لاسرإل احالصإو EM_PARK ءاطخأ فاشكتسأ إشارات CAS الرقمي ل E&M

المحتويات

المقدمة
المتطلبات الأساسية
المتطلبات
المكونات المستخدمة
النظرية الأساسية
الاصطلاحات
المشكلة
الحل
جواب زائف
معلومات ذات صلة

المقدمة

في إرسال إشارات E&M الرقمية على الأنظمة الأساسية لموجهات Cisco 2600 و 3600 و MC3810، يمكن أن تصبح بعض الفتحات الزمنية T1/E1 عالقة في حالة EM_PARK. ويكون هذا مرئيا عند إصدار الأمر **show voice call summary**. يشرح هذا المستند كيفية أستكشاف أخطاء هذه المشكلة وإصلاحها.

يوضح هذا الإخراج أن بعض فتحات الوقت موجودة في حالة EM_PARK. لا يتم إستخدام فتحة زمنية في حالة EM_PARK للمكالمات الصوتية.

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

المتطلبات الأساسية

المتطلبات

لا توجد متطلبات خاصة لهذا المستند.

المكونات المستخدمة

تستند المعلومات الواردة في هذا المستند إلى إصدارات البرامج والمكونات المادية التالية:

- الموجهات Cisco 2600 و Cisco 3600 و Cisco VG200 و MC3810
- البرامج - الكل

تم إنشاء المعلومات الواردة في هذا المستند من الأجهزة الموجودة في بيئة معملية خاصة. بدأت جميع الأجهزة المستخدمة في هذا المستند بتكوين ممسوح (افتراضي). إذا كانت شبكتك مباشرة، فتأكد من فهمك للتأثير المحتمل لأي أمر.

النظرية الأساسية

في T1 CAS، على سبيل المثال، إرسال إشارات بدء التشغيل الخاملة، عندما يذهب PBX بعيدا عن المسار، يبقى الفتحة الزمنية للموجه/البوابة في حالة الخمول (EM_ONHOOK) حتى يتم الرد على المكالمة بواسطة وجهة بعيدة. تتغير حالة فتحة وقت الموجه إلى EM_OFFHOOK عندما يتم الرد على المكالمة بواسطة الوجهة البعيدة.

إذا لم يتم توصيل المكالمة، سيقوم الموجه/البوابة بتشغيل نغمات إعادة الترتيب داخل النطاق للمتصل. بما أن حالة القناة على جانب الموجه لا تزال EM_ONHOOK، فإن الموجه غير قادر على تعليق القناة. بعد توقف المستدعي، يحتاج PBX إلى تغيير حالة القناة الخاصة به من "خارج" إلى "خارج".

في بعض الحالات، لا ترسل PBX رسائل onhook، مع مساعدة من ال PBX انتقالات. يوجد لدى الموجه حل بديل لهذه الإجابة المزيفة. و بدون إستخدام الإجابة المزيفة، ستبقى القنوات في حالة من عدم الاستقرار. راجع قسم الإجابات المزيفة للحصول على مزيد من المعلومات.

ملاحظة: يمكن تثبيت المكالمات في حالة EM_PARK على بعض قنوات T1 إذا لم يتم إنشاء هيكل موجه العبارة الصوتية بشكل صحيح. ارجع إلى دليل تثبيت الأجهزة للحصول على مزيد من المعلومات حول الأساس الكهربائي.

الاصطلاحات

راجع اصطلاحات تلميحات Cisco التقنية للحصول على مزيد من المعلومات حول اصطلاحات المستندات.

المشكلة

هناك سببان محتملان و رئيسيان أن الفترة الزمنية لا تزال عالقة في حالة EM_PARK:

- معالج الإشارة الرقمية (DSP) سيئ ويعاني من مشكلات في الأجهزة أو البرامج.
- يرسل محول PSTN/PBX إشارة تدفق مستمرة إلى الموجه ولا يطلقها.

الحل

هذه هي الحلول لهذه المشكلة:

إذا كانت الفتحات الزمنية في النظام الخاص بك عالقة في حالة EM_PARK، فتحقق من مزودي الخدمة المعتمدين من Dell (المعروفين باسم DSP). ارجع إلى أستكشاف أخطاء DSP وإصلاحها على NM-HDV لموجهات سلسلة 3600/2600 من Cisco للتحقق من DSP.

