

H.323 T.38 ファックス リレーの問題

目次

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[T.38 の基本](#)

[T.38 の制約事項](#)

[T.38 のネゴシエーション](#)

[T.38 のトラブルシューティング](#)

[H.323 または SIP T.38 ファックス リレーのトラブルシューティングのヒント](#)

[debug コマンドと show コマンド](#)

[正常な T.38 コールの出力](#)

[失敗した T.38 コールの例](#)

[関連情報](#)

概要

T.38 ファックス リレーの問題は、通常は、シスコ製とサードパーティ製の T.38 ファックス ゲートウェイの間の相互運用性の問題と関連しています。このドキュメントでは、成功および失敗する T.38 ファックス リレー コールの詳細な debug コマンドの例を示します。これらの debug コマンド出力には、相互運用性の問題を識別およびトラブルシューティングできるように、リファレンスポイントとなるコメントが含まれています。このドキュメントでは、関連するトラブルシューティング コマンドおよび確認コマンドも示してあります。

前提条件

要件

このドキュメントの読者はファックス リレーの基本概念に関する知識が必要です。ファックス リレーの概念および基本トラブルシューティング手順の詳細については、『[ファックスリレートラブルシューティングガイド](#)』を参照してください。

使用するコンポーネント

このドキュメントは、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、クリアな (デフォルト) 設定で作業を開始して

います。ネットワークが稼働中の場合は、コマンドが及ぼす潜在的な影響を十分に理解しておく必要があります。

表記法

ドキュメント表記の詳細は、『[シスコテクニカルティップスの表記法](#)』を参照してください。

T.38 の基本

T.38 ファクスリレーの問題には、ファクス トーンが聞こえて音声コールが確立されるが、ファクスのネゴシエーションが完了せず、最終的に通話がドロップされるという共通の症状があります。この問題は、Cisco T.38 ゲートウェイとサードパーティ製の T.38 ゲートウェイの相互運用性の問題に関連することがよくあります。

T.38 ファクスリレーは、リアルタイム ファクス送信です。つまり、2 台のファクス マシンは、この 2 台を直結する電話回線が存在しているかのように互いに通信します。ファクスリレーは、音声コール用にすでに定義および設定されているゲートウェイダイヤルピア上でいくつかの追加のコマンドを使用して設定されます。

シスコでは、2 方式のファクスリレーに対応しています。シスコ独自の方式と、ITU-T T.38 標準に基づく方式です。大部分のプラットフォームでは、ファクス方式を明示的に設定していない場合のデフォルトは、シスコファクスリレーです。シスコファクスリレーについては、「[シスコファクスリレーの設定](#)」を参照してください。

T.38 の制約事項

現時点では、Cisco T.38 ファクスリレーには、次の制限があります。

- T.38 相互運用性では、Cisco H.323 バージョン 2 が必要です。
- T.38 は、音声圧縮モジュール (VCM) を搭載した Cisco MC3810 シリーズ コンセントレータでは、サポートされていません。
- T.38 は、Multimedia Conference Manager (MCM) H.323 プロキシでは、サポートされていません。
- H.323 T.38 については、ユーザ データグラム プロトコル (UDP) だけが実装されています。
- さまざまな製造業者が H.323 および T.38 の特定の部分を選択して自社のゲートウェイおよびゲートキーパーに実装できるため、一部のサードパーティ製ゲートウェイおよびゲートキーパーは、T.38 ファクスリレー用の Cisco 音声ゲートウェイと非互換です。T.38 ファクスリレーが成功することを確認するために、これらのサードパーティ製ゲートウェイおよびゲートキーパーとの音声相互運用性テストを実行する必要があります。

T.38 のネゴシエーション

このセクションでは、シスコゲートウェイの内部における T.38 ネゴシエーションの処理方法に関する段階的な要約を示します。ファクスリレーの基本の詳細については、『[ファクスリレートラブルシューティングガイド](#)』を参照してください。

1. 初期メッセージ内で、発信元ゲートウェイ (OGW) によって T.38 データ機能がアナウンスされます。終端ゲートウェイ (TGW) が T.38 データ機能をサポートしている場合は、

OGW に送信される後続のメッセージ内に、この情報をリレーできます。

2. 音声コールが確立されて、TGW のデジタル信号プロセッサ (DSP) でファクストーンを検出すると、音声テレフォニー サービス プロバイダー (VTSP) のステート マシンが H.323 コール レッグを通知します。これにより、OGW と T.38 モードがネゴシエートされます。
3. T.38 モードの確認応答が行われると、音声チャンネルが閉じられ、T.38 論理チャンネルが両端でオープンされます。
4. VTSP コード レベルで、ファクス コーダ/デコーダ (コーデック) のダウンロードが行われます。
5. T.38 Open Logical Channel (OLC) とコーデック ダウンロードが成功すると、VTSP はファクス モードになります。
6. ファクス送信が完了すると通話は音声コールに戻ります。注: T.38 モードのネゴシエーション中に他方の端で T.38 モードを応答確認しない場合、通話は音声コールに戻って切断されます。T.38 OLC に関する否定応答を他方の端から受信した場合にも、通話は音声コールに戻って切断されます。

