] packet_len=38 channel_id=3 packet_id=72 tone_id=4
n_freq=2 freq_of_first=350 freq_of_second=440 amp_of_first= 5514 amp_of_second=5514 direction=1
on_time_first=65535 off_time_first=0 on_time4_second=0 off_time_second=0 Jan 11 17:19:00.775:
htsp_process_event: [4/1:1(10), EM_WAIT_SETUP_ACK, E_HTSP_SETUP_ACK]em_wait_setup_ack_get_ack
Jan 11 17:19:00.775: htsp_timer_stop Jan 11 17:19:00.775: htsp_timer2 - 172 msec Jan 11
17:19:00.947: htsp_process_event: [4/1:1(10), EM_WAIT_SETUP_ACK,
E_HTSP_EVENT_TIMER2]em_wait_prewink_timer **Jan 11 17:19:00.947: em_offhook (0)[recEive and
transMit4/1:1(10)] set signal st
ate = 0x8em_onhook (200)[recEive and transMit4/1:1(10)] set signal state = 0x0
!--- A wink of duration 200 msec is sent out to the switch. Jan 11 17:19:01.471:
vtsp_process_dsp_message: MSG_TX_DTMF_DIGIT_BEGIN: digit=9, rtp_timestamp=0xED31C493 Jan 11
17:19:01.471: vtsp:[4/1:1:32995, S_DIGIT_COLLECT, E_DSP_DTMF_DIGIT_BEGIN] Jan 11 17:19:01.471:
act_report_digit_begin Jan 11 17:19:01.471: cc_api_call_digit_begin (dstVdbPtr=0x0,
dstCallId=0xFFFFFFFF F, srcCallId=0x80E3, digit=9, digit_begin_flags=0x1,
rtp_timestamp=0xED31C493 rtp_expiration=0x0, dest_mask=0x1) Jan 11 17:19:01.471: sess_appl:
ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(32995), disp(0) Jan 11 17:19:01.471:
cid(32995)st(SSA_CS_MAPPING)ev(SSA_EV_DIGIT_BEGIN) oldst(SSA_CS_MAPPING)cfid(-
1)csz(0)in(1)fDest(0) Jan 11 17:19:01.471: ssaIgnore cid(32995), st(SSA_CS_MAPPING),oldst(0),
ev(10) Jan 11 17:19:01.503: vtsp_process_dsp_message: MSG_TX_DTMF_DIGIT_OFF: digit=9,
duration=65 Jan 11 17:19:01.503: vtsp:[4/1:1:32995, S_DIGIT_COLLECT, E_DSP_DTMF_DIGIT] Jan 11
17:19:01.503: act_report_digit_end Jan 11 17:19:01.503: vtsp_timer_stop: 31305308 Jan 11
17:19:01.503: dsp_cp_tone_off: [4/1:1:32995] packet_len=8 channel_id=3 pa cket_id=71 Jan 11
17:19:01.503: cc_api_call_digit_end (dstVdbPtr=0x0, dstCallId=0xFFFFFFFF, srcCallId=0x80E3,
digit=9,duration=65,xruleCallingTag=0,xruleCalledTag=0, dest_mask=0x1), digi t_tone_mode=0 Jan
11 17:19:01.503: htsp_digit_ready: digit = 39 Jan 11 17:19:01.503: vtsp_timer: 31305308 Jan 11
17:19:01.503: htsp_process_event: [4/1:1(10), EM_OFFHOOK, E_VTSP_DIGIT]em_offhook_digit_collect
Jan 11 17:19:01.503: sess_appl: ev(9=CC_EV_CALL_DIGIT_END), cid(32995), disp(0) Jan 11
17:19:01.503: cid(32995)st(SSA_CS_MAPPING)ev(SSA_EV_CALL_DIGIT) oldst(SSA_CS_MAPPING)cfid(-
1)csz(0)in(1)fDest(0) Jan 11 17:19:01.503: ssaDigit Jan 11 17:19:01.503: ssaDigit, 0. sct-**

>digit , sct->digit len 0, usrDigit 9, digit_tone_mode=0 Jan 11 17:19:01.503: ssaDigit,1.
