



Основы устранения неполадок и отладки вызовов по протоколу VoIP

Содержание

Введение

Предварительные условия

- Требования
- Используемые компоненты
- Условные обозначения

Поток вызовов в сети

Поток вызова маршрутизатора

Архитектура интерфейса IP-телефонии

Проверка цифровой и аналоговой сигнализации (ветвь вызова POTS)

- show controllers T1 / E1 (digital)
- show voice port
- debug vpm (voice processor module)

Проверка полученных и отправленных цифр (по ветвям вызовов POTS)

- show dialplan number
- debug vtsp session

Проверка сквозной передачи сигналов VoIP (участок вызова VoIP)

- debug voip ccapi inout

Сведения об ошибках качества обслуживания (QoS) VoIP

Данные кодов причины и значений отладки для VoIP

- Q.931 Причины отключения вызова (cause_codes из debug voip ccapi inout)
- Значения согласования кодека (из debug voip ccapi inout)
- Типы сигналов
- Значения возможностей FAX-Rate и VAD

Дополнительные сведения

Введение

В этом документе демонстрируются основные приемы и команды для устранения неполадок и отладки сетей VoIP. Представлен общий обзор потока голосового вызова и архитектуры телефонии в маршрутизаторе Cisco, а также пошаговый подход к устранению неполадок Cisco, описанный в нескольких пунктах:

1. Проверка цифровой и аналоговой сигнализации.
2. Проверка цифр, полученных и отправленных с аналоговых и цифровых голосовых портов.
3. Проверка сквозной передачи сигналов VoIP.
4. Сведения об ошибках качества обслуживания (QoS) VoIP.
5. Данные кодов причины и значений отладки для VoIP.

Примечание: в этом документе не объясняются все аспекты архитектуры Cisco IOS®, используемые в шлюзах и привратниках Cisco VoIP. Он предназначен для объяснения того, какие команды могут использоваться и какие поля из выходных данных команд могут иметь наибольшее значение.



Внимание: отладка Cisco IOS может вызвать сильную нагрузку на процессор. Будьте внимательны при использовании процедур отладки, приведенных в этом документе. Для получения дополнительных сведений обратитесь к разделу Важные сведения о командах

отладки.

Процедуры отладки необходимо запускать со включенными в журнале метками времени. Включите установку меток времени, добавив следующие команды: **service timestamps debug datetime msec**, **service timestamps log datetime msec** в режиме enable. Метки времени помогают определить интервал времени между изменениями состояния.

Предварительные условия

Требования

Этот документ предназначен для сетевых специалистов, участвующих в проектировании и внедрении сетей VoIP. Читатели этого документа должны знать следующее:

- настройку VoIP
- Voice QoS

Используемые компоненты

Данный документ не ограничен отдельными версиями программного и аппаратного обеспечения. Приводимые выходные данные получены на программном обеспечении Cisco IOS®, релиз 12.3(8).

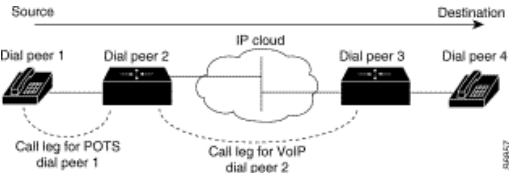
Сведения, представленные в данном документе, были получены на тестовом оборудовании в специально созданных лабораторных условиях. При написании данного документа использовались только данные, полученные от устройств с конфигурацией по умолчанию. При работе с реально функционирующей сетью необходимо полностью осознавать возможные последствия выполнения команд до их применения.

Условные обозначения

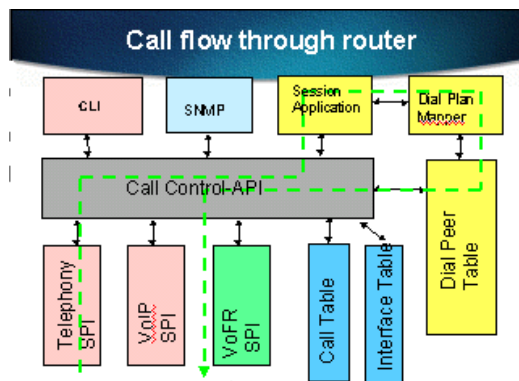
Дополнительные сведения об условных обозначениях в документах см. в разделе Технические советы Cisco. Условные обозначения.

Поток вызовов в сети

Важным фактором, который надо учитывать до начала любого устранения неполадок VoIP или отладки, является то, что соединения VoIP состоят из трех участков. Этими тремя участками соединения являются исходная обычная телефонная сеть (POTS), VoIP и POTS назначения. Это показано на следующем рисунке. Устранение неполадок и отладка должны быть нацелены сначала отдельно на каждый участок, а затем на все соединение VoIP в целом.