T.38 のトラブルシューティング

H.323 または SIP T.38 ファクス リレーのトラブルシューティングのヒント

T.38 ファクス リレーをトラブルシューティングするには、次の手順を実行します。

- **音声コールを行うことができることを確認します。** ファックスの接続性を調べる前に、通常の音声コールが実行できることを確認します。電話機が取り付けられていない場合は、ファックス機器を取り外して通常の電話機を接続します。通常の音声コールが接続されない場合、VoX 関連の問題である可能性があります。この場合は、ファクスのトラブルシューティングに進む前に、通常の音声接続の問題としてトラブルシューティングを行います。
- 発信元ゲートウェイと終端ゲートウェイの両方で fax protocol コマンドを使用して、目的のファクス プロトコルが設定されていることを確認します。
- 発信元ゲートウェイと終端ゲートウェイの両方について、グローバル設定レベルまたはダイヤルピア設定レベルで、ファクス プロトコルに T.38 が設定されていることを確認します。

debug コマンドと show コマンド

T.38 ファクス リレーのトラブルシューティングでは、次の debug コマンドと show コマンドを使用します。

- **debug voip ccapi inout** : このコマンドは、コール制御アプリケーション プログラム インターフェイス (API) を介して実行パスを追跡します。コール制御 API は、コール セッション アプリケーションと、下位のネットワーク固有のソフトウェアとの間のインターフェイスとして機能します。このコマンドからの出力結果を使用すると、音声ゲートウェイでのコールの処理方法がわかります。
- **debug vtsp all** —このコマンドはこれらのデバッグ VTSP コマンドを有効に します: **debug vtsp session**、**debug vtsp error**、および **debug vtsp dsp**) をイネーブルにするために使用します。
- **debug h245 asn1** : このコマンドでは、H.245 メッセージの抽象構文記法 1 (ASN.1) の内容が表示されます。デバッグ出力をディセーブルにするには、このコマンドの no 形式を使用します。

- **debug cch323 h245** : このコマンドは、処理されたイベントに基づいて、H.245 ステート マシンの状態遷移のトレースを表示します。デバッグ出力をディセーブルにするには、このコマンドの no 形式を使用します。
- **show call active fax brief** : このコマンドは、実行中のストア アンド フォワード ファクス送信の通話情報を表示します。
- **show call history fax** : このコマンドは、ファクスの最新の通話履歴を表示します。

正常な T.38 コールの出力

このセクションでは、AS5300 シリーズ ルータと Cisco 3640 モジュラ アクセス ルータの間の成功する T.38 ファクス設定の分析を詳しく説明しています。debug コマンドおよび show コマンドの出力は、Cisco AS5300 ユニバーサル ゲートウェイ上で TGW IOS 12.2 としてキャプチャされました。