callinfo.called , digit 9, callinfo.calling , x rulecallingtag 0, xrulecalledtag 0 Jan 11
17:19:01.503: ssaDigit, 7. callinfo.calling , sct->digit 9, result 1 Jan 11 17:19:01.603:
vtsp_process_dsp_message: MSG_TX_DTMF_DIGIT_BEGIN: digit=1, rtp_timestamp=0xED31C493 Jan 11
17:19:01.603: vtsp:[4/1:1:32995, S_DIGIT_COLLECT, E_DSP_DTMF_DIGIT_BEGIN] Jan 11 17:19:01.603:
act_report_digit_begin Jan 11 17:19:01.603: cc_api_call_digit_begin (dstVdbPtr=0x0,
dstCallId=0xFFFFFFFF F, srcCallId=0x80E3, digit=1, digit_begin_flags=0x1,
rtp_timestamp=0xED31C493 rtp_expiration=0x0, dest_mask=0x1) Jan 11 17:19:01.603: sess_appl:
ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(32995), disp(0) Jan 11 17:19:01.603:
cid(32995)st(SSA_CS_MAPPING)ev(SSA_EV_DIGIT_BEGIN) oldst(SSA_CS_MAPPING)cfid(-
1)csz(0)in(1)fDest(0) Jan 11 17:19:01.603: ssaIgnore cid(32995), st(SSA_CS_MAPPING),oldst(0),
ev(10) Jan 11 17:19:01.643: vtsp_process_dsp_message: MSG_TX_DTMF_DIGIT_OFF: digit=1,
duration=75 Jan 11 17:19:01.643: vtsp:[4/1:1:32995, S_DIGIT_COLLECT, E_DSP_DTMF_DIGIT] Jan 11
17:19:01.643: act_report_digit_end Jan 11 17:19:01.643: vtsp_timer_stop: 31305322 Jan 11
17:19:01.643: cc_api_call_digit_end (dstVdbPtr=0x0, dstCallId=0xFFFFFFFF, srcCallId=0x80E3,
digit=1,duration=75,xruleCallingTag=0,xruleCalledTag=0, dest_mask=0x1), digit_tone_mode=0 Jan 11
17:19:01.643: htsp_digit_ready: digit = 31 Jan 11 17:19:01.643: vtsp_timer: 31305322 Jan 11
17:19:01.643: htsp_process_event: [4/1:1(10), EM_OFFHOOK, E_VTSP_DIGIT]em_offhook_digit_collect
Jan 11 17:19:01.643: sess_appl: ev(9=CC_EV_CALL_DIGIT_END), cid(32995), disp(0) Jan 11
17:19:01.643: cid(32995)st(SSA_CS_MAPPING)ev(SSA_EV_CALL_DIGIT) oldst(SSA_CS_MAPPING)cfid(-
1)csz(0)in(1)fDest(0) Jan 11 17:19:01.643: ssaDigit Jan 11 17:19:01.643: ssaDigit, 0. sct-
>digit 9, sct->digit len 1, usrDigit 1, digit_tone_mode=0 Jan 11 17:19:01.643: ssaDigit,1.