Поток вызова маршрутизатора



В данных определениях объясняются функции главных компонентов, показанных на диаграмме потока вызова маршрутизатора:

API-интерфейс контроля вызовов — API-интерфейс контроля вызовов используется тремя клиентами. Это интерфейс командной строки, агент SNMP и приложение сеанса. Главными функциями API-интерфейса контроля вызовов (CCAPI) являются:

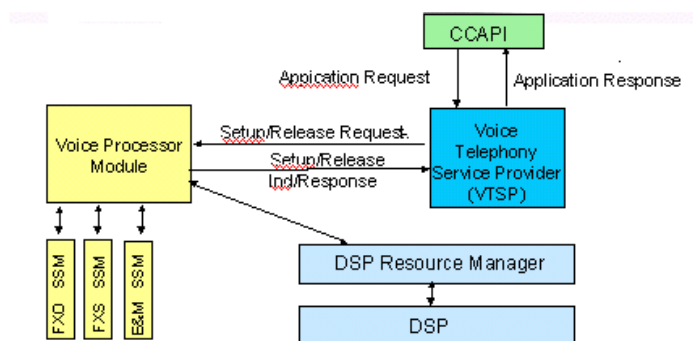
- Идентификация участков вызова (например, какая это точка вызова, откуда это соединение).
- Принятие решения о том, какое приложение сеанса примет вызов (например, кто его будет обрабатывать).
- Вызов обработчика пакетов.
- Общая обработка всех ветвей вызова.
- Начало записи статистики вызовов.

Приложение сеанса и программа сопоставления плана соединений — Приложение сеанса использует программу сопоставления плана соединений для сопоставления номера точки вызова (местной POTS или удаленной VoIP). Программа сопоставления плана соединений использует таблицу одноранговых узлов для поиска активных адресуемых точек вызова.

SPI-интерфейс телефонии и VoIP — Компонент SPI телефонии соединяется с точками вызова POTS (аналоговыми: fxs, fxo, e&m, цифровыми: isdn, qsig, e&m и т.д.). SPI VoIP - особый интерфейс для одноранговых узлов VoIP. Драйверы телефонии/DSP обеспечивают доставку услуг к SPI телефонии, в то время как SPI VoIP опирается на протоколы сеанса.

Архитектура интерфейса IP-телефонии

На этой схеме показана архитектура компоновочных блоков телефонии маршрутизатора Cisco и их взаимодействие друг с другом.



Далее описываются функции и даются определения главных компонентов схемы:

- **API-интерфейс контроля вызовов (CCAPI)** — Элемент программного обеспечения, который устанавливает, завершает и связывает ветви вызова.
- **Поставщик услуг голосовой телефонии (VTSP)** — Процесс IOS, который обслуживает запросы из API-интерфейса контроля вызовов и формулирует соответствующий запросы к процессору цифровых сигналов (DSP) или к VPM.

- **Модуль голосового процессора (VPM)** — VPM ответственен за объединение и координирование процессов передачи сигнала между SSM портов телефонии, менеджером ресурсов DSP и VTSP.
- **DSP Resource Manager** — DSPRM предоставляет интерфейсы, с помощью которых VTSP может обмениваться сообщениями с DSP.
- **Обработчик пакетов** — Обработчик пакетов пересылает пакеты между DSP и ветвями одноранговых соединений.
- **Партнер по вызову** — партнер по вызову - это противоположная ветвь вызова. Это может быть другое телефонное голосовое соединение (POTS), соединение VoFR, VoATM или VoIP.

Проверьте цифровую и аналоговую сигнализацию (ветвь вызова POTS)

Целью проверки цифровой и аналоговой сигнализации является:

- Определить, что принимаются правильные аналоговые или цифровые сигналы подключения и отключения.
- Убедиться, что на маршрутизаторе и коммутаторе (центральная или офисная АТС) правильно настроена сигнализация E&M, FXO и FXS.
- Убедиться, что DSP находятся в режиме сбора цифровых данных.

Подчеркнутые команды из этого раздела могут использоваться для проверки сигнализации.

show controllers T1 / E1 (digital)

show controllers t1 [slot/port] — используйте эту команду первой. Она показывает, установлено или нет цифровое соединение T1 между маршрутизатором и коммутатором (центральная или офисная АТС), и правильно ли оно работает. Выходные данные этой команды выглядят следующим образом:

```
router# show controllers T1 1/0
T1 1/0 is up.
Applique type is Channelized T1
Cablelength is short 133
No alarms detected.
Framing is ESF, Line Code is B8ZS, Clock Source is Line
Primary.
Data in current interval (6 seconds elapsed):

0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs,
 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs,
 0 Unavail Secs
```

При использовании E1 употребляйте команду **show controllers e1**. Подробнее см.

- Устранение неполадок уровня 1 T1
- Блок-схема устранения неполадок T1
- Устранение неполадок, связанных с проблемами последовательного канала

show voice port

show voice port slot-number/port — используйте эту команду для отображения состояния порта и параметров, настроенных на голосовом порте карты речевого интерфейса (VIC) Cisco. Как и во всех командах IOS, параметры по умолчанию не отображаются в

выходных данных **show running-config**, но отображаются с помощью этой команды.