debug vtsp all コマンド出力

```
!---After the voice call setup: !--- Usually, after the
call is connected, the ccCallConnect debug !--- message
is seen as follows: May 3 21:41:21.424: ccCallConnect
(callID=0x9), prog_ind = 0 May? 3 21:41:21.424:
ssaFlushPeerTagQueue cid(9) peer list: (empty) May 3
21:41:21.424: H.225 SM: process event
H225_EVENT_SETUP_CFM, for callID 9 May 3 21:41:21.424:
cch323_run_h225_sm: received event H225_EVENT_SETUP_CFM
while at state H225_ALERT May 3 21:41:21.424: H.225 SM:
changing from H225_ALERT state to H225_ACTIVE state for
callID 9 May 3 21:41:21.424: ==== PI in
cch323_h225_generic_send_setup_cfm = 0 !---After the
voice call is established, the TGW DSP detected fax
tone: May 3 21:41:26.741: vtsp_process_dsp_message:
MSG_TX_TONE_DETECT: type=0 trigger=1 tone_id=0 May 3
21:41:26.741: vtsp:[1:D (10), S_CONNECT,
E_DSP_TONE_DETECT] May 3 21:41:26.745:
vtsp_modem_proto_from_cdb: cap_modem_proto 0 May 3
21:41:26.745: cc_api_call_feature: (vdbPtr=0x624130C0,
callID=0xA,feature_ind.type=1 !---Switched to fax mode:
May 3 21:41:26.745: act_lfax_switch: cap_modem_proto=16,
fax_relay_on=1, state=19 May 3 21:41:26.745:
vtsp_t38_switchover:2 - data_mode:1 !--- Note that 2
means T.38; 1 means Cisco proprietary. May 3
21:41:26.745: cc_api_t38_fax_start
(dstVdbPtr=0x61B45A90, dstCallId=0x9, srcCallId=0xA,???)
caps={codec=0x10000, fax_rate=0x2, vad=0x2,
modem=0x0codec_bytes=160, signal_type=1}) May 3
21:41:26.745: vtsp_timer: 2016656 May 3 21:41:26.745:
sess_appl: ev(28=CC_EV_CALL_FEATURE), cid(10), disp(0)
May 3 21:41:26.745:
cid(10)st(SSA_CS_ACTIVE)ev(SSA_EV_CALL_FEATURE)
oldst(SSA_CS_CONFERENCED_ALERT)cfid(5)csize(0)in(0)fDest
(0) May 3 21:41:26.745: -
cid2(9)st2(SSA_CS_ACTIVE)oldst2(SSA_CS_CONFERENCING_ALER
T) !---H245 ModeRequest was sent to the OGW: May 3
21:41:26.745: ccCallFeature (callID=0x9, feature.type=1)
Set new event H245_EVENT_MR, for callID 9 May 3
21:41:26.745: cch323_run_h245_mr_sm: received event
H245_EVENT_MR while at state H245_MR_NONE? !---Above,
state H245_MR_NONE refers to ModeRequest state. May 3
21:41:26.745: H245 MSC OUTGOING PDU ::= value
MultimediaSystemControlMessage ::= request : requestMode
```

```
: ??? { ????? sequenceNumber 1 ????? requestedModes
?????{ ???????? { ?????????? { ????????????? type dataMode :
????????????? { ????????????????? application t38fax :
????????????????? { ????????????????????? t38FaxProtocol udp :
NULL ?????????????????? t38FaxProfile ?????????????? ???{
????????????????????? fillBitRemoval FALSE ??????????????????
transcodingJBIG FALSE ????????????????????? transcodingMMR
FALSE ????????????????????? version 0 ?????????????????????
t38FaxRateManagement transferredTCF : NULL
????????????????????? t38FaxUdpOptions ??????? ??????????????{
????????????????????????? t38FaxMaxBuffer 200
????????????????????????? t38FaxMaxDatagram 72
????????????????????????? t38FaxUdpEC t38UDPRedundancy : NULL
????????????????????????? } ????????????????????? } ?????????????????? }
???????????????????? bitRate 144 ?????????????? } ?????????????? }
????????? } ?????? } ??? } May 3 21:41:26.753: changing
from H245_MR_NONE state to H245_MR_WAIT_FOR_ACK state
May 3 21:41:26.861: vtsp_process_dsp_message:
MSG_TX_TONE_DETECT: type=0 trigger=0 tone_id=0 May 3
21:41:26.861: vtsp:[1:D (10), S_LFAX_WAIT_CAPS_ACK,
E_DSP_TONE_DETECT] May 3 21:41:26.865:
vtsp_process_event(): prev_state = 0.11 , state =
S_LFAX_WAIT_CAPS_ACK, event = E_DSP_TONE_DETECT ?Invalid
FSM? Input on channel 1:D
(10)h323chan_chn_process_read_socket: fd (3) of type
ACCEPTED has data PROCESS_READ: NOT COMPLETE, rc 10,
fd=3 May? 3 21:41:27.001: vtsp_process_dsp_message:
MSG_TX_TONE_DETECT: type=0 trigger=1 tone_id=0 May? 3
21:41:27.001: vtsp:[1:D (10), S_LFAX_WAIT_CAPS_ACK,
E_DSP_TONE_DETECT] May? 3 21:41:27.005:
vtsp_process_event(): prev_state = 0.11 , ?state =
S_LFAX_WAIT_CAPS_ACK, event = E_DSP_TONE_DETECT Invalid
FSM?Input on channel 1:D (10) May 3 21:41:27.101:
vtsp_process_dsp_message: MSG_TX_TONE_DETECT: type=0
trigger=0 tone_id=0 May 3 21:41:27.101: vtsp:[1:D (10),
S_LFAX_WAIT_CAPS_ACK, E_DSP_TONE_DETECT] May 3
21:41:27.105: vtsp_process_event(): prev_state = 0.11 ,
state = S_LFAX_WAIT_CAPS_ACK, event = E_DSP_TONE_DETECT
Invalid FSM Input on channel 1:D
(10)h323chan_chn_process_read_socket: fd (3) of type
ACCEPTED has data Hex representation of the received
TPKT0321000827000100 May 3 21:41:27.173: ? state = 0
bytesLeftToDecode = 4 May 3 21:41:27.173: H245 MSC
INCOMING ENCODE BUFFER ::= 27 000100 !---Received
ModeRequestAck from the OGW: May 3 21:41:27.173: H245
MSC INCOMING PDU ::= value
MultimediaSystemControlMessage ::= response :
requestModeAck : ??? { ????? sequenceNumber 1 ?????