إذا كانت وحدات DSP على قيد الحياة، فقد تكون المشكلة على محول PSTN/PBX من جانب أو أن Cisco IOS® (لا يقوم الموجه/البوابة بتشغيل إجراء الإجابة المزيفة). راجع قسم الإجابات المزيفة للحصول على مزيد من المعلومات.

جواب زائف

ينتظر موجه/عبارة Cisco قيمة افتراضية من 30 ثانية (أستخدم أوامر **حالات انتهاء المهلة** و**حالات انتهاء المهلة** لتغيير

هذه القيم) بعد أن يعلم أنه يجب تعيين فتحة الوقت من PBX إلى onhook أثناء تشغيله لطريقة إعادة الترتيب.

إذا لم يحدث ذلك، يقوم الموجه بنقل فتحة الوقت إلى حالة EM_PARK ويقوم بتشغيل مؤقت آخر لمدة 10 ثوان. إذا كان PBX لا يزال لا يعمل بعد المدة 10 ثانية، فيخدع الموجه PBX. يرسل الموجه *إجابة وهمية* لمدة ثانية واحدة ثم ينتقل إلى.

بعد أن يرسل الموجه إشارة الإجابة المزيفة، يبدأ الموجه في تشغيل مؤقت آخر مدته خمس دقائق. إذا تم تشغيل PBX، يتوقف المؤقت ويقوم الموجه بنقل فتحة الوقت إلى حالة EM_ONHOOK. وإلا، بعد خمس دقائق ترسل إشارة إجابة وهمية أخرى لمدة ثانية واحدة. يقوم الموجه بتكرار هذه العملية حتى تنتقل PBX إلى مكان آخر. يفرض الموجه على PBX مسح المكالمة.

ملاحظة: لم يتم تحديث انتقال الإجابة هذا إلى أي سجل محاسبي نظرا لأنه تم مسح المكالمة الفعلية. ولكن ال PBX يفهمها على أنها إجابة والمستخدم على الأرجح سيتحمل تكاليف مكالمة ثانية واحدة.

إذا كان DSP المقترن بفتحة الوقت في حالة EM_PARK حيا وصحيا واستمرت المشكلة، فقم بتشغيل **الأمر debug** vpm all و**debug vtsp all** لمعرفة ما إذا كان Cisco IOS يحاول إرسال الإجابة المزيفة.

ملاحظة: تحتاج إلى تشغيل عمليات تصحيح الأخطاء لأكثر من خمس دقائق.

ملاحظة: في معظم الحالات، إذا كان DSP غير صالح، لا يقوم الموجه بتنفيذ حل الإجابة المزيف. راجع أستكشاف أخطاء DSP وإصلاحها على NM-HDV لموجهات سلسلة 3600/2600 من Cisco للحصول على مزيد من المعلومات.

يوضح إخراج تصحيح الأخطاء هذا كيفية تعلق فتحة الوقت في em_park وكيفية عمل الحل البديل للإجابة المزيفة.

```
Jan 11 17:19:00.767: htsp_dsp_message: SEND/RESP_SIG_STATUS: state=0xC timestamp
                                                        systime=31305235 44262=
                                                :Jan 11 17:19:00.767: htsp_process_event
                                        EM_ONHOOK, E_DSP_SIG_1100]em_onhook_offhook htsp_setup_ind ,(10)4/1:1]
                                    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 p acket_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)csize(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 state = 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=0xFFFFFFF 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)csize(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)csize(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=0xFFFFFFF 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)csize(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)csize(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=0xFFFFFFF 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)csize(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)csize(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 amp_of_second=2928 direction=1 on_time_first=250 off_time_first=250 5206**

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

معلومات ذات صلة

- دعم تقنية الصوت
- دعم منتجات الاتصالات الصوتية والاتصالات الموحّدة
- استكشاف أخطاء خدمة IP الهاتفية من Cisco وإصلاحها
- الدعم التقني والمستندات - Cisco Systems

حول هذه الترجمة
ترجمت Cisco هذا المستند باستخدام مجموعة من التقنيات الآلية والبشرية لتقديم محتوى دعم للمستخدمين في جميع أنحاء العالم بلغتهم الخاصة. يُرجى ملاحظة أن أفضل ترجمة آلية لن تكون دقيقة كما هو الحال مع الترجمة الاحترافية التي يقدمها مترجم محترف. تخلي Cisco Systems مسئوليتها عن دقة هذه الترجمات وتُوصي بالرجوع دائمًا إلى المستند الإنجليزي الأصلي (الرابط متوفر).