

# Résolution des problèmes EM\_PARK pour signalisation CAS numérique E&M

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Théorie générale](#)

[Conventions](#)

[Problème](#)

[Solution](#)

[Fausse réponse](#)

[Informations connexes](#)

## Introduction

Dans la signalisation numérique E&M sur le Cisco 2600, 3600, et les Plateformes du routeur MC3810, les intervalles de temps un certain T1/E1 peuvent être bloqué dans l'état EM\_PARK. C'est visible quand vous émettez la commande de **résumé de show voice call**. Ce document explique comment dépanner cette question.

Cette sortie prouve que quelques intervalles de temps sont dans l'état EM\_PARK. Un intervalle de temps dans l'état EM\_PARK n'est pas utilisé pour des communications voix.

```
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
```

## Conditions préalables

### Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

### Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Matériel — Cisco 2600, Cisco 3600, Cisco VG200, et Routeurs MC3810
- Logiciel — Tous

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

## [Théorie générale](#)

Dans le T1 CAS, par exemple, le démarrage Wink signalant, quand le PBX disparaît l'offhook, l'intervalle de temps latéral de routeur/passerelle demeure dans l'état (EM\_ONHOOK) de veille jusqu'à ce que l'appel soit répondu par une destination distante. Les modifications d'état d'intervalle de temps de routeur à EM\_OFFHOOK quand l'appel est répondu par la destination distante.

Si l'appel ne se connecte pas, le routeur/passerelle lit l'intrabande commandent à nouveau des tonalités à l'appelant. Puisque l'état de canal du côté routeur est toujours EM\_ONHOOK, le routeur ne peut pas arrêter le canal. Après que l'appelant raccroche, le PBX doit changer son état de canal d'offhook à l'onhook.

Dans certains cas, les PBX n'envoient pas les messages d'onhook, avec l'aide des transitions ABCD. Le routeur a un contournement pour ceci réponse appelée de faux. Sans faux contournement de réponse, les canaux s'arrêtent dans un état d'EM\_PARK indéfiniment. Voyez le [faux](#) pour en savoir plus de section de [réponse](#).

**Remarque:** Des appels peuvent être coincés dans l'état EM\_PARK sur quelques canaux de t1 si le chassis de routeur de passerelle de Voix n'est pas électriquement fondu correctement. Référez-vous au guide d'installation du matériel pour plus d'informations sur fondre électrique.

## [Conventions](#)

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

## [Problème](#)

Il y a de deux possibles, des raisons importantes pour laquelle l'intervalle de temps est coincé dans l'état EM\_PARK :

- Le processeur de signaux numériques (DSP) est mauvais et a l'un ou l'autre de problèmes matériels ou logiciels.
- Le PSTN switch/PBX envoie un signal continu d'offhook au routeur et ne le libère pas.

## [Solution](#)

Ce sont les solutions au problème :

Si les intervalles de temps dans votre système sont coincés dans l'état EM\_PARK, vérifiez les DSP. Référez-vous à [dépanner le DSP sur NM-HDV pour des Routeurs de gamme Cisco](#)

[2600/3600](#) afin de vérifier les DSP.

Si les DSP sont actifs, le problème peut être sur le côté PSTN switch/PBX ou le Cisco IOS® (le routeur/passarelle ne commence pas la fausse procédure de réponse). Voyez le [faux](#) pour en savoir plus de section de [réponse](#).

## [Fausse réponse](#)

Le routeur/passarelle de Cisco attend une valeur par défaut de 30 secondes (employez les commandes de [timeouts wait-release](#) et de [timeouts call-disconnect](#) afin de changer ces valeurs) après qu'elle sache que l'intervalle de temps doit être placé du PBX à l'onhook tandis qu'il joue la tonalité de réarrangement.

Si ceci ne se produit pas, le routeur déplace l'intervalle de temps à l'état EM\_PARK et met en marche un autre temporisateur avec une durée de 10 secondes. Si le PBX ne va toujours pas l'onhook après la durée de 10 secondes, le routeur dupe le PBX. Le routeur envoie une *fausse réponse d'une* durée d'une seconde et puis va l'onhook.

Après que le routeur envoie le faux signal de réponse, le routeur met en marche un autre temporisateur de cinq minutes. Si le PBX disparaît l'onhook, le temporisateur arrête et les transitions de routeur l'intervalle de temps à l'état EM\_ONHOOK. Autrement, après cinq minute l'envoi un autre faux signal de réponse d'une un-deuxième durée. Le routeur répète ce processus jusqu'à ce que le PBX disparaisse l'onhook. Le routeur force le PBX pour effacer l'appel.

**Remarque:** Cette transition de réponse n'est pas mise à jour à aucun enregistrement des comptes puisque l'appel réel est effacé. Mais, le PBX le comprend pendant qu'une réponse et l'utilisateur est probablement facturée l'un-deuxième appel de durée.

Si le DSP associé avec l'intervalle de temps dans l'état EM\_PARK est actif et sain et le problème persiste, exécutez le [debug vpm all](#) et les commandes [debug vtsp all](#) afin de voir si des tentatives de Cisco IOS d'envoyer la fausse réponse.

**Remarque:** Vous devez exécuter met au point pendant plus de cinq minutes.

**Remarque:** Dans la plupart des cas, si le DSP est mauvais, le routeur n'exécute pas le faux contournement de réponse. Référez-vous à [dépanner le DSP sur NM-HDV pour le](#) pour en savoir plus de [Routeurs de gamme Cisco 2600/3600](#).

Cette sortie de débogage affiche comment un intervalle de temps devient collé dans EM\_PARK et comment le faux contournement de réponse fonctionne.

```
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 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).*

## [Informations connexes](#)

- [Assistance technique concernant la technologie vocale](#)
- [Assistance concernant les produits vocaux et de communications unifiées](#)
- [Dépannage des problèmes de téléphonie IP Cisco](#)
- [Support et documentation techniques - Cisco Systems](#)