callinfo.called , digit 91, callinfo.calling , xrulecallingtag 0, xrulecalledtag 0 Jan 11
17:19:01.643: ssaDigit, 7. callinfo.calling , sct->digit 91, result 1 Jan 11 17:19:01.743:
vtsp_process_dsp_message: MSG_TX_DTMF_DIGIT_BEGIN: digit=8, rtp_timestamp=0xED31C493 Jan 11
17:19:01.743: vtsp:[4/1:1:32995, S_DIGIT_COLLECT, E_DSP_DTMF_DIGIT_BEGIN] Jan 11 17:19:01.743:
act_report_digit_begin Jan 11 17:19:01.743: cc_api_call_digit_begin (dstVdbPtr=0x0,
dstCallId=0xFFFFFFFF F, srcCallId=0x80E3, digit=8, digit_begin_flags=0x1,
rtp_timestamp=0xED31C493 rtp_expiration=0x0, dest_mask=0x1) Jan 11 17:19:01.743: sess_appl:
ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(32995), disp(0) Jan 11 17:19:01.743:
cid(32995)st(SSA_CS_MAPPING)ev(SSA_EV_DIGIT_BEGIN) oldst(SSA_CS_MAPPING)cfid(-
1)csz(0)in(1)fDest(0) Jan 11 17:19:01.743: ssaIgnore cid(32995), st(SSA_CS_MAPPING),oldst(0),
ev(10) radius_decrypt: null length Jan 11 17:19:01.843: vtsp_process_dsp_message:
MSG_TX_DTMF_DIGIT_OFF: digit=8, duration=75 Jan 11 17:19:01.843: vtsp:[4/1:1:32995,
S_DIGIT_COLLECT, E_DSP_DTMF_DIGIT] Jan 11 17:19:01.843: act_report_digit_end Jan 11
17:19:01.843: vtsp_timer_stop: 31305342 Jan 11 17:19:01.843: cc_api_call_digit_end
(dstVdbPtr=0x0, dstCallId=0xFFFFFFFF, srcCallId=0x80E3,
digit=8,duration=75,xruleCallingTag=0,xruleCalledTag=0, dest_mask=0x1), digi t_tone_mode=0 Jan
11 17:19:01.843: htsp_digit_ready: digit = 38 Jan 11 17:19:01.843: vtsp_timer: 31305342 Jan 11
17:19:01.843: htsp_process_event: [4/1:1(10), EM_OFFHOOK, E_VTSP_DIGIT]em_offhook_digit_collect
Jan 11 17:19:01.843: sess_appl: ev(9=CC_EV_CALL_DIGIT_END), cid(32995), disp(0) Jan 11
17:19:01.843: cid(32995)st(SSA_CS_MAPPING)ev(SSA_EV_CALL_DIGIT) oldst(SSA_CS_MAPPING)cfid(-
1)csz(0)in(1)fDest(0) Jan 11 17:19:01.843: ssaDigit Jan 11 17:19:01.843: ssaDigit, 0. sct-
>digit 91, sct->digit len 2, usrDigit 8, d igit_tone_mode=0 Jan 11 17:19:01.843: ssaDigit,1.
callinfo.called , digit 918, callinfo.calling , xrulecallingtag 0, xrulecalledtag 0 Jan 11
17:19:01.843: ssaDigit, 7. callinfo.calling , sct->digit 918, result -1 Jan 11 17:19:01.843:
ccCallDisconnect (callID=0x80E3, cause=0x1C tag=0x0) Jan 11 17:19:01.843: vtsp:[4/1:1:32995,
S_DIGIT_COLLECT, E_CC_DISCONNECT] Jan 11 17:19:01.843: act_pre_con_disconnect Jan 11
17:19:01.843: vtsp_ring_noan_timer_stop: 31305342 Jan 11 17:19:01.843: dsp_cp_tone_off:
[4/1:1:32995] packet_len=8 channel_id=3 pa cket_id=71 Jan 11 17:19:01.843: dsp_voice_mode:
[4/1:1:32995] cdb 62DCEA70, cdb->codec_para ms.modem 2, inband_detect flags 0x21 Jan 11
17:19:01.843: map_dtmf_relay_type--digit relay mode: 2 Jan 11 17:19:01.843: dsp_voice_mode:
[4/1:1:32995] packet_len=24 channel_id=3 pa cket_id=73 coding_type=1 voice_field_size=160
VAD_flag=0 echo_length=256 comfort_noise=1 inband_detect=33 digit_relay_mode=2 AGC_flag=0 Jan
11 17:19:01.843: **dsp_cp_tone_on: [4/1:1:32995] packet_len=38 channel_id=3 pa
cket_id=72 tone_id=3 n_freq=2 freq_of_first=480 freq_of_second=620amp_of_first=
5206 amp_of_second=2928 direction=1 on_time_first=250 off_time_first=250
on_time_second=0 off_time_second=0**
Jan 11 17:19:01.843: vtsp_timer: 31305342
Jan 11 17:19:01.843: htsp_pre_connect_disconnect, cdb = 62DCEA70 cause = 1C
*!--- Since the call is disconnected because the number received is "unassigned" !--- or
