

# H.323 T.38 Fax Relay-Probleme

## Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Grundlagen zu T.38](#)

[T.38 Einschränkungen](#)

[T.38 Verhandlungen](#)

[T.38 Fehlerbehebung](#)

[Tipps zur Fehlerbehebung für H.323- oder SIP T.38-Fax-Relay](#)

[Befehle debuggen und anzeigen](#)

[Ausgabe eines erfolgreichen T.38-Anrufs](#)

[Ein Beispiel für einen fehlgeschlagenen T.38-Anruf](#)

[Zugehörige Informationen](#)

## Einführung

T.38-Fax-Relay-Probleme werden häufig mit Interoperabilitätsproblemen zwischen T.38-Fax-Gateways von Cisco und T.38-Drittanbietern in Verbindung gebracht. Dieses Dokument enthält detaillierte **Debug**-Befehlsbeispiele für erfolgreiche und erfolglose T.38-Faxweiterleitungsaufrufe. Diese **Debug**-Befehlsausgaben enthalten Kommentare, die Referenzpunkte enthalten, sodass Sie solche Interoperabilitätsprobleme identifizieren und beheben können. Entsprechende Befehle zur Fehlerbehebung und Überprüfung sind ebenfalls in diesem Dokument enthalten.

## Voraussetzungen

### Anforderungen

Die Leser dieses Dokuments sollten sich mit den grundlegenden Konzepten des Fax Relay auskennen. Weitere Informationen zu Fax Relay-Konzepten und grundlegenden Schritten zur Fehlerbehebung finden Sie im [Leitfaden](#) zur [Fehlerbehebung](#) für [Faxweiterleitungen](#).

### Verwendete Komponenten

Dieses Dokument ist nicht auf bestimmte Software- und Hardwareversionen beschränkt.

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie

die potenziellen Auswirkungen eines Befehls verstehen.

## Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie unter [Cisco Technical Tips Conventions](#) (Technische Tipps zu Konventionen von Cisco).

## Grundlagen zu T.38

Ein häufiges Symptom für T.38-Faxübertragungsprobleme ist ein Sprachanruf, bei dem ein Faxton ausgegeben wird, der jedoch nicht beendet wird und der Anruf schließlich verworfen wird. Dieses Problem ist häufig mit Interoperabilitätsproblemen des Cisco T.38-Gateways und der T.38-Gateways von Drittanbietern verbunden.

Das T.38-Fax-Relay ist eine Echtzeit-Faxübertragung, d. h. zwei Faxgeräte, die miteinander kommunizieren, als gäbe es eine direkte Telefonleitung zwischen den beiden. Fax Relay wird mit einigen zusätzlichen Befehlen für Gateway-DFÜ-Peers konfiguriert, die bereits für Sprachanrufe definiert und konfiguriert wurden.

Cisco bietet zwei Methoden für die Fax-Weiterleitung: eine proprietäre Methode von Cisco und eine auf dem ITU-T T.38-Standard basierende Methode. Auf den meisten Plattformen ist Cisco Fax Relay der Standardwert, wenn eine Faxmethode nicht explizit konfiguriert ist. Cisco Fax Relay wird unter [Konfigurieren des Cisco Fax Relay](#) beschrieben.

## T.38 Einschränkungen

Zu diesem Zeitpunkt gelten für das Cisco T.38-Fax-Relay folgende Einschränkungen:

- Für die T.38-Interoperabilität ist Cisco H.323 Version 2 erforderlich.
- T.38 wird auf Cisco Concentrators der Serie MC3810 mit einem Sprachkomprimierungsmodul (VCM) nicht unterstützt.
- T.38 wird vom MCM-Proxy (Multimedia Conference Manager) H.323 nicht unterstützt.
- Nur User Datagram Protocol (UDP) wird für H.323 T.38 implementiert.
- Einige Gateways und Gatekeeper von Drittanbietern sind nicht mit Cisco Voice Gateways für T.38-Fax-Relay kompatibel, da verschiedene Hersteller bestimmte Teile von H.323 und T.38 auswählen können, um diese in ihre Gateways und Gatekeeper zu implementieren. Tests der Sprachinteroperabilität mit diesen Gateways und Gatekeeper müssen durchgeführt werden, um sicherzustellen, dass das T.38-Fax-Relay erfolgreich sein kann.

## T.38 Verhandlungen

In diesem Abschnitt wird Schritt für Schritt zusammengefasst, wie die T.38-Aushandlung in Cisco Gateways gehandhabt wird. Weitere Informationen zu den Grundlagen der Faxweiterleitung finden Sie im [Leitfaden zur Fehlerbehebung](#) für [Fax-Relay](#).

1. In der ersten Setup-Meldung wird die T.38-Datenfähigkeit vom ursprünglichen Gateway (OGW) angekündigt. Wenn das Terminating Gateway (TGW) T.38-Datenfunktionen unterstützt, kann es diese Informationen in den folgenden, an den OGW gesendeten Nachrichten weitergeben.