response willTransmitMostPreferredMode : NULL ??? } Set
new event H245_EVENT_MR_CFM, for callID 9 May 3
21:41:27.173: cch323_run_h245_mr_sm: received event
H245_EVENT_MR_CFM while at state H245_MR_WAIT_FOR_ACK !-
--The voice LC is closed and the T.38 fax data LC is
opened: May 3 21:41:27.173: H245 MSC OUTGOING PDU ::=
value MultimediaSystemControlMessage ::= request :
closeLogicalChannel :? !---In the previous line,
LogicalChannel refers to the voice LC. ??? { ?????
forwardLogicalChannelNumber 1 ????? source user : NULL
??? } May 3 21:41:27.173: H245 MSC OUTGOING ENCODE
BUFFER ::= 04 00000000 May 3 21:41:27.173: send result :0
May 3 21:41:27.173: changing from H245_OLC_DONE state to
H245_OLC_NONE state May 3 21:41:27.173:
cch323_update_new_codec_info: Remote codec 17 May 3
21:41:27.173: cch323_update_new_codec_info:
```

```
negotiated_codec set(17)(40 bytes) May 3 21:41:27.173:
Changing to new event H245_EVENT_OLC May 3 21:41:27.177:
cch323_h245_olc_sm: received event H245_EVENT_OLC while
at state H245_OLC_NONE May 3 21:41:27.177: changing from
H245_OLC_NONE state to H245_OLC_WAIT state May 3
21:41:27.177: H245 MSC OUTGOING PDU ::= value
MultimediaSystemControlMessage ::= request :
openLogicalChannel :? !---In the previous line,
LogicalChannel refers to the T.38 or data LC. ??? {
????? forwardLogicalChannelNumber 2 ??????
forwardLogicalChannelParameters ?????? { ???????? dataType
data : ???????? { ?????????? application t38fax :
???????????? { ?????????????? t38FaxProtocol udp : NULL
???????????????? t38FaxProfile ?????????????? { ??????????????????
fillBitRemoval FALSE ?????????????????? transcodingJBIG FALSE
???????????????????? transcodingMMR FALSE ?????????????????? version
0 ?????????????????? t38FaxRateManagement transferredTCF :
NULL ?????????????????? t38FaxUdpOptions ?????????????? ??{
???????????????????? t38FaxMaxBuffer 200 ?????????????????????
t38FaxMaxDatagram 72 ????????????????????? t38FaxUdpEC
t38UDPRedundancy : NULL ?????????????????? } ?????????????? }
????????????? } ?????????????? maxBitRate 144 ?????????? } ??????????
multiplexParameters h2250LogicalChannelParameters :
????????? { ?????????????? sessionID 3? !---The previous line
refers to the data session ID. ??????????????
mediaControlChannel unicastAddress : ipAddress :
????????????? { ?????????????????? network 'AB44BA66'H ??????????????????
tsapIdentifier 17517 ?????????????? } ??????????????
silenceSuppression FALSE ?????????? } ?????? } ??? } May 3
21:41:27.181: H245 MSC OUTGOING ENCODE BUFFER::= 03
00000111 04118601 00805C01 00014007 C00200C8 01484000
90800B05 000300AB 44BA6644 6D00 May 3 21:41:27.181: send
result :0 May 3 21:41:27.181: OLC using T38Fax May 3
21:41:27.181: changing from H245_MR_WAIT_FOR_ACK state
to H245_MR_NONE state h323chan_chn_process_read_socket:
fd (3) of type ACCEPTED has data Hex representation of
the received TPKT032100090400000000 May 3 21:41:27.185:
? state = 0 bytesLeftToDecode = 5 May 3 21:41:27.185:
H245 MSC INCOMING ENCODE BUFFER::= 04 00000000 May 3
21:41:27.185: May 3 21:41:27.185: H245 MSC INCOMING PDU
::= value MultimediaSystemControlMessage ::= request :
closeLogicalChannel :?? !---In the previous line,
LogicalChannel refers to the voice LC. ??? { ??????
forwardLogicalChannelNumber 1 ?????? source user : NULL
??? } May? 3 21:41:27.185: H245 MSC OUTGOING PDU ::=
value MultimediaSystemControlMessage ::= response :
closeLogicalChannelAck :??? !---In the previous line,
LogicalChannel refers to the voice LC. ??? { ??????
forwardLogicalChannelNumber 1 ??? } May 3 21:41:27.185:
H245 MSC OUTGOING ENCODE BUFFER::= 23 800000 May 3
21:41:27.185: H245 MSC INCOMING ENCODE BUFFER::= 03
00000111 04118601 00805C01 00014007 C00200C8 01484000
90800B05 000300AC 10AF6941 7100 May 3 21:41:27.189: H245
MSC INCOMING PDU ::= value
MultimediaSystemControlMessage ::= request :
openLogicalChannel :? !---In the previous line,
LogicalChannel refers to the T.38 or data LC. ??? {
??????? forwardLogicalChannelNumber 2 ??????
forwardLogicalChannelParameters ?????? { ?????????? dataType
data : ?????????? { ?????????????? application t38fax :
???????????????? { ?????????????????? t38FaxProtocol udp : NULL
???????????????????? t38FaxProfile ?????????????????? { ?????????????????????
fillBitRemoval FALSE ????????????????????? transcodingJBIG FALSE
???????????????????? transcodingMMR FALSE ????????????????????? version
```



```
0 ?????????????? t38FaxRateManagement transferredTCF :
NULL ?????????????? t38FaxUdpOptions ?????????????? {
????????????????? t38FaxMaxBuffer 200 ??????????????????
t38FaxMaxDatagram 72 ?????????????????? t38FaxUdpEC
t38UDPRedundancy : NULL ?????????????????? } ?????????????? }
????????????? } ?????????????? maxBitRate 144 ?????????? } ??????????
multiplexParameters h2250LogicalChannelParameters :
????????? { ?????????????? sessionID 3 ??????????????
mediaControlChannel unicastAddress : ipAddress :
????????????? { ?????????????????? network 'AC10AF69'H ??????????????
tsapIdentifier 16753 ?????????????? } ??????????????
silenceSuppression FALSE ???? ???} ?????? } ??? } !---DSP
started T.38 fax codec download: May 3 21:41:27.193:
cc_api_t38_fax_start (dstVdbPtr=0x624130C0,
dstCallId=0xA, srcCallId=0x9, ??? caps={codec=0x10000,
fax_rate=0x2, vad=0x2, modem=0x codec_bytes=160,
signal_type=1}) May 3 21:41:27.193: vtsp:[1:D (10),
S_LFAX_WAIT_CAPS_ACK, E_CC_T38_START] May 3
21:41:27.193: act_caps_ack_lfax_dnld May 3 21:41:27.193:
vtsp_timer_stop: 2016700 May 3 21:41:27.193:
dsp_idle_mode: [1:D (10)] packet_len=8 channel_id=8481
packet_id=68 May 3 21:41:27.193:
cc_api_local_codec_dnld_done (dstVdbPtr=0x61B45A90,
dstCallId=0x9, srcCallId=0xA) May 3 21:41:27.193:
vtsp_timer: 2016700cch323_h245_local_codec_dnld_done:
negotiatedCodec[17] May 3 21:41:27.197: Changing to new
event H245_EVENT_OLC_IND May 3 21:41:27.197:
cch323_h245_olc_sm: received event H245_EVENT_OLC_IND
while at state H245_OLC_WAIT May 3 21:41:27.197: H245
MSC OUTGOING PDU ::= value
MultimediaSystemControlMessage ::= response :
openLogicalChannelAck : ??? { ??????
forwardLogicalChannelNumber 2 ??????
forwardMultiplexAckParameters
h2250LogicalChannelAckParameters : ?????? { ??????????
sessionID 1 ?????????? mediaChannel unicastAddress :
ipAddress : ?????????? { ?????????????? network 'AB44BA66'H
????????????? tsapIdentifier 17516 ?????????? } ??????
??mediaControlChannel unicastAddress : ipAddress :
????????? { ?????????????? network 'AB44BA66'H ??????????????
tsapIdentifier 17517 ?????????? } ?????????? flowControlToZero
FALSE ?????? } ??? } May 3 21:41:27.197: H245 MSC
OUTGOING ENCODE BUFFER: := 22 C0000104 80145C00 00AB44BA
66446C00 AB44BA66 446D0300 0100 May 3 21:41:27.589: ?
state = 0 bytesLeftToDecode = 4 May 3 21:41:27.589: H245
MSC INCOMING ENCODE BUFFER::= 23 800000 May 3
21:41:27.589: May 3 21:41:27.589: H245 MSC INCOMING PDU
::= value MultimediaSystemControlMessage ::= response :
closeLogicalChannelAck : ??? { ??????
forwardLogicalChannelNumber 1 ??? } May 3 21:41:27.789:
H245 MSC INCOMING ENCODE BUFFER: := 22 C0000104 80145C00
00AC10AF 69417000 AC10AF69 41710300 0100 May 3
21:41:27.789: H245 MSC INCOMING PDU ::= value
MultimediaSystemControlMessage ::= response :
openLogicalChannelAck : ??? { ??????
forwardLogicalChannelNumber 2 ??????
forwardMultiplexAckParameters
h2250LogicalChannelAckParameters : ?????? { ??????????
sessionID 3 ?????????? mediaChannel unicastAddress :
ipAddress : ?????????? { ?????????????? network 'AC10AF69'H
????????????? tsapIdentifier 16752 ?????????? } ??????????
mediaControlChannel unicastAddress : ipAddress : ??????????
{ ?????????????? network 'AC10AF69'H ?????????????? tsapIdentifier
16753 ?????????? } ?????????? flowControlToZero FALSE ?????? }
```

```

??? } May 3 21:41:27.793: Changing to new event
H245_EVENT_OLC_CFM May 3 21:41:27.793:
cch323_h245_olc_sm: received event H245_EVENT_OLC_CFM
while at state H245_OLC_WAIT May 3 21:41:27.793:
changing from H245_OLC_WAIT state to H245_OLC_DONE state
May 3 21:41:27.793: cc_api_t38_fax_start
(dstVdbPtr=0x624130C0, dstCallId=0xA, srcCallId=0x9,
???? caps={codec=0x10000, fax_rate=0x2, vad=0x2,
modem=0x0 codec_bytes=160, signal_type=1}) May 3
21:41:27.793: H.225 SM: process event
H225_EVENT_H245_SUCCESS, for callID 9 May 3
21:41:27.793: cch323_run_h225_sm: received event
H225_EVENT_H245_SUCCESS while at state H225_ACTIVE May 3
21:41:27.793: cc_api_remote_codec_dnld_done
(dstVdbPtr=0x624130C0, dstCallId=0xA, srcCallId=0x9) May
3 21:41:27.793: vtsp:[1:D (10), S_LFAX_WAIT_FAX,
E_CC_T38_START] May 3 21:41:27.793: vtsp:[1:D (10),
S_LFAX_WAIT_FAX, E_CC_T30_CAP_ACK] May 3 21:41:27.793:
act_t38_lfax_mode May 3 21:41:27.793: vtsp_timer_stop:
2016760 May 3 21:41:27.793: cc_api_set_fax_mode
(dstVdbPtr=0x61B45A90, dstCallId=0x9, srcCallId=0xA) May
3 21:41:27.793: dsp_idle_mode: [1:D (10)] packet_len=8
channel_id=8481 packet_id=68 May 3 21:41:27.793:
dsp_encap_config: T38 May 3 21:41:27.793: dsp_fax_mode:
[1:D (10)] FaxRate 0x2, Codec 0x10000? dsp_fax_mode()
ECM_DISABLE not set, debug_info not requested May 3
21:41:27.793: dsp_fax_mode:[1:D (10)] packet_len=28
channel_id=8481 packet_id=69 max_trans=6 info_size=20,
fax_protocol_type=3,hs_data_len=40, ls_data_red=0,
hs_data_red=0, tcf_handling=2, fax_relay_cntl=0x0
nsf_country = 0xAD, nsf_mfg = 0x0051 May 3 21:41:29.621:
ccGetCallActive (next=1, setup_time=0x0, index=0x0,
p=0x6293A8C0) May 3 21:41:29.621: ccGetCallActive
(next=1, setup_time=0x1EC241, index=0x1, p=0x6293A8C0)