"invalid" the router starts to play the reorder !--- tone and a timer, which is the wait-release
!--- timeout timer, starts with default 30 seconds. !--- This call is disconnected !--- prior to
the connect state.* Jan 11 17:19:01.843: htsp_process_event: [4/1:1(10), EM_OFFHOOK,

```
E_HTSP_PRE_CONN_DISC] Jan 11 17:19:31.844: vtsp_main: timer: 31308342
!--- The wait-release timer expires after 30 seconds. Jan 11 17:19:31.844: vtsp:[4/1:1:32995,
S_WAIT_RELEASE_NC, E_TIMER]
!--- The VTSP module is in a wait release state for that call. It also receives !--- event
timer, which means that the timer expires so that it !--- goes into another state. Jan 11
17:19:31.844: act_pre_con_disc_rel htsp_release_req: cause 28, no_onhook 0 Jan 11 17:19:31.844:
htsp_process_event: [4/1:1(10), EM_OFFHOOK,
E_HTSP_RELEASE_REQ]em_offhook_release
Jan 11 17:19:31.844: htsp_timer_stop2 em_onhook (0)[recEive and transMit4/1:1(10
)] set signal state = 0x0
Jan 11 17:19:31.844: htsp_timer_stop
Jan 11 17:19:31.844: em_start_timer: 400 ms
Jan 11 17:19:31.844: htsp_timer - 400 msec
!--- HTSP receives an event that requests the release of !--- the time slot and it goes into EM
wait !--- onhook state. But, it cannot do anything since it says I am onhook already. !--- Also,
the router starts a timer of 400 msec. Jan 11 17:19:32.296: htsp_process_event: [4/1:1(10),
EM_WAIT_ONHOOK,
E_HTSP_EVENT_TIMER]em_wait_timeout
Jan 11 17:19:32.296: em_stop_timers
Jan 11 17:19:32.296: htsp_timer_stop
Jan 11 17:19:32.296: em_start_timer: 400 ms
Jan 11 17:19:32.296: htsp_timer - 400 msec
!--- When the 400 msec timer expires, HTSP gets into EM clear pending state. !--- It also starts
another timer of 400 msec. Jan 11 17:19:32.696: htsp_process_event: [4/1:1(10), EM_CLR_PENDING,
E_HTSP_EVENT_TIMER]em_clr_timeout Jan 11 17:19:32.696: em_stop_timers Jan 11 17:19:32.696:
htsp_timer_stop Jan 11 17:19:32.696: em_start_timer: 10000 ms Jan 11 17:19:32.696: htsp_timer -
10000 msec Jan 11 17:19:32.700: htsp_dsp_message: SEND/RESP_SIG_STATUS: state=0xC timestamp=1533
systime=31308428 Jan 11 17:19:32.700: htsp_process_event: [4/1:1(10), EM_PARK,
E_DSP_SIG_1100]em_park_offhook !--- When the 400 msec timer expires, the router puts the time
slot into !--- the EM_PARK state, and it starts another timer of 10 seconds. !--- The router
still sees the ABCD=1100 from the switch. Jan 11 17:19:42.760: htsp_process_event: [4/1:1(10),
EM_PARK, E_HTSP_EVENT_TIMER]em_park_timerhtsp_report_onhook_sig
Jan 11 17:19:42.760: em_offhook (0)[recEive and transMit4/1:1(10)] set signal st
ate = 0x8em_onhook (1000)[recEive and transMit4/1:1(10)] set signal state = 0x0
Jan 11 17:19:42.760: htsp_timer2 - 300000 msec
Jan 11 17:19:42.760: htsp_process_event: [4/1:1(10),
EM_PARK, E_HTSP_EVENT_TIMER]em_park_timerhtsp_report_onhook_sig
Jan 11 17:19:42.760: em_offhook (0)[recEive and transMit4/1:1(10)]
set signal state = 0x8em_onhook (1000)[recEive and
transMit4/1:1(10)] set signal state = 0x0
Jan 11 17:19:42.760: htsp_timer2 - 300000 msec
!--- As seen from the timestamps, when the timer expires in ten seconds, !--- the router goes
offhook for one second (1000 msec) and then onhook. !--- It also starts another timer of 300000
msec (5 minutes).
```

[相关信息](#)

- [语音技术支持](#)
- [语音和统一通信产品支持](#)
- [Cisco IP 电话故障排除](#)
- [技术支持和文档 - Cisco Systems](#)