

# Resolver problemas los problemas del EM\_PARK para la señalización del CAS Digital E&M

## Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Teoría Precedente](#)

[Convenciones](#)

[Problema](#)

[Solución](#)

[Respuesta falsa](#)

[Información Relacionada](#)

## Introducción

En la señalización de E&M digital en el Cisco 2600, 3600, y las plataformas del router MC3810, algunos slots de tiempo T1/E1 pueden conseguir pegados en el estado del EM\_PARK. Esto es visible cuando usted publica el **comando show voice call summary**. Este documento explica cómo resolver problemas este problema.

Esta salida muestra que algunos slots de tiempo están en el estado del EM\_PARK. Un slot de tiempo en el estado del EM\_PARK no se utiliza para las llamadas de voz.

```
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
```

## prerrequisitos

### Requisitos

No hay requisitos específicos para este documento.

### Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- Hardware — Cisco 2600, Cisco 3600, Cisco VG200, y Routers MC3810
- Software — Todos

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

## [Teoría Precedente](#)

En el T1 CAS, por ejemplo, sigue habiendo la Señalización de inicialización de Wink, cuando va el PBX offhook, el slot de tiempo lateral del router/del gateway en el estado ocioso (del EM\_ONHOOK) hasta que la llamada sea contestada por un destino remoto. Los cambios de estado del slot de tiempo del router a EM\_OFFHOOK cuando la llamada es contestada por el destino remoto.

Si la llamada no conecta, el router/el gateway juega los Tonos de reordenamiento inband al llamador. Puesto que el estado del canal en el lado del router sigue siendo EM\_ONHOOK, el router no puede colgar para arriba el canal. Después de que el llamador cuelgue para arriba, el PBX necesita cambiar a su estado del canal del offhook al onhook.

En algunos casos, los PBX no envían los mensajes ONHOOK, con la ayuda de las transiciones ABCD. El router tiene una solución alternativa para esto respuesta falsa llamada. Sin el método alternativo de respuestas falsas, los canales cuelgan en un estado del EM\_PARK indefinidamente. Vea la sección de la [respuesta falsa](#) para más información.

**Nota:** Las llamadas se pueden pegar en el estado del EM\_PARK en algunos canales T1 si el chasis del router de gateway de voz no se pone a tierra eléctricamente correctamente. Refiera al guía de instalación del hardware para más información sobre poner a tierra eléctrico.

## [Convenciones](#)

Consulte [Convenciones de Consejos TécnicosCisco](#) para obtener más información sobre las convenciones del documento.

## [Problema](#)

Hay dos posibles, los motivos principales que el slot de tiempo está pegado en el estado del EM\_PARK:

- El procesador de señales digitales (DSP) es malo y tiene cualquier problemas de hardware o de software.
- El switch/PBX PSTN envía una señal continua del offhook al router y no la libera.

## [Solución](#)

Éstas son las soluciones a este problema:

Si los slots de tiempo en su sistema se pegan en el estado del EM\_PARK, marque el DSPs. Refiera a [resolver problemas el DSP en el NM-HDV para los Cisco 2600/3600 Series Router](#) para marcar el DSPs.

Si el DSPs está vivo, el problema puede estar en el lado o el <sup>®</sup> del Cisco IOS (el router/el gateway del switch/PBX PSTN no comienza el procedimiento de la respuesta falsa). Vea la sección de la [respuesta falsa](#) para más información.

## [Respuesta falsa](#)

El router/gateway de Cisco espera un valor predeterminado de 30 segundos (utilice los [descansos espera-versión](#) y [comandos timeouts call-disconnect](#) para cambiar estos valores) después de que sepa que el slot de tiempo necesita ser fijado del PBX al onhook mientras que juega el Tono de reordenamiento.

Si no sucede esto, el router mueve el slot de tiempo al estado del EM\_PARK y comienza otro temporizador con una duración de 10 segundos. Si el PBX todavía no va onhook después de la duración de 10 segundos, el router engaña el PBX. El router envía una *respuesta falsa de una duración del segundo* y después va onhook.

Después de que el router envíe la señal de respuesta falsa, el router comienza otro temporizador de cinco minutos. Si va el PBX onhook, el temporizador para y las transiciones de router el slot de tiempo al estado del EM\_ONHOOK. Si no, después de cinco lo anota envía otra señal de respuesta falsa de a de un segundo de duración. El router relanza este proceso hasta que vaya el PBX onhook. El router fuerza el PBX para borrar la llamada.

**Nota:** Esta transición de contestación no se pone al día a ningún registro de contabilidad puesto que se borra la llamada real. Pero, el PBX la entiende mientras que cargan una respuesta y al usuario probablemente para de un segundo de duración la llamada.

Si el DSP asociado al slot de tiempo en el estado del EM\_PARK está vivo y sano y el problema persiste, funcione con el [vpm todo del debug](#) y los [comandos debug vtsp all](#) para ver si el Cisco IOS intenta enviar la respuesta falsa.

**Nota:** Usted necesita ejecutar los debugs por más de cinco minutos.

**Nota:** En la mayoría de los casos, si el DSP es malo, el router no realiza el método alternativo de respuestas falsas. Refiera a [resolver problemas el DSP en el NM-HDV para los Cisco 2600/3600 Series Router](#) para más información.

Esta salida de los debugs muestra cómo un slot de tiempo se pega en el EM\_PARK y cómo el método alternativo de respuestas falsas trabaja.

```
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=false, 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 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).*

## [Información Relacionada](#)

- [Soporte de tecnología de voz](#)
- [Soporte de Productos de Voice and Unified Communications](#)
- [Troubleshooting de Cisco IP Telephony](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)