Это пример выходных данных для голосового порта E&M:

```
router# show voice port 1/0:1
recEive and transMit Slot is 1, Sub-unit is 0, Port is 1
Type of VoicePort is E&M
Operation State is DORMANT
Administrative State is UP
No Interface Down Failure
Description is not set
Noise Regeneration is enabled
Non Linear Processing is enabled
Music On Hold Threshold is Set to -38 dBm
In Gain is Set to 0 dB
Out Attenuation is Set to 0 dB
Echo Cancellation is enabled
Echo Cancel Coverage is set to 16 ms
Connection Mode is normal
Connection Number is not set
Initial Time Out is set to 10 s
Interdigit Time Out is set to 10 s
Call-Disconnect Time Out is set to 60 s
Region Tone is set for US

Voice card specific Info Follows:
Out Attenuation is Set to 0 dB
Echo Cancellation is enabled
Echo Cancel Coverage is set to 16 ms
Connection Mode is normal (could be trunk or plar)
Connection Number is not set
Initial Time Out is set to 10 s
Interdigit Time Out is set to 10 s
Call-Disconnect Time Out is set to 60 s
Region Tone is set for US

Voice card specific Info Follows:
Signal Type is wink-start
Operation Type is 2-wire
E&M Type is 1
Dial Type is dtmf
In Seizure is inactive
Out Seizure is inactive
Digit Duration Timing is set to 100 ms

InterDigit Duration Timing is set to 100 ms
Pulse Rate Timing is set to 10 pulses/second
InterDigit Pulse Duration Timing is set to 500 ms
Clear Wait Duration Timing is set to 400 ms
Wink Wait Duration Timing is set to 200 ms
Wink Duration Timing is set to 200 ms
Delay Start Timing is set to 300 ms
Delay Duration Timing is set to 2000 ms
Dial Pulse Min. Delay is set to 140 ms
```

debug vpm (voice processor module)

Эти команды используются для отладки интерфейса телефонии VPM:

- **debug vpm signal** — эта команда используется для сбора информации отладки для событий сигнализации и может быть полезна при разрешении проблем с сигнализацией в PBX.
- **debug vpm spi** — эта команда отслеживает, как SPI модуля голосового порта взаимодействует с API-интерфейсом контроля вызовов. Команда отладки **debug** выводит на экран сведения об обработке запросов индикации и приложений сети.
- **debug vpm dsp** — эта команда показывает сообщения из DSP на VPM к маршрутизатору и может быть полезна, если есть подозрение, что VPM не работает. Это простой способ проверить, реагирует ли VPM на индикаторы отключения и оценить промежутки между поступлением сообщений сигнализации из интерфейса.
- **debug vpm all** — эта команда EXEC включает все команды debug vpm: **debug vpm spi**, **debug vpm signal** и **debug vpm dsp**.
- **debug vpm port** — используйте эту команду для вывода данных отладки только для определенного порта. Например, в данном

примере показан вывод сообщений **debug vpm dsp** только для порта 1/0/0:

```
debug vpm dsp  
  
debug vpm port 1/0/0
```

Подробнее см. Команды отладки VoIP.

Пример выходных данных для команды **debug vpm signal**

```
maui-voip-austin#debug vpm signal  
  
!--- FXS порт 1/0/0 переходит от состояния подключения к состоянию  
!--- отключения.  
  
htsp_process_event: [1/0/0, 1.2 , 36]  
fxsls_onhook_offhook htsp_setup_ind  
*Mar 10 16:08:55.958: htsp_process_event:  
[1/0/0, 1.3 , 8]  
  
!--- Посылает сигнал вызова на вызываемый телефон.  
  
*Mar 10 16:09:020,410: htsp_process_event:  
[1/0/0, 1.3 , 10] htsp_alert_notify  
*Mar 10 16:09:030,378: htsp_process_event:  
[1/0/0, 1.3 , 11]  
  
!--- Завершает звонок, порт переходит в состояние подключения.  
  
*Mar 10 16:09:110,966: htsp_process_event:  
[1/0/0, 1.3 , 6]  
*Mar 10 16:09:170,218: htsp_process_event:  
[1/0/0, 1.3 , 28] fxsls_offhook_onhook  
*Mar 10 16:09:170,370: htsp_process_event:  
[1/0/0, 1.3 , 41] fxsls_offhook_timer  
*Mar 10 16:09:170,382: htsp_process_event:  
[1/0/0, 1.2 , 7] fxsls_onhook_release
```

Если в состоянии подключения и отключения сигнализация происходит неправильно, проверьте следующее:

- Проверьте подключение кабеля.
- Проверьте заземление маршрутизатора и коммутатора (центральная или офисная АТС).
- Убедитесь в том, что конечные точки соединения имеют соответствующие настройки сигнализации. Несоответствующие конфигурации могут вызывать неполную или одностороннюю сигнализацию.

Подробнее об устранении неполадок E&M см. Общие сведения и устранение неполадок аналоговых интерфейсов E & M и схем проводки.

Пример выходных данных для команды **debug vpm spi**

```
maui-voip-austin#debug vpm spi  
Voice Port Module Session debugging is enabled  
  
!--- DSP переведен в режим сбора цифр.  
  
*Mar 10 16:48:55.710: dsp_digit_collect_on:  
[1/0/0] packet_len=20 channel_id=128  
packet_id=35 min_inter_delay=290  
max_inter_delay=3200 mim_make_time=18 max_make
```

```
_time=75 min_brake_time=18 max_brake_time=75
```

Проверка полученных и отправленных цифр (по ветвям вызовов POTS)

После того как сигнализация подключения и отключения проверена и работает правильно, убедитесь в том, что на голосовой порт (цифровой или аналоговый) принимаются и отправляются правильные цифры. Если принимаются или отправляются неправильные цифры, точка вызова может не согласовываться, а коммутатор (центральная или офисная АТС) может не соединяться с нужной станцией. Некоторые команды, которые можно использовать для проверки полученных/отправленных цифр:

- **show dialplan number** — эта команда используется для отображения того, какая точка вызова была достигнута при наборе конкретного телефонного номера.
- **debug vtsp session** — эта команда показывает данные об обработке запросов индикации и приложений сети, об индикации сигнализации и контрольных сообщениях DSP.
- **debug vtsp dsp** — в программном обеспечении Cisco IOS ранее релиза 12.3 эта команда показывает цифры в том виде, в котором они приняты голосовым портом. Однако в программном обеспечении Cisco IOS, релиз 12.3 и более поздние выходные данные команды **debug** больше не показывают цифры. Для просмотра входящих цифр можно использовать комбинацию команд **debug hpi detail** и **debug hpi notification**.
- **debug vtsp all** — эта команда включает следующие команды VTSP: **debug vtsp session**, **debug vtsp error** и **debug vtsp dsp**.

Подробнее см. Команды отладки VoIP.

show dialplan number

show dialplan number <digit string> — эта команда показывает точку вызова, которая сопоставлена строкой цифр. Если могут быть сопоставлены несколько точек вызова, они показываются все, в том порядке, в котором были сопоставлены.

Примечание: чтобы провести сопоставление по шаблонам назначения, оканчивающимся на Т, необходимо использовать знак # в конце телефонных номеров для точек вызова с переменной длиной.

Выходные данные этой команды выглядят следующим образом:

```
maui-voip-austin#show dialplan number 5000
Dial string terminator: #
Macro Exp.: 5000

VoiceOverIpPeer2
  information type = voice,
  tag = 2, destination-pattern = `5000',
  answer-address = `', preference=0,
  group = 2, Admin state is up, Operation
  state is up,
  incoming called-number = `',
  connections/maximum = 0/unlimited,
  application associated:
  type = voip, session-target =
  `ipv4:192.168.10.2',
  technology prefix:
  ip precedence = 5, UDP checksum =
  disabled, session-protocol = cisco,
  req-qos = best-effort,
  acc-qos = best-effort,
  dtmf-relay = cisco-rtp,
  fax-rate = voice,
  payload size = 20 bytes
  codec = g729r8,
  payload size = 20 bytes,
  Expect factor = 10, Icpif = 30,
  signaling-type = cas,
  VAD = enabled, Poor QOV Trap = disabled,
  Connect Time = 25630, Charged Units = 0,
```

```
Successful Calls = 25, Failed Calls = 0,  
Accepted Calls = 25, Refused Calls = 0,  
Last Disconnect Cause is "10 ",  
Last Disconnect Text is "normal call  
clearing.",  
Last Setup Time = 84427934.  
Matched: 5000 Digits: 4  
Target: ipv4:192.168.10.2
```

debug vtsp session

Команда **debug vtsp session** показывает сведения о том, как маршрутизатор взаимодействует с DSP, на основании индикаторов сигнализации из стека сигнализации и запросов из приложения. Эта команда **debug** показывает данные о том, как обрабатываются каждый индикатор сети и запрос приложения, об индикаторах сигнализации и контрольных сообщения DSP.

```
maui-voip-austin#debug vtsp session  
Voice telephony call control session debugging is on  
  
!--- Некоторые выходные данные опущены.  
!--- Действие: Звонящий поднял трубку.  
!--- Определены DSP, буферы дрожания,  
!--- пороги VAD и установлены уровни сигнала.  
  
*Mar 10 18:14:22.865: dsp_set_playout: [1/0/0 (69)]  
packet_len=18 channel_id=1 packet_id=76 mode=1  
initial=60 min=4 max=200 fax_nom=300  
*Mar 10 18:14:22.865: dsp_echo_canceller_control:  
[1/0/0 (69)] packet_len=10 channel_id=1 packet_id=66  
flags=0x0  
*Mar 10 18:14:22.865: dsp_set_gains: [1/0/0 (69)]  
packet_len=12 channel_id=1 packet_id=91  
in_gain=0 out_gain=65506  
*Mar 10 18:14:22.865: dsp_vad_enable: [1/0/0 (69)]  
packet_len=10 channel_id=1 packet_id=78  
thresh=-38act_setup_ind_ack  
*Mar 10 18:14:22.869: dsp_voice_mode: [1/0/0 (69)]  
packet_len=24 channel_id=1 packet_id=73 coding_type=1  
voice_field_size=80  
VAD_flag=0 echo_length=64 comfort_noise=1  
inband_detect=1 digit_relay=2  
AGC_flag=0act_setup_ind_ack(): dsp_dtmf_mod  
e()act_setup_ind_ack: passthru_mode = 0,  
no_auto_switchover = 0dsp_dtmf_mode  
(VTSP_TONE_DTMF_MODE)  
  
!--- DSP переведен в "голосовой режим", и сгенерирован  
!--- тональный сигнал.  
  
*Mar 10 18:14:22.873: dsp_cp_tone_on: [1/0/0 (69)]  
packet_len=30 channel_id=1 packet_id=72 tone_id=4  
n_freq=2 freq_of_first=350 freq_of_second=440 amp_of_first=  
4000 amp_of_second=4000 direction=1 on_time_first=65535  
off_time_first=0 on_time  
_second=65535 off_time_second=0
```

Если определено, что цифры были отправлены или приняты неправильно, возможно, будет необходимо использовать сборщик цифр (средство проверки) или тестер T1, которые проверяют, отправляются ли цифры с правильной частотой и интервалами времени. Если они "неправильно" отправляются для коммутатора (центральная или офисная АТС), некоторые значения маршрутизатора или коммутатора (центральная или офисная АТС), возможно, потребуются скорректировать, чтобы они согласовывались и были совместимы. Обычно это значения длительности цифры и интервала между цифрами. Необходимо также проверить, не являются ли цифры, которые отправляются как будто правильно, какими-либо таблицами трансляции номеров в коммутаторе (центральная или офисная АТС), которые могут добавлять или удалять цифры.

Проверка сквозной передачи сигналов VoIP (участок вызова VoIP)

После того, как вы убедитесь, что сигнализация голосового порта работает правильно и принимаются правильные цифры, переходите к устранению неполадок и отладке управления вызовами VoIP. Следующие факторы объясняют, почему отладка управления вызовами может оказаться сложной работой.

- Шлюзы Cisco VoIP используют для выполнения вызовов сигнализацию H.323. H.323 состоит из трех уровней согласования вызовов и установления вызовов: H.225, H.245 и H.323. Эти протоколы используют сочетание TCP и UDP для установления и организации вызова.
- Отладка сквозного VoIP показывает число конечных автоматов IOS. Проблемы с любым из конечных автоматов могут вызвать обрыв соединения.
- Отладка сквозного VoIP может оказаться очень многословной и произвести большой объем выходных данных отладки.

debug voip ccapi inout

Основной командой для отладки сквозных вызовов VoIP является **debug voip ccapi inout**. В данном примере показаны выходные данные отладки вызова.

```
!--- Действие: Вызов VoIP инициирован через
!--- SPI телефонии (ветвь pots) на расширение 5000.
!--- Некоторые выходные данные опущены.

maui-voip-austin#debug voip ccapi inout
voip ccAPI function enter/exit debugging is on

!--- Идентификация ветви вызова, узел источника: Вызов
!--- исходит из pots, точка вызова 1.
!--- (расширение 4000).

*Mar 15 22:07:11.959: cc_api_call_setup_ind
(vdbPtr=0x81B09EFC, callInfo={called=,
calling=4000, fdest=0 peer_tag=1}, callID=0x81B628F0)

!--- CCAPI вызывает приложение сеанса.

*Mar 15 22:07:11.963: cc_process_call_setup_ind
(event=0x81B67E44) handed call to app "SESSION"
*Mar 15 22:07:11.963: sess_appl:
ev(23=CC_EV_CALL_SETUP_IND), cid(88), disp(0)

!--- Определите местонахождение идентификаторов ветви вызова "callid = 0x59"

*Mar 15 22:07:11.963: ccCallSetContext
(callID=0x58, context=0x81BAF154)
*Mar 15 22:07:11.963: ccCallSetupAck
(callID=0x58)

!--- Используйте VTSP для генерации тонового сигнала
.

*Mar 15 22:07:11.963: ccGenerateTone
(callID=0x58 tone=8)

!--- VTSP передает цифры в CCAPI.

*Mar 15 22:07:20.275:cc_api_call_digit_begin
(vdbPtr=0x81B09EFC,callID=0x58,digit=5, flags=0x1, timestamp=0xC2E63BB7, expiration=0x0)
*Mar 15 22:07:20.279: sess_appl:
ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(88), disp(0)
*Mar 15 22:07:20.279: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0)
*Mar 15 22:07:20.279: ssaIgnore cid(88),
st(0),oldst(0), ev(10)
*Mar 15 22:07:20.327: cc_api_call_digit
(vdbPtr=0x81B09EFC, callID=0x58, digit=5
, duration=100)
```

```
*Mar 15 22:07:20.327: sess_appl:
ev(9=CC_EV_CALL_DIGIT), cid(88), disp(0)
*Mar 15 22:07:20.327: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csiz(0)in(1)fDes
t(0)
*Mar 15 22:07:21.975:cc_api_call_digit_begin
(vdbPtr=0x81B09EFC,callID=0x58,digit=0,
flags=0x1, timestamp=0xC2E63BB7, expiration=0x0)
*Mar 15 22:07:21.979: sess_appl:
ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(88), disp(0)
*Mar 15 22:07:21.979: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csiz(0)in(1)fDes
t(0)
*Mar 15 22:07:21.979: ssaIgnore cid(88),
st(0),oldst(0), ev(10)
*Mar 15 22:07:22.075: cc_api_call_digit
(vdbPtr=0x81B09EFC, callID=0x58, digit=0
, duration=150)
*Mar 15 22:07:22.079: sess_appl:
ev(9=CC_EV_CALL_DIGIT), cid(88), disp(0)
*Mar 15 22:07:22.079: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csiz(0)in(1)fDest(0)
*Mar 15 22:07:23.235: cc_api_call_digit_begin
(vdbPtr=0x81B09EFC, callID=0x58, dgit=0,
flags=0x1, timestamp=0xC2E63BB7, expiration=0x0)
*Mar 15 22:07:23.239: sess_appl:
ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(88), disp(0)
*Mar 15 22:07:23.239: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csiz(0)in(1)fDest(0)
*Mar 15 22:07:23.239: ssaIgnore cid(88),
st(0),oldst(0), ev(10)
*Mar 15 22:07:23.335: cc_api_call_digit
(vdbPtr=0x81B09EFC, callID=0x58, digit=0
, duration=150)
*Mar 15 22:07:23.339: sess_appl:
ev(9=CC_EV_CALL_DIGIT), cid(88), disp(0)
*Mar 15 22:07:23.339: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csiz(0)in(1)fDes
t(0)
*Mar 15 22:07:25.147: cc_api_call_digit_begin
(vdbPtr=0x81B09EFC, callID=0x58, d
igit=0, flags=0x1, timestamp=0xC2E63BB7,
expiration=0x0)
*Mar 15 22:07:25.147: sess_appl:
ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(88), disp(0)
*Mar 15 22:07:25.147: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csiz(0)in(1)fDest(0)
*Mar 15 22:07:25.147: ssaIgnore cid(88),
st(0),oldst(0), ev(10)
*Mar 15 22:07:25.255: cc_api_call_digit
(vdbPtr=0x81B09EFC, callID=0x58, digit=0
, duration=160)
*Mar 15 22:07:25.259: sess_appl:
ev(9=CC_EV_CALL_DIGIT), cid(88), disp(0)
*Mar 15 22:07:25.259: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csiz(0)in(1)fDest(0)

!--- Сопоставляется точка вызова voip 2 . Номер назначения
!--- 5000

*Mar 15 22:07:25.259: ssaSetupPeer cid(88)
peer list:tag(2) called number(5000)
*Mar 15 22:07:25.259: ssaSetupPeer cid(88),
destPat(5000), matched(4), prefix(),
peer(81C04A10)

!--- Продолжайте вызывать интерфейс и начинайте
!--- следующий участок вызова.

*Mar 15 22:07:25.259: ccCallProceeding
(callID=0x58, prog_ind=0x0)
*Mar 15 22:07:25.259: ccCallSetupRequest
(Inbound call = 0x58, outbound peer =2,
dest=, params=0x81BAF168 mode=0,
*callID=0x81B6DE58)
*Mar 15 22:07:25.259: callingNumber=4000,
calledNumber=5000, redirectNumber=

!--- Установка вызова VoIP.
```

```
*Mar 15 22:07:25.263: ccIFCallSetupRequest:
(vdbPtr=0x81A75558, dest=,
callParams={called=5000, calling=4000,
fdest=0, voice_peer_tag=2}, mode=0x0)
*Mar 15 22:07:25.263: ccCallSetContext
(callID=0x59, context=0x81BAF3E4)
*Mar 15 22:07:25.375: ccCallAlert
(callID=0x58, prog_ind=0x8, sig_ind=0x1)

!--- Участки вызова POTS и VoIP связаны друг с другом.

*Mar 15 22:07:25.375: ccConferenceCreate
(confID=0x81B6DEA0, callID1=0x58, callI
D2=0x59, tag=0x0)
*Mar 15 22:07:25.375: cc_api_bridge_done
(confID=0x1E, srcIF=0x81B09EFC, srcCall
ID=0x58, dstCallID=0x59, disposition=0,
tag=0x0)

!--- Обменяйтесь битовыми масками пропускной способности
!--- с удаленным шлюзом VoIP
!--- (Codec, VAD, VoIP или FAX, FAX-rate и т.д.).

*Mar 15 22:07:26.127: cc_api_caps_ind
(dstVdbPtr=0x81B09EFC, dstCallId=0x58, src
CallId=0x59, caps={codec=0x4, fax_rate=0x2,
vad=0x2, modem=0x1 codec_bytes=20,
signal_type=0})

!--- Оба шлюза согласовали пропускную способность.

*Mar 15 22:07:26.127: cc_api_caps_ack
(dstVdbPtr=0x81B09EFC, dstCallId=0x58, src
CallId=0x59, caps={codec=0x4, fax_rate=0x2,
vad=0x2, modem=0x1 codec_bytes=20,
signal_type=0})
*Mar 15 22:07:26.139: cc_api_caps_ack
(dstVdbPtr=0x81A75558, dstCallId=0x59, src
CallId=0x58, caps={codec=0x4, fax_rate=0x2,
vad=0x2, modem=0x1 codec_bytes=20,
signal_type=0})
*Mar 15 22:07:35.259: cc_api_call_digit
(vdbPtr=0x81B09EFC, callID=0x58, digit=T
, duration=0)
*Mar 15 22:07:35.259: sess_appl:
ev(9=CC_EV_CALL_DIGIT), cid(88), disp(0)
*Mar 15 22:07:35.259: ssaTraceSct:
cid(88)st(4)oldst(3)cfid(30)csize(0)in(1)
fDest(0)-cid2(89)st2(4)oldst2(1)
*Mar 15 22:07:35.399: cc_api_call_connected
(vdbPtr=0x81A75558, callID=0x59)
*Mar 15 22:07:35.399: sess_appl:
ev(8=CC_EV_CALL_CONNECTED), cid(89), disp(0)
*Mar 15 22:07:35.399: ssaTraceSct:
cid(89)st(4)oldst(1)cfid(30)csize(0)in(0)
fDest(0)-cid2(88)st2(4)oldst2(4)

!--- Вызов VoIP установлен.

*Mar 15 22:07:35.399: ccCallConnect
(callID=0x58)

!--- Вызов VoIP разорван. Cause = 0x10

*Mar 15 23:29:39.530: ccCallDisconnect
(callID=0x5B, cause=0x10 tag=0x0)
```

Если вызов разрывается, и причина, скорее всего, в VoIP-части установки сигнала, вам, возможно, понадобится взглянуть на части установки сигнала H.225 или H.245 TCP, а не просто на UDP-часть установки H.323. Команды, которые можно использовать для отладки настройки вызова H.225 или H.245, приведены ниже:

- **debug ip tcp transactions** и **debug ip tcp packet** — эти команды проверяют TCP-часть согласования H.225 и H.245. Они возвращают IP-адреса, TCP-порты и состояния соединений TCP.
- **debug cch323 h225** — эта команда проверяет H.225-часть согласования вызова и отслеживает передачу статуса конечного автомата H.225 на основании обработанного события. Это можно представить как часть "уровень 1" из трех частей установки звонка H.323.
- **debug cch323 h245** — эта команда проверяет H.245-часть согласования вызова и отслеживает передачу статуса конечного автомата H.245 на основании обработанного события. Это можно представить как часть "уровень 2" из трех частей установки звонка H.323.

Сведения об ошибках качества обслуживания (QoS) VoIP

После того как все проблемы с вызовами VoIP должным образом решены, необходимо убедиться в качестве передаваемого голосового сигнала. Хотя в данном документе не описывается устранение неполадок, связанных с качеством обслуживания, данные инструкции следует учитывать при работе над хорошим качеством передачи голоса.

- Следует выяснить ширину полосы пропускания, которую вызов VoIP задействует с каждым из кодеков. Это включает заголовки уровня 2 и IP/UDP/RTP. Подробнее см. IP-телефония: использование полосы пропускания для каждого вызова.
- Следует выяснить характеристики IP-сети, через которую передается вызов. Например, полоса пропускания в сети frame-relay при согласованной скорости передачи данных (CIR) сильно отличается от полосы при скорости выше CIR (или при всплеске), когда пакеты могут отбрасываться или скапливаться в облаке Frame-Relay. Убедитесь, что задержка и помехи контролируются и устраняются, насколько это возможно. Задержка передачи по одному направлению не должна превышать 150 мс (по рекомендациям G.114).
- Используйте метод очереди, который позволяет идентифицировать трафик VoIP и устанавливать в нем приоритеты.
- При передаче VoIP по низкоскоростным каналам помните о возможности использования методов фрагментации пакетов уровня 2, таких как MLPPP с фрагментацией чередования каналов (LFI), в каналах точка-точка или FRF.12 в каналах Frame-Relay. Фрагментация пакетов данных большого размера позволяет снизить объем помех и задержки при передаче трафика VoIP, так как выполняется чередование пакетов в канале.
- Попытайтесь использовать разные кодеки и произвести вызов со включенным и выключенным VAD, чтобы по возможности уменьшить вывод в DSP, в отличие от IP-сети.

При наличии VoIP, устраняя неполадки QoS, следует прежде всего искать отброшенные пакеты и критические места сети, которые могут вызвать задержки и помехи.

Ищите:

- сбросы интерфейса
- сбросы буфера
- перегрузку интерфейса
- перегрузку канала

Необходимо проверить каждый интерфейс по пути соединения VoIP. Необходимо также устранить сбросы и перегрузки. Двухсторонняя задержка должна быть уменьшена настолько, насколько это возможно. Двухстороннюю задержку канала можно определить при помощи команд `ring`, посылающих пакеты между конечными точками VoIP. Когда это возможно, двухсторонняя задержка не должна превышать 300 мс. Если задержка вынужденным образом превышает это значение, следует предпринять усилия и убедиться в том, что она постоянна и не вносит помехи или переменную задержку.

