

Fehlerbehebung bei EM_PARK-Problemen für E&M Digital CAS Signaling

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Hintergrundtheorie](#)

[Konventionen](#)

[Problem](#)

[Lösung](#)

[Gefälschte Antwort](#)

[Zugehörige Informationen](#)

Einführung

Bei der digitalen E&M-Signalisierung auf Cisco Router-Plattformen 2600, 3600 und MC3810 können einige T1/E1-Zeitsteckplätze im EM_PARK-Zustand stecken. Dies wird angezeigt, wenn Sie den Befehl **show voice call summary** aufrufen. In diesem Dokument wird erläutert, wie dieses Problem behoben wird.

Diese Ausgabe zeigt, dass einige Zeitsteckplätze im EM_PARK-Zustand sind. Ein Zeitschlitz im EM_PARK-Status wird nicht für Sprachanrufe verwendet.

```
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
```

Voraussetzungen

Anforderungen

Für dieses Dokument bestehen keine speziellen Anforderungen.

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Software- und

Hardwareversionen:

- Hardware - Cisco Router 2600, Cisco 3600, Cisco VG200 und MC3810
- Software - Alle

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Hintergrundtheorie

In T1 CAS beispielsweise erfolgt die Wink-Start-Signalisierung. Wenn das PBX-System abgehoben wird, bleibt der Router-/Gateway-seitige Zeitschlitz im Leerlaufzustand (EM_ONHOOK), bis der Anruf von einem Remote-Ziel angenommen wird. Der Status des Zeitsteckplatzes des Routers wechselt zu EM_OFFHOOK, wenn der Anruf vom Remote-Ziel entgegengenommen wird.

Wenn der Anruf nicht verbunden wird, gibt der Router/Gateway dem Anrufer die Inband-Reorder-Töne wieder. Da der Kanalstatus auf der Routerseite immer noch EM_ONHOOK ist, kann der Router den Kanal nicht auflegen. Nachdem der Anrufer aufgelegt hat, muss das PBX-System seinen Kanalstatus von "Abgenommener Hörer" in "Abgenommen" ändern.

In einigen Fällen senden die PBX-Systeme die Nachrichten bei aufgelegtem Hörer nicht mithilfe der ABCD-Übergänge. Der Router hat eine Lösung für diese so genannte gefälschte Antwort. Ohne die gefälschte Antwort-Problemumgehung hängen die Kanäle unbegrenzt im Zustand EM_PARK. Weitere Informationen finden Sie im Abschnitt "[Fake Answer](#)".

Hinweis: Anrufe können auf einigen T1-Kanälen im EM_PARK-Status festgehalten werden, wenn das Gehäuse des Sprach-Gateway-Routers nicht ordnungsgemäß geerdet ist. Weitere Informationen zur elektrischen Erdung finden Sie im Hardware-Installationsleitfaden.

Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie unter [Cisco Technical Tips Conventions](#) (Technische Tipps zu Konventionen von Cisco).

Problem

Es gibt zwei mögliche Hauptgründe dafür, dass der Zeitschlitz im EM_PARK-Zustand feststeckt:

- Der digitale Signalprozessor (DSP) ist defekt und weist Hardware- oder Softwareprobleme auf.
- Der PSTN-Switch bzw. die PBX-Anlage sendet ein kontinuierliches Abhaksignal an den Router und gibt es nicht.

Lösung

Hier sind die Lösungen für dieses Problem:

Wenn die Zeitsteckplätze in Ihrem System im EM_PARK-Zustand feststecken, überprüfen Sie die DSPs. Informationen zur Überprüfung der DSPs finden Sie unter [Troubleshooting the DSP on NM-HDV for Cisco Routers der Serien 2600/3600 \(Fehlerbehebung beim DSP auf NM-HDV\)](#).

Wenn die DSPs aktiv sind, kann sich das Problem entweder auf der Seite des PSTN-Switches/PBX oder auf der Seite des Cisco IOS® (Router/Gateway startet den gefälschten Antwortvorgang nicht) befinden. Weitere Informationen finden Sie im Abschnitt "[Fake Answer](#)".

Gefälschte Antwort

Der Router/Gateway von Cisco wartet auf einen Standardwert von 30 Sekunden (verwenden Sie die Befehle [für das Wartelassen](#) und das [Timeouts zum Abbrechen von Anrufen](#), um diese Werte zu ändern), nachdem er weiß, dass der Zeitschlitz vom PBX auf "On Hook" gesetzt werden muss, während er den Reorder-Ton wiedergibt.

Wenn dies nicht der Fall ist, verschiebt der Router den Zeitschlitz in den EM_PARK-Status und startet einen weiteren Timer mit einer Dauer von 10 Sekunden. Wenn das PBX-System nach 10 Sekunden immer noch nicht aufgelegt wird, trickt der Router das PBX-System. Der Router sendet eine *gefälschte Antwort* von einer Sekunde Dauer und nimmt dann den Hörer auf.

Nachdem der Router das gefälschte Antwortsignal gesendet hat, startet der Router einen weiteren Timer von fünf Minuten. Wenn das PBX-System aufgelegt wird, wird der Timer angehalten, und der Router wechselt den Zeitschlitz in den Status EM_ONHOOK. Andernfalls sendet er nach fünf Minuten ein weiteres gefälschtes Antwortsignal von einer Dauer von einer Sekunde. Der Router wiederholt diesen Vorgang, bis das PBX-System abgehoben wird. Der Router zwingt das PBX-System, den Anruf zu beenden.

Hinweis: Dieser Wechsel zur Anrufbeantwortung wird nicht auf einen Buchhaltungsdatensatz aktualisiert, da der eigentliche Anruf gelöscht wird. Das PBX-System versteht es jedoch als Antwort, und dem Benutzer wird wahrscheinlich eine Sekunde lang ein Anruf in Rechnung gestellt.

Wenn der mit dem Zeitschlitz im EM_PARK-Zustand verknüpfte DSP aktiv und fehlerfrei ist und das Problem weiterhin besteht, führen Sie den Befehl [debug vpm all](#) aus und [debug vtsp all](#)-Befehle aus, um festzustellen, ob Cisco IOS versucht, die gefälschte Antwort zu senden.

Hinweis: Sie müssen das Debuggen für mehr als fünf Minuten ausführen.

Hinweis: In den meisten Fällen führt der Router, wenn der DSP fehlerhaft ist, keine Problemumgehung durch. Weitere Informationen finden Sie unter [Troubleshooting the DSP on NM-HDV for Cisco Routers der Serien 2600/3600 \(Fehlerbehebung für DSP auf NM-HDV\)](#).

Diese Debug-Ausgabe zeigt, wie ein Zeitschlitz in EM_PARK feststeckt und wie die gefälschte Antwort-Problemumgehung funktioniert.

```
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
systeme=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 state = 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).

Zugehörige Informationen

- [Unterstützung von Sprachtechnologie](#)
- [Produkt-Support für Sprach- und Unified Communications](#)
- [Fehlerbehebung bei Cisco IP-Telefonie](#)
- [Technischer Support und Dokumentation - Cisco Systems](#)