2. Sobald ein Sprachanruf eingerichtet ist und der Digital Signal Processor (DSP) des TGW einen Faxton erkennt, informiert das Voice Telephony Service Provider (VTSP)-Statussystem den H.323-Anrufabschnitt, der den T.38-Modus mit dem OGW aushandelt.
3. Nach Bestätigung des T.38-Modus wird der Audiokanal geschlossen, und an beiden Enden wird der logische T.38-Kanal geöffnet.
4. Auf VTSP-Code-Ebene wird der Fax-Codierer-Decoder (Codec) heruntergeladen.
5. Nach erfolgreichem Download von T.38 Open Logical Channel (OLC) und Codec wechselt der VTSP in den Faxmodus.
6. Nach Beendigung der Faxübertragung wird der Anruf wieder zu einem Sprachanruf zurückgeleitet. **Hinweis:** Wenn während der Aushandlung des T.38-Modus am anderen Ende der T.38-Modus nicht bestätigt wird, wird der Anruf zurück zu einem Sprachanruf geleitet und die Verbindung getrennt. Wenn eine negative Bestätigung vom anderen Ende bezüglich der T.38-OLC eingeht, wird der Anruf ebenfalls zurück zu einem Sprachanruf geleitet und getrennt.

## T.38 Fehlerbehebung

### Tipps zur Fehlerbehebung für H.323- oder SIP T.38-Fax-Relay

Führen Sie folgende Schritte aus, um die Fehlerbehebung für das T.38-Fax-Relay durchzuführen:

- **Stellen Sie sicher, dass Sie einen Anruf tätigen können.** Vergewissern Sie sich, dass normale Sprachanrufe ausgeführt werden können, bevor Sie eine Faxverbindung herstellen. Wenn kein Telefon angeschlossen ist, trennen Sie das Faxgerät, und schließen Sie ein normales Telefon an. Wenn normale Sprachanrufe keine Verbindung herstellen, kann das Problem VoX-bezogen sein, und Sie können das Problem als normales Problem bei der Sprachverbindung beheben, bevor Sie mit der Faxbehebung fortfahren.
- Stellen Sie sicher, dass das gewünschte Faxprotokoll auf den Ausgangs- und den Terminierungs-Gateways mit dem Faxprotokoll-Befehl festgelegt wurde.
- Stellen Sie sicher, dass das Faxprotokoll auf globaler Konfigurationsebene oder auf DFÜ-Peer-Konfigurationsebene als T.38 konfiguriert ist, sowohl für die Ursprungs- als auch für die terminierenden Gateways.

### Befehle debuggen und anzeigen

Die Befehle **debug** und **show** zur Fehlerbehebung für das T.38-Fax-Relay sind:

- **debug voip ccapi inout** - Dieser Befehl verfolgt den Ausführungspfad über die API (Call Control Application Program Interface), die als Schnittstelle zwischen der Anrufsitungsanwendung und der zugrunde liegenden netzwerkspezifischen Software dient. Sie können die Ausgabe dieses Befehls verwenden, um zu verstehen, wie Anrufe vom Sprach-Gateway verarbeitet werden.
- **debug vtsp all** - Mit diesem Befehl werden die VTSP-Befehle **debug** aktiviert: **debug vtsp session**, **debug vtsp error** und **debug vtsp dsp**.
- **debug h245 asn1**: Dieser Befehl zeigt den Inhalt von ASN.1 (Abstract Syntax Notation One) von H.245-Nachrichten an. Um die Debugausgabe zu deaktivieren, verwenden Sie die no-Form dieses Befehls.

- **debug cch323 h245** - Dieser Befehl gibt die Ablaufverfolgung des Zustandsübergangs des H.245-Zustandscomputers basierend auf den verarbeiteten Ereignissen an. Um die Debugausgabe zu deaktivieren, verwenden Sie die no-Form dieses Befehls.
- **show call active fax brief**: Dieser Befehl zeigt Anrufinformationen für laufende Store-and-Forward-Faxübertragungen an.
- **show call history fax**: Dieser Befehl zeigt den letzten Anrufverlauf für Faxe an.

## Ausgabe eines erfolgreichen T.38-Anrufs

In diesem Abschnitt wird die Anatomie einer erfolgreichen T.38-Faxeinrichtung zwischen einem Router der Serie AS5300 und einem modularen Cisco 3640 Access Router beschrieben. Die **Debug**- und **Show**-Befehlsausgaben wurden auf dem Universal Gateway Cisco AS5300 als TGW IOS 12.2 erfasst:

### debug vtsp all-Befehlsausgabe

```

!---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)
```

## Ein Beispiel für einen fehlgeschlagenen T.38-Anruf

Dies ist ein Beispiel für die Ausgabe des **Debug**-Befehls für einen fehlgeschlagenen T.38-Aufruf:

### **debug vtsp all-Befehlsausgabe**

```
!---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 whil

e 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:

```

In diesem Abschnitt wird die Anatomie einer erfolgreichen T.38-Faxeinrichtung zwischen einem Router der Serie AS5300 und einem modularen Cisco 3640 Access Router beschrieben. Die **Debug**- und **Show**-Befehlsausgaben wurden mit dem Befehl `debug vtsp all` auf einem modularen Cisco 3640 Access Router als TGW IOS 12.4 erfasst:

### debug vtsp all-Befehlsausgabe

```

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/xxxxxxxxