

Устранение неполадок голосовых магистральных каналов

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Условные обозначения](#)

[Проблема](#)

[Решение](#)

[Общие проблемы магистральных каналов](#)

[Начните устранять неполадки](#)

[Определите, какие Вызовы подключены](#)

[Устранение неполадок DTMF](#)

[Дополнительные сведения](#)

Введение

Голосовые магистральные каналы постоянно устанавливают голосовые вызовы, или Передача голоса по IP (VoIP), передача голоса по Frame Relay (VoFR), или Передача голоса по ATM (VoATM). Вызовы устанавливаются, как только включается маршрутизатор и завершается настройка. Как только голосовые порты включены, голосовые порты автоматически набирают фиктивный номер телефона, заданный под голосовым портом, и заказывают телефонный разговор с местоположением. Голосовые порты завершают вызов к другому концу через соответствующие точки вызова. Как только это соединение установлено, насколько маршрутизатор затронут, голосовой вызов находится на сеансе и связан.

Предварительные условия

Требования

Для данного документа отсутствуют предварительные условия.

Используемые компоненты

Настоящий документ не имеет жесткой привязки к каким-либо конкретным версиям программного обеспечения и оборудования.

Сведения, содержащиеся в данном документе, были получены с устройств в специальной

лабораторной среде. Все устройства, описанные в данном документе, были запущены с конфигурацией по умолчанию. Если ваша I сеть является оперативной, удостоверьтесь, что вы понимаете потенциальное воздействие любой команды перед использованием его.

Условные обозначения

[Дополнительные сведения об условных обозначениях см. в документе Технические рекомендации Cisco. Условные обозначения.](#)

Проблема

Типичные проблемы, которые принадлежат транкам, являются очевидными для маршрутизатора и очень трудными устранить неполадки. Общие проблемы, замеченные с речевыми соединительными линиями, проявлены, когда вызов размещен по транкам, и ничто не слышат. Это - одна из известных неполадок с магистралями и вызвано многими другими вопросами. Другой проблемой является Двухтональный многочастотный набор (DTMF) тоны, которые не передают должным образом, и сигнализирующий от PrivateBranch Exchange (PBX) (внутренняя автоматическая телефонная станция) до УАТС не транспортируются должным образом. В данном документе приводится описание устранения этих проблем.

Когда речевые грузовики подключены и активны, сигналы ведут себя по-другому в грузовиках соединения. Любые команды, которые вы обычно выполняете под голосовым портом для характеристик сигнала, не релевантны и полезны. Речевая соединительная линия становится сигнальным conduit и передает сигнал через ссылку VoIP. При использовании речевых соединительных линий сигнализация УАТС должна совпасть от начала до конца. Насколько две машины УАТС затронуты, цель состоит в том, чтобы заставить соединение речевой соединительной линии выглядеть идентичным арендованной линии T1 к УАТС с маршрутизаторами, абсолютно прозрачными, в то время как ясная ссылка установлена между двумя PBXs во всем процессе.

Когда транк подходит, транк становится кабелем программного обеспечения, и тип сигнала считают типом разъема. Транк не заботится о типе сигнала, который используется. Даже если сигнал не совпадает в обоих концах, транк все еще подходит. Пока PBXs в обоих концах делают ту же сигнализацию, транки функционируют должным образом.

Решение

Подход для взятия, когда вы решаете проблемы магистрали, является другим, чем это, используется для коммутируемых вызовов. Для наблюдения, что действительно происходит, после, транки проверены, необходимо обратиться к сигнализации УАТС. Прежде чем вы продолжите посмотреть на сигнализацию, проверить, что транки подключены и что Цифровые процессоры сигналов (DSP) обрабатывают голосовые пакеты.

Примечание: Вы, вероятно, хотите отключить Обнаружение активности речи (VAD) для устранения проблем. Как только это проверено, что транки функционируют правильно, необходимо посмотреть на Телефонную сигнализацию для устранения проблем далее.

Если транки установлены, и никто не пытается позвонить, сообщения поддержки транка передаются назад и вперед между удаленными коробками. Эти пакеты Keeralive проверяют

магистральное подключение и несут сигнальную информацию от от начала до конца. Для проверки этих пакетов Keeralive выполните [команду debug vpm signal](#). Если существует много транков, выходные данные от команд `debug vpm`, можно ограничить выходные данные одним портом при запуске `debug vpm port x` параметра командной строки где "x" является рассматриваемым голосовым портом. Это - выходные данные от команды `debug vpm signal`, выполненной при рассмотрении всех портов:

```
21:18:12: [3/0:10(11)] send to dsp sig DCBA state 0x0
21:18:12: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:12(13)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:20(21)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:12(13)] send to dsp SIG DCBA state 0x0
21:18:12: [3/0:20(21)] send to dsp SIG DCBA state 0x0
21:18:12: [3/0:0(1)] send to dsp SIG DCBA state 0x0
21:18:12: [3/0:3(4)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:9(10)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:3(4)] send to dsp SIG DCBA state 0x0
21:18:13: [3/0:9(10)] send to dsp SIG DCBA state 0x0
21:18:13: [3/0:19(20)] rcv from dsp SIG DCBA state 0x0
```

Если вы ограничиваете это, с [debug vpm port x](#) команда, отладки, намного легче интерпретировать, как показано в данном примере:

```
21:21:08: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:12: [3/0:0(1)] send to dsp SIG DCBA state 0x0
21:21:13: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:17: [3/0:0(1)] send to dsp SIG DCBA state 0x0
21:21:18: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:22: [3/0:0(1)] send to dsp SIG DCBA state 0x0
21:21:23: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:27: [3/0:0(1)] send to dsp SIG DCBA state 0x0
21:21:28: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:32: [3/0:0(1)] send to dsp SIG DCBA state 0x0
```

Пакеты Keeralive передаются и получаются каждые пять секунд. Сроки, "передаваемые dsp" и "полученный от dsp", с точки зрения Cisco IOS®. УАТС замены для DSP для создания его более понятным. Это сообщения, которые замечены, в то время как нет никакого действия на транках. С помощью сообщений проверки активности обеспечивается оповещение всех маршрутизаторов сети об активности магистралей. Когда пять из этих сообщений пропущены подряд, транк выключается. То, если транки постоянно колеблются в сети, одна из причин. Чтобы проверить, переданы ли пакеты Keeralive речевой соединительной линии и получены, выполните команду `debug vpm trunk-sc`. Эта отладка не генерирует выходных данных, пока не пропущены поддержки транка. Когда пакеты Keeralive пропущены, это - пример выходных данных команды `debug vpm trunk-sc`:

```
22:22:38: 3/0:22(23): lost Keepalive
22:22:38: 3/0:22(23): TRUNK_SC state : TRUNK_SC_CONN_WO_CLASS, event TRUNK_RTC_LOST_KEEPALIVE
22:22:38: 3/0:22(23): trunk_rtc_set_AIS on
22:22:38: 3/0:22(23): trunk_rtc_gen_pattern : SIG pattern 0x0
22:22:38: 3/0:22(23): TRUNK_SC, TRUNK_SC_CONN_WO_CLASS ==> TRUNK_SC_CONN_DEFAULT_IDLE
22:22:39: 3/0:13(14): lost Keepalive 22:22:39: 3/0:13(14): TRUNK_SC state :
TRUNK_SC_CONN_WO_CLASS, event TRUNK_RTC_LOST_KEEPALIVE 22:22:39: 3/0:13(14): trunk_rtc_set_AIS
on 22:22:39: 3/0:13(14): trunk_rtc_gen_pattern : SIG pattern 0x0 22:22:39: 3/0:13(14): TRUNK_SC,
TRUNK_SC_CONN_WO_CLASS ==> TRUNK_SC_CONN_DEFAULT_IDLE
```

Если никакие выходные данные не замечены, когда [команда debug vpm trunk-sc](#) выполнена, то никакие пакеты Keeralive не пропущены. Даже если пакеты Keeralive пропущены, транк остается вплоть до пяти последовательных сообщений, пропущены. Это означает, что соединение должно не работать в течение 25 секунд, прежде чем выключатся транки.

Общие проблемы магистральных каналов

Существует несколько дефектов, привязанных к соединениям речевой соединительной линии. Проверьте эти дефекты, если вы видите что-либо необычное. К тому времени, когда программное обеспечение Cisco IOS 12.2 было освобождено, большинство этих проблем было решено и интегрировано. Можно просмотреть дефекты для создания себя знающими, что это причины проблем с более старым кодом. Одна из наиболее распространенных проблем должна заставить PBXs сигнализировать правильно по магистральному соединению. Это походит на хорошую идею перевести транки в нерабочее состояние и настроить маршрутизаторы так, чтобы они работали над каждым концом, но подход действительно контрпроизводителен начиная с чего-либо, что вы изменили, теперь становится спорным, как только установлены транки. Лучший способ устранить неполадки с транками и функционален.

Начните устранять неполадки

Необходимо посмотреть на основы, чтобы установить, что они функционируют правильно:

- Созданы ли магистрали? Выполните команду **show voice call summary** и удостоверьтесь, что транки находятся в состоянии `S_CONNECTED`.
- Обработывают ли DSP пакеты? Выполните команду **show voice dsp** для проверки этого. Если вы не видите, что пакеты обработаны DSP, это - потому что VAD включен и, подавляет пакеты. Выключите VAD, создайте магистраль повторно и посмотрите снова. Кроме того, проверьте, что счетчики пакетов инкрементно увеличиваются, когда выполнена команда **show call active voice brief**. Эта команда также показывает, включен ли VAD для рассматриваемого журнала вызовов.

Если транки подключение к аналоговым портам на каком-либо узле, лучше проверять использование УАТС в переданном по транку режиме. Для устранения проблем аналоговых неполадок подключения Е&М обратитесь к [Пониманию и Устранению проблем Аналоговых Типов интерфейса Е&М и Проводному соединению Расположений](#). Как только все проверено и функционирует правильно, переведите транки в рабочее состояние и посмотрите на сигнализацию, которую передают между PBXs.

Идеальный способ для устранения проблем с подключением речевой соединительной линии должен исследовать сигнализацию, которую передают между PBXs. Лучше иметь сеанс Telnet к каждому рассматриваемому маршрутизатору так, чтобы сигнализация могла наблюдаться, поскольку это передают от одного конца до другого. Этот документ использует подмигивание Е&М, сигнализирующее, так как это довольно популярно, и время контрольного импульса должно быть учтено.

Это - выходные данные от маршрутизатора, связанного с УАТС, которая инициирует вызов:

```
May 22 19:39:03.582: [3/0:0(1)] rcv from dsp sig DCBA state 0x0
!--- It is in idle state. May 22 19:39:07.774: [3/0:0(1)] send to dsp SIG DCBA state 0x0 !---
ABCD bits=0000. May 22 19:39:08.586: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22
19:39:12.778: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:13.586: [3/0:0(1)] rcv from
dsp SIG DCBA state 0x0 May 22 19:39:17.777: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22
19:39:18.593: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:22.781: [3/0:0(1)] send to
dsp SIG DCBA state 0x0 May 22 19:39:23.593: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22
19:39:27.781: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:28.597: [3/0:0(1)] rcv from
dsp SIG DCBA state 0x0 May 22 19:39:32.785: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22
19:39:33.597: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:37.789: [3/0:0(1)] send to
dsp SIG DCBA state 0x0 May 22 19:39:38.601: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22
```


19:40:05.564: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:10.272: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:10.568: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:15.276: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:15.572: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:19.676: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:19.696: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:19.716: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.736: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.756: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.776: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.796: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.796: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:19.816: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.816: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:19.836: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.836: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 *!--- Both side hung up, back to idle state.* May 22 19:40:19.856: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.856: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.876: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.876: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.896: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.896: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.916: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.916: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.936: [3/0:0(1)] send to dsp SIG DCBA state 0x0

Эти выходные данные показывают, что маршрутизатор завершает вызов. Сетевой протокол синхронизации (NTP) синхронизирован.

May 22 19:39:03.582: [3/0:0(1)] send to dsp SIG DCBA state 0x0
May 22 19:39:07.774: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
!--- Idle state, both side on-hook. May 22 19:39:08.586: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:12.774: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:13.586: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:15.383: [1/0:0(1)] Signaling RTP packet has no particle *!--- You will see this message if you are running Cisco IOS !--- Software Release 12.2(1a) or later. It is not an error !--- message, it is a normal functioning state.*
May 22 19:39:17.774: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:18.590: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:22.778: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:23.594: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:27.782: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:28.598: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:32.782: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:33.598: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:37.786: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:38.602: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:39.778: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:39.798: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:39.818: [3/0:0(1)] send to dsp SIG DCBA state 0xF *!--- Remote side off-hook, this is conveyed to the PBX.* May 22 19:39:39.838: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.858: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.878: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.898: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.918: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.938: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.958: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.978: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.998: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.018: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.038: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.058: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.078: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.090: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.098: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.110: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.118: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.130: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF *!--- Receive wink from PBX.* May 22 19:39:40.138: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.150: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.158: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.170: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.178: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.190: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.198: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.210: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.218: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.230: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.238: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.250: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.258: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.270: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.290: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.310: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.330: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.350: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 *!--- Wink ended, waiting for an answer.* May 22 19:39:40.370: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.390:

[3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.410: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.430: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.450: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.470: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.490: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.510: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.530: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.550: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.570: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.590: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.610: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.630: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.650: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.670: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.690: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.710: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.730: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.750: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.770: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:45.262: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:45.770: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:50.077: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:50.097: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:50.117: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF *!--- Receive off-hook from PBX.* May 22 19:39:50.137: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.157: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.177: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.197: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.217: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.237: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.257: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.261: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.277: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.297: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.317: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.337: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.357: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.377: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.397: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.417: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.437: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.457: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.477: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.497: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.517: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.537: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.557: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF *!--- Both sides off-hook, the conversation happens.* May 22 19:39:55.265: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:55.557: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:00.269: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:00.561: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:05.269: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:05.561: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:10.273: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:10.565: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:15.273: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:15.569: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:19.673: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:19.693: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:19.713: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.733: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.753: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.773: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.793: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.797: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:19.813: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.817: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:19.833: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.837: [3/0:0(1)] send to dsp SIG DCBA state 0x0 *!--- Both sides are back on-hook, back to idle.* May 22 19:40:19.853: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.857: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.873: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.877: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.893: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.897: [3/0:0(1)] send to dsp SIG DCBA state 0x0

Примечание: Эти выходные данные показывают сигнализацию, которая происходит с обеих сторон речевой соединительной линии, которая использует сигнализацию подмигивания E&M. Другие типы сигнализации могут быть замечены, который использует эти те же отладки. Если вы видите вызовы, установленные правильно (как показано здесь), двухстороннее аудио должно присутствовать. Это может быть проверено при рассмотрении или **show voice dsp** или на выходных данных команды **show call active voice brief**. Если все надеется быть прекрасным там, и вы получаете проблемы со звуком (никакое аудио или односторонние) с аналоговыми соединениями, проверяете эти соединения снова.

Определите, какие Вызовы подключены

Так как это делает минимальное хорошее для рассмотрения **show call active voice** или выходных данных **команды show voice call summary** для вызовов обеспеченный с помощью транка, вам нужен простой метод для определения, какие речевые соединительные линии поддерживают активные вызовы. Один из самых легких способов сделать это должно выполнить **команду show voice trunk-conditioning signaling** в сочетании с **включать** параметром и **ABCD** использования как включенная строка, как показано в здесь:

```
Phoenix#show voice trunk-conditioning signaling | include ABCD last-TX-ABCD=0000, last-RX-
ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-
ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-
ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=1111, last-RX-ABCD=0000 !---
Timeslot 8. last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=1111, last-RX-ABCD=1111 !---
Timeslot 10. last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-
ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-
ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000
```

Примечание: Эти выходные данные показывают вызов, активный на временном интервале 10 и другой вызов, запускаемый на временном интервале восемь. Вы хотите сделать псевдоним для этой довольно длинной команды при использовании ее много.

Устранение неполадок DTMF

Кроме со снятой трубкой и сигнала свободно, единственная другая вещь, что проходом маршрутизаторов между PBXs (помимо голоса) являются Тоны DTMF. Существует также аудиопуть, таким образом, это обычно - не проблема, но существует проблема. Проблема возникает с тем, как вы делаете аудио по тому пути. Иногда предпочтительно использовать низкоскоростные кодеки для сохранения пропускной способности. Проблема подходит, что эти низкоскоростные кодеки разработаны посредством алгоритмов, которые были записаны для речи. Тоны DTMF не соответствуют этим алгоритмам очень хорошо и нуждаются в некотором другом методе для передачи, пока клиент не использует g711 кодек. Ответ заключается в **команде dtmf-relay**. Эта функция позволяет DSP в конце, запускает тон, чтобы распознать Тон DTMF и разделить его от обычного аудиопотка. Основанный о том, как это настроено, DSP тогда кодирует этот тон или как разный тип пакета Протокола RTP или как поскольку h245 обменивается сообщениями, чтобы быть переданным через ссылку отдельно от аудиопотка. Тот же процесс используется при выполнении команд **fax-relay** и **modem-relay**.

Эта функция излагает другую проблему отладки устранению проблем магистрали. Как вы проверяете, какие цифры передают, если нет никакой настройки вызова, и необходимо ли извлечь ту информацию из потока пакетов между маршрутизаторами? Как сделать, это зависит от того, какая **команда dtmf-relay** используется.

Как показано в данном примере, **команде cisco-rtsp dtmf-relay**, использует составляющий собственность тип полезных данных Cisco, таким образом, необходимо вниз посмотреть на DSP для наблюдения этого. Можно выполнить **команду debug vpm signal** в сочетании с **командой debug vpm port x/x:y.z** (для ограничения выходных данных рассматриваемым портом), чтобы видеть, что цифры прошли к DSP в вызывающей стороне. Эти выходные данные отображены в вызывающей стороне, не в оконечной стороне.

```
*Mar 1 00:22:39.592: htsp_digit_ready: digit = 31
*Mar 1 00:22:39.592: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar 1 00:22:40.021: htsp_digit_ready: digit = 32
*Mar 1 00:22:40.021: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
```



```

*Mar 1 00:22:40.562: htsp_digit_ready: digit = 33
*Mar 1 00:22:40.562: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar 1 00:22:40.810: [1/0:1(2)] rcv from dsp SIG DCBA state 0xF
*Mar 1 00:22:41.131: htsp_digit_ready: digit = 34
*Mar 1 00:22:41.131: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar 1 00:22:41.499: [1/0:1(2)] Signaling RTP packet has no partical
*Mar 1 00:22:41.499: [1/0:1(2)] send to dsp SIG DCBA state 0xF
*Mar 1 00:22:41.672: htsp_digit_ready: digit = 35
*Mar 1 00:22:41.672: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar 1 00:22:42.192: htsp_digit_ready: digit = 36
*Mar 1 00:22:42.192: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar 1 00:22:42.789: htsp_digit_ready: digit = 37
*Mar 1 00:22:42.789: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar 1 00:22:43.350: htsp_digit_ready: digit = 38
*Mar 1 00:22:43.350: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar 1 00:22:44.079: htsp_digit_ready: digit = 39
*Mar 1 00:22:44.079: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar 1 00:22:45.249: htsp_digit_ready: digit = 30
*Mar 1 00:22:45.249: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar 1 00:22:45.810: [1/0:1(2)] rcv from dsp SIG DCBA state 0xF
*Mar 1 00:22:46.007: htsp_digit_ready: digit = 2A
*Mar 1 00:22:46.011: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar 1 00:22:46.572: [1/0:1(2)] Signaling RTP packet has no partical
*Mar 1 00:22:46.572: [1/0:1(2)] send to dsp SIG DCBA state 0xF
*Mar 1 00:22:46.628: htsp_digit_ready: digit = 23
*Mar 1 00:22:46.628: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar 1 00:22:50.815: [1/0:1(2)] rcv from dsp SIG DCBA state 0xF
all digits 0-9 are represented by 30-39, * = 2A and # = 23.

```

Можно проверить то, какие цифры передаются от вызывающей стороны с [командой dtmf-relay h245-alphanumeric](#). Команда `dtmf-relay h245-alphanumeric` использует алфавитно-цифровую часть h.245 для передачи тонов. Когда команда `debug h245 asn1` включена, как показано в данном примере, цифры могут легко быть замечены и при возникновении и при окончных сторонах транка:

Вызывающая сторона:

```

*Mar 1 00:34:17.749: H245 MSC OUTGOING PDU ::=
value MultimediaSystemControlMessage ::= indication : userInput : alphanumeric : "1"

*Mar 1 00:34:17.749: H245 MSC OUTGOING ENCODE BUFFER::= 6D 400131
*Mar 1 00:34:17.753:
*Mar 1 00:34:18.350: H245 MSC OUTGOING PDU ::=
value MultimediaSystemControlMessage ::= indication : userInput : alphanumeric : "2"

*Mar 1 00:34:18.350: H245 MSC OUTGOING ENCODE BUFFER::= 6D 400132
*Mar 1 00:34:18.350:
*Mar 1 00:34:18.838: H245 MSC OUTGOING PDU ::=
value MultimediaSystemControlMessage ::= indication : userInput : alphanumeric : "3"

*Mar 1 00:34:18.838: H245 MSC OUTGOING ENCODE BUFFER::= 6D 400133

```

Сторона завершения:

```

*Mar 1 17:45:16.424: H245 MSC INCOMING ENCODE BUFFER::= 6D 400131
*Mar 1 17:45:16.424:
*Mar 1 17:45:16.424: H245 MSC INCOMING PDU ::=
value MultimediaSystemControlMessage ::= indication : userInput : alphanumeric : "1"

*Mar 1 17:45:17.025: H245 MSC INCOMING ENCODE BUFFER::= 6D 400132
*Mar 1 17:45:17.025:
*Mar 1 17:45:17.025: H245 MSC INCOMING PDU ::=
value MultimediaSystemControlMessage ::= indication : userInput : alphanumeric : "2"

```

*Mar 1 17:45:17.514: H245 MSC INCOMING ENCODE BUFFER::= 6D 400133

*Mar 1 17:45:17.514:

*Mar 1 17:45:17.514: H245 MSC INCOMING PDU ::=

value MultimediaSystemControlMessage ::= indication : userInput : alphanumeric : "3"

Когда те же отладки как команда **dtmf-relay h245-alphanumeric** используются, команда [h245-сигнала dtmf-relay](#) подобна и может быть замечена. В целом, устранять неполадки магистралей с командой **dtmf-relay** является довольно трудным без упомянутых отладок.

[Дополнительные сведения](#)

- [Настройка и устранение неполадок Transparent CCS](#)
- [Поддержка голосовых технологий](#)
- [Поддержка продуктов голосовой и IP-связи](#)
- [Устранение неполадок в системах IP-телефонии Cisco](#)
- [Техническая поддержка - Cisco Systems](#)