```

失敗した T.38 コールの例

これは、失敗する T.38 コールの debug コマンド出力の例です。

debug vtsp all コマンド出力

```

!---When the ModeRequest was sent, T35 nonStandard was
sent instead of T38: *Jun 14 15:35:01.743: PDU DATA =
61593960 value MultimediaSystemControlMessage ::=
request : requestMode : ??? { ????? sequenceNumber 12
????? requestedModes ????? { ?????? { ?????????? {
????????????? type dataMode : ?????????????? { ??????????????
application nonStandard : ?????????????? {
????????????????? nonStandardIdentifier h221NonStandard :
????????????????? { ?????????????????? t35CountryCode 181 ?
?????????????????t35Extension 0 ??????????????????
manufacturerCode 20 ?????????????????? } ??????????????????
data '543338466178554450'H ?????????????????? } ??????????????????
bitRate 144 ?????????????? } ?????????? } ?????? }
??? } Set new event H245_EVENT_MR_IND, for callID C *Jun
14 15:35:01.751: cch323_run_h245_mr_sm: received event
H245_EVENT_MR_IND wh ile at state H245_MR_NONE *Jun 14
15:35:01.751: Scan Preferred List for g729r8PDU DATA =
61593960 value MultimediaSystemControlMessage ::=
response : requestModeAck : ??? { ????? sequenceNumber
12 ????? response willTransmitMostPreferredMode : NULL

```



```
??? } RAW_BUFFER::= 27 000C00 *Jun 14 15:35:01.751: PDU
DATA = 61593960 value MultimediaSystemControlMessage ::=
request : closeLogicalChannel : ??? { ??
???forwardLogicalChannelNumber 2 ????? source user :
NULL ??? } RAW_BUFFER::= 04 00000100 *Jun 14
15:35:01.751: *Jun 14 15:35:01.751: changing from
H245_OLC_DONE state to H245_OLC_NONE state *Jun 14
15:35:01.751: cch323_update_new_codec_info: Remote codec
17 *Jun 14 15:35:01.751: cch323_update_new_codec_info:
negotiated_codec set(17)(40 bytes) *Jun 14 15:35:01.751:
Changing to new event H245_EVENT_OLC *Jun 14
15:35:01.751: cch323_h245_olc_sm: received event
H245_EVENT_OLC while atstate H245_OLC_NONE *Jun 14
15:35:01.751: changing from H245_OLC_NONE state to
H245_OLC_WAIT state PDU DATA = 61593960 value
MultimediaSystemControlMessage ::= request :
openLogicalChannel : ??? { ?????
forwardLogicalChannelNumber 3 ?????
forwardLogicalChannelParameters ????? { ?????? dataType
data : ?????? { ?????? application nonStandard :
???????? { ????????? nonStandardIdentifier
h221nonStandard : ????????? { ?????????
t35CountryCode 181 ????????? t35Extension 0
???????? manufacturerCode 18 ? ?????????}
???????? data '543338466178554450'H ????????? }
???????? maxBitRate 144 ?????? } ??????
multiplexParameters h2250LogicalChannelParameters :
???????? { ????????? sessionID 3 ?????????
mediaControlChannel unicastAddress : ipAddress : ??????
?? { ????????? network 'C95C381E'H ?????????
tsapIdentifier 18101 ????????? } ?????? } ?????? } ??? }
RAW_BUFFER::= 03 00000210 08B50000 12095433 38466178
55445000 90800A04 000300C9 5C381E46 B5 *Jun 14
15:35:01.759: *Jun 14 15:35:01.759: OLC using T38Fax
*Jun 14 15:35:01.783: Changing to new event
H245_PROCESS_H245CONTROL *Jun 14 15:35:01.783:
cch323_h245_connection_sm:H245_CONNECT: received event
H24 5_PROCESS_H245CONTROL while at H245_CONNECTED state
RAW_BUFFER::= 04 80000100 800100 *Jun 14 15:35:01.783:
PDU DATA = 61593960 value MultimediaSystemControlMessage
::= request : closeLogicalChannel : ??? { ?????
forwardLogicalChannelNumber 2 ????? source user : NULL
????? reason unknown : NULL ??? } PDU DATA = 61593960
value MultimediaSystemControlMessage ::= response :
closeLogicalChannelAck : ??? { ?????
forwardLogicalChannelNumber 2 ??? } RAW_BUFFER::= 23
800001 *Jun 14 15:35:01.787: *Jun 14 15:35:01.787:
Changing to new event H245_PROCESS_H245CONTROL *Jun 14
15:35:01.787: cch323_h245_connection_sm:H245_CONNECT:
received event H24 5_PROCESS_H245CONTROL while at
H245_CONNECTED state RAW_BUFFER::= 03 00000310 08B50000
14095433 38466178 55445000 90800300 0003 *Jun 14
15:35:01.787: PDU DATA = 61593960 value
MultimediaSystemControlMessage ::= request :
openLogicalChannel : ??? { ?????
forwardLogicalChannelNumber 4 ?????
forwardLogicalChannelParameters ????? { ?????? dataType
data : ?????? { ?????? application nonStandard :
???????? { ????????? nonStandardIdentifier
h221NonStandard : ?? ????????? { ?????????
t35CountryCode 181 ????????? t35Extension 0
???????? manufacturerCode 20 ????????? }
???????? data '543338466178554450'H ????????? }
???????? maxBitRate 144 ?????? } ??????
```

```

multiplexParameters h2250LogicalChannelParameters :
??????? { ????????? sessionID 3 ???????? } ????? } ??? }
*Jun 14 15:35:01.831: Changing to new event
H245_PROCESS_H245CONTROL *Jun 14 15:35:01.831:
cch323_h245_connection_sm:H245_CONNECT: received event
H24 5_PROCESS_H245CONTROL while at H245_CONNECTED state
RAW_BUFFER:= 23 800001 *Jun 14 15:35:01.831: PDU DATA =
61593960 value MultimediaSystemControlMessage ::=
response : closeLogicalChannelAck : ??? { ?????
forwardLogicalChannelNumber 2 ??? } *Jun 14
15:35:01.883: Changing to new event
H245_PROCESS_H245CONTROL *Jun 14 15:35:01.883:
cch323_h245_connection_sm:H245_CONNECT: received event
H24 5_PROCESS_H245CONTROL while at H245_CONNECTED state
RAW_BUFFER:= 22 C0000204 800C5804 00875C34 CB1B4801
0100 *Jun 14 15:35:01.883: PDU DATA = 61593960 value
MultimediaSystemControlMessage ::= response :
openLogicalChannelAck : ??? { ?????
forwardLogicalChannelNumber 3 ?????
forwardMultiplexAckParameters
h2250LogicalChannelAckParameters : ????? { ????????
sessionID 3 ???????? mediaChannel unicastAddress :
IPAddress : ???????? { ????????? network '875C34CB'H
????????? tsapIdentifier 6984 ???????? } ????????
flowControlToZero FALSE ????? } ??? } *Jun 14
15:35:01.887: Changing to new event H245_EVENT_OLC_CFM
*Jun 14 15:35:01.887: cch323_h245_olc_sm: received event
H245_EVENT_OLC_CFM while at state H245_OLC_WAIT *Jun 14
15:35:01.887: changing from H245_OLC_WAIT state to
H245_OLC_DONE state cch323_h245_local_codec_dnld_done:
negotiatedCodec[17] *Jun 14 15:35:01.979: Changing to
new event H245_EVENT_OLC_IND *Jun 14 15:35:01.979:
cch323_h245_olc_sm: received event H245_EVENT_OLC_IND
while at state H245_OLC_DONE !---Session ID was sent as
voice session ID, fallback to voice and the call
disconnected: PDU DATA = 61593960 value
MultimediaSystemControlMessage ::= response :
openLogicalChannelAck : ??? { ?????
forwardLogicalChannelNumber 4 ?????
forwardMultiplexAckParameters
h2250LogicalChannelAckParameters : ????? { ????????
sessionID 1 ???????? mediaChannel unicastAddress :
IPAddress : ???????? { ??? ??????network 'C95C381E'H
????????? tsapIdentifier 18100 ???????? } ????????
mediaControlChannel unicastAddress : IPAddress : ????????
{ ????????? network 'C95C381E'H ?????????? tsapIdentifier
18101 ???????? } ???????? flowControlToZero FALSE ????? }
??? } RAW_BUFFER:= 22 C0000304 80145C00 00C95C38
1E46B400 C95C381E 46B50300 0100 *Jun 14 15:35:01.983:

```

このセクションでは、AS5300 シリーズ ルータと Cisco 3640 モジュラ アクセス ルータの間の成功する T.38 ファクス設定の分析を詳しく説明しています。debug コマンドおよび show コマンドの出力は、Cisco 3640 モジュラ アクセス ルータ上で TGW IOS 12.4 として debug vtsp all コマンド上でキャプチャされました。

debug vtsp all コマンド出力

```

Router# debug vtsp all

Voice telephony call control all debugging is on
!--- At this point, the VTSP is not aware of anything.
The format of this message is //callid/GUID/VTSP:(voice-
port):T1-channel_number:DSP_number:DSP_channel_number:

```

```
•CallEntry ID is -1. •GUID is xxxxxxxxxxxx. •The voice
port is blank. •Channel ID is -1. •DSP ID is -1. •DSP
channel ID is -1. *Mar 1 08:23:10.869: //-
1/xxxxxxxxxxxxx/VTSP:():-1:-1:-
1/vtsp_do_regxrule_translate: !--- The original and the
translated calling number are the same (55555) and the
original and the translated called number are the same
(888545). These numbers are often the same because if a
translation rule is applied, it will be on the dial
peers or the ports, both of which comes later than these
VTSP messages in the Cisco IOS code execution. *Mar 1
08:23:10.869: //-1/xxxxxxxxxxxxx/VTSP:():-1:-1:-1/vtsp
_do_regxrule_translate: calling_number(original)=
calling_number(xlated)=55555 called_number(original)=
called_number(xlated)=888545 redirectNumber(original)=
redirectNumber(xlated)= !--- The VTSP got a call setup
indicator from the TSP layer with called number 888545
and calling number 55555. There is no awareness of the
CallEntry ID (-1) or the GUID (xxxxxxxxxxxx). *Mar 1
08:23:10.873: //-1/xxxxxxxxxxxxx/VTSP:():-1:-1:-
1/vtsp_tsp_call_setup_ind: (sdb=0x634C90EC,
tdm_info=0x0, tsp_info=0x63083950, calling_number=55555
calling_oct3 = 0x80, called_number=888545 called_oct3 =
0x80, oct3a=0x0): peer_tag=10002 *Mar 1 08:23:10.873:
//-1/xxxxxxxxxxxxx/VTSP:():-1:-1:-
1/vtsp_tsp_fill_setup_ind : ev.clg.clir is 0
ev.clg.clid_transparent is 0 ev.clg.null_orig_clg is 0
ev.clg.calling_translated is false *Mar 1 08:23:10.873:
//-1/xxxxxxxxxxxxx/VTSP:():-1:-1:-
1/vtsp_do_call_setup_ind: . *Mar 1 08:23:10.873: //-
1/xxxxxxxxxxxxx/VTSP:():-1:-1:-1/vtsp_allocate_cdb: ,cdb
0x635FC480 *Mar 1 08:23:10.873: //-
1/xxxxxxxxxxxxx/VTSP:():-1:-1:-1/vtsp_do_call_setup_ind:
*Mar 1 08:23:10.873: source route label !--- At this
point, the VTSP is not aware of anything. The format of
this message is //callid/GUID/VTSP:(voice-port):T1-
channel_number:DSP_number:DSP_channel_number: •CallEntry
ID is -1. •GUID is D2F6429A8A8A. •The voice port is
1/0:23 where 23 indicates D channel. •The T1 channel is
still unknown at this point (-1). •The digital signal
processor (DSP) is 0. •The DSP channel is 4. *Mar 1
08:23:10.873: //-1/D2F6429A8A8A/VTSP:(1/0:23):-
1:0:4/vtsp_do_call_setup_ind: Call ID=101002,
guid=635FCB08 !--- The VTSP learns about the B channel
(changed from -1 to 22), and the CallEntry ID is still
unknown (-1). *Mar 1 08:23:10.873: //-
1/D2F6429A8A8A/VTSP:
(1/0:23):22:0:4/vtsp_do_call_setup_ind: type=0,
under_spec=1615186336, name=, id0=23, id1=0, id2=0,
calling=55555,called=888545
subscriber=RegularLinevtsp_do_call_setup_ind: redirect
DN = reason = -1 *Mar 1 08:23:10.877: //-
1/xxxxxxxxxxxxx/VTSP:():-1:-1:-
1/vtsp_do_normal_call_setup_ind: . !--- The VTSP learns
the CallEntry ID. The format of this message is
//callid/GUID/VTSP:(voice-port):T1-
channel_number:DSP_number:DSP_channel_number: •CallEntry
ID is 899 (changed from -1 to 899) •GUID is D2F6429A8A8A
•The voice port is 1/0:23 where 23 indicates D channel
•The T1 channel is 22 •The DSP is 12 •The DSP channel is
4 *Mar 1 08:23:10.877: //899/D2F6429A8A8A/VTSP:(1/0:23)
:22:12:4/vtsp_insert_cdb:,cdb 0x635FC480, CallID=899
*Mar 1 08:23:10.877:
//899/D2F6429A8A8A/VTSP:(1/0:23):22:12:4/vtsp_open_voice
```

```
_and_set_params: . !--- In these outputs, VTSP sets some
of the voice parameters for this call: •Modem capability
•Playout delay •Dial-peer tag 10003 •Digit timeouts *Mar
1 08:23:10.877: //899/D2F6429A8A8A/VTSP:(1/0:23):
22:12:4/vtsp_modem_proto_from_cdb: cap_modem_proto 0
*Mar 1 08:23:10.881:
//899/D2F6429A8A8A/VTSP:(1/0:23):22:12:4/set_playout_cdb
:playout default *Mar 1 08:23:10.881:
//899/D2F6429A8A8A/VTSP:(1/0:23):22:12:4/vtsp_dsp_echo_c
anceller_control: echo_cancel: 1 *Mar 1 08:23:10.885:
//899/D2F6429A8A8A/VTSP:
(1/0:23):22:12:4/vtsp_save_dialpeer_tag: tag = 10003
*Mar 1 08:23:10.885: //899/D2F6429A8A8A/VTSP:
(1/0:23):22:12:4/vtsp_report_digit_control:
vtsp_report_digit_control: enable=0: *Mar 1
08:23:10.885: //899/D2F6429A8A8A/VTSP:(1/0:23):
22:12:4/vtsp_report_digit_control: digit reporting
disabled *Mar 1 08:23:10.885:
//899/D2F6429A8A8A/VTSP:(1/0:23):
22:12:4/vtsp_get_digit_timeouts: :
vtsp_get_digit_timeouts !--- VTSP sends out a call-
proceeding message to the POTS leg *Mar 1 08:23:10.885:
//899/D2F6429A8A8A/VTSP:(1/0:23):22:12:4/vtsp_process_ev
ent:vtsp:[1/0:23:899, S_SETUP_INDICATED,
E_CC_PROCEEDING] *Mar 1 08:23:10.885:
//899/D2F6429A8A8A/VTSP:(1/0:23):22:12:4/act_proceeding:
. *Mar 1 08:23:10.941: //899/D2F6429A8A8A/VTSP:
(1/0:23):22:12:4/vtsp_get_dialpeer_tag: tag = 10003 *Mar
1 08:23:10.949: //899/D2F6429A8A8A/VTSP:(1/0:23):
22:12:4/vtsp_get_dialpeer_tag: tag = 10003 !--- VTSP
sends out an alerting to the POTS leg; the phone is
ringing at this time. *Mar 1 08:23:10.949:
//899/D2F6429A8A8A/VTSP:
(1/0:23):22:12:4/vtsp_process_event: vtsp:[1/0:23:899,
S_PROCEEDING, E_CC_ALERT] *Mar 1 08:23:10.949:
//899/D2F6429A8A8A/VTSP:(1/0:23):22:12:4/act_alert: .
*Mar 1 08:23:10.949:
//899/D2F6429A8A8A/VTSP:(1/0:23):22:12:4/vtsp_timer_stop
:3019095 *Mar 1 08:23:18.769:
//899/D2F6429A8A8A/VTSP:(1/0:23):
22:12:4/vtsp_get_dialpeer_tag: tag = 10003 !--- The
phone gets answered here, a bridge is now set up between
the two call legs. *Mar 1 08:23:10.949:
//899/D2F6429A8A8A/VTSP:
(1/0:23):22:12:4/vtsp_process_event: vtsp:[1/0:23:899,
S_PROCEEDING, E_CC_ALERT] *Mar 1 08:23:10.949:
//899/D2F6429A8A8A/VTSP:(1/0:23):22:12:4/act_alert: .
*Mar 1 08:23:10.949:
//899/D2F6429A8A8A/VTSP:(1/0:23):22:12:4/vtsp_timer_stop
:3019095 *Mar 1 08:23:18.769:
//899/D2F6429A8A8A/VTSP:(1/0:23):
22:12:4/vtsp_get_dialpeer_tag: tag = 10003 !--- The call
is now connected. Mar 1 08:23:18.769:
//899/D2F6429A8A8A/VTSP:(1/0:23)
:22:12:4/vtsp_process_event: vtsp:[1/0:23:899,
S_ALERTING, E_CC_CONNECT] *Mar 1 08:23:18.769:
//899/D2F6429A8A8A/VTSP:(1/0:23):22:12:4/act_alert_conne
ct: . *Mar 1 08:23:18.773:
//899/D2F6429A8A8A/VTSP:(1/0:23):22:12:4/vtsp_ring_noan_
timer_stop: 3019877
```

- [VoIP でのファックス リレーの設定 \(T.38 \)](#)
- [ファックス リレートラブルシューティング ガイド](#)
- [Cisco AVVID ゲートウェイによるファックスリレーおよびファックス パススルーのサポート](#)
- [音声テレフォニーのサービス プロバイダーによるデバッグ](#)
- [ファックスに関するトラブルシューティング](#)
- [音声に関する技術サポート](#)
- [音声とユニファイド コミュニケーションに関する製品サポート](#)
- [Cisco IP Telephony のトラブルシューティング](#)
- [テクニカルサポートとドキュメント - Cisco Systems](#)