Следует также выполнить проверку, чтобы убедиться, что механизм IOS формирования очереди размещает пакеты VoIP в правильной очереди. В проверке формирования очереди могут помочь такие команды IOS, как **show queue *interface*** или **show priority**.

Данные кодов причины и значений отладки для VoIP

Используйте данные таблицы при чтении процедур отладки и соответствующих значений в процессе отладки.

Q.931 Причины отключения вызова (cause_codes из debug voip ssaip inout)

Подробнее о кодах и значениях причин Q.931 см. "Типы коммутаторов ISDN, коды и значения"

Значение причины отключения вызова (в шестнадцатеричной системе)	Значение и номер (в десятичных числах)
CC_CAUSE_UANUM = 0x1	неназначенный номер. (1)
CC_CAUSE_NO_ROUTE = 0x3	нет маршрута к адресу назначения. (3)
CC_CAUSE_NORM = 0x10	обычный сброс вызова. (16)
CC_CAUSE_BUSY = 0x11	пользователь занят. (17)
CC_CAUSE_NORS = 0x12	пользователь не отвечает. (18)
CC_CAUSE_NOAN = 0x13	нет ответа от пользователя. (19)
CC_CAUSE_REJECT = 0x15	вызов отклонен. (21)
CC_CAUSE_INVALID_NUMBER = 0x1C	недопустимый номер. (28)
CC_CAUSE_UNSP = 0x1F	нормальный, не уточненный. (31)
CC_CAUSE_NO_CIRCUIT = 0x22	канал отсутствует. (34)
CC_CAUSE_NO_REQ_CIRCUIT = 0x2C	канал отсутствует. (44)
CC_CAUSE_NO_RESOURCE = 0x2F	нет ресурса. (47)
CC_CAUSE_NOSV = 0x3F	служба или параметр недоступны или не определены. (63)

Значения согласования кодека (из debug voip ssaip inout)

Подробнее о кодеках см. VoIP - общие сведения о кодеках: уровень сложности, поддержка, оценка MOS и согласование.

Значение согласования	Значение
codec=0x00000001	G711 ULAW 64K PCM
codec=0x00000002	G711 ALAW 64K PCM
codec=0x00000004	G729
codec=0x00000004	G729IETF
codec=0x00000008	G729a
codec=0x00000010	G726r16
codec=0x00000020	G726r24
codec=0x00000040	G726r32
codec=0x00000080	G728
codec=0x00000100	G723r63
codec=0x00000200	G723r53
codec=0x00000400	GSMFR
codec=0x00000800	G729b
codec=0x00001000	G729ab
codec=0x00002000	G723ar63
codec=0x00004000	G723ar53
codec=0x00008000	CLEAR_CHANNEL

Типы сигналов

Типы сигналов	Значение
---------------	----------

CC_TONE_RINGBACK 0x1	Сигнал вызова
CC_TONE_FAX 0x2	Сигнал вызова факса
CC_TONE_BUSY 0x4	Сигнал "занято"
CC_TONE_DIALTONE 0x8	Тональный сигнал
CC_TONE_OOS 0x10	Тональный сигнал нерабочего состояния
CC_TONE_ADDR_ACK 0x20	Сигнал запроса адреса
CC_TONE_DISCONNECT 0x40	Тоновый сигнал отключения
CC_TONE_OFF_HOOK_NOTICE 0x80	Тональный сигнал, означающий, что телефон оставлен в состоянии ответа абонента.
CC_TONE_OFF_HOOK_ALERT 0x100	Более современная версия CC_TONE_OFF_HOOK_NOTICE
CC_TONE_CUSTOM 0x200	Абонентский сигнал - используется при указании специального сигнала абонента
CC_TONE_NULL 0x0	Нулевой сигнал

Значения возможностей FAX-Rate и VAD

Значения	Значение
CC_CAP_FAX_NONE 0x1	Факс выключен или недоступен
CC_CAP_FAX_VOICE 0x2	Голосовой вызов
CC_CAP_FAX_144 0x4	14400 бод
CC_CAP_FAX_96 0x8	9600 бод
CC_CAP_FAX_72 0x10	7200 бод
CC_CAP_FAX_48 0x20	4800 бод

CC_CAP_FAX_24 0x40	2400 бод
CC_CAP_VAD_OFF 0x1	VAD выключен
CC_CAP_VAD_ON 0x2	VAD включен

Дополнительные сведения

- Настройка планов дозвона, точек вызова и обработки цифр
- Команды отладки VoIP
- "Устранение неполадок уровня 1 T1"
- Блок-схема устранения неполадок T1
- Устранение неполадок, связанных с проблемами последовательного канала
- Поддержка технологий голосовой связи
- Поддержка продуктов для IP-телефонии и голосовой связи
- Системы электронного обучения Центра технической поддержки в области передачи голосовых сообщений, IP-телефонии и обмена сообщениями
- Рекомендуемая литература: Устранение неполадок в системах IP-телефонии Cisco
- Техническая поддержка - Cisco Systems

© 1992-2010 Cisco Systems, Inc. Все права защищены.

Дата генерации PDF файла: Jan 05, 2010

http://www.cisco.com/support/RU/customer/content/9/92126/voip_debugcalls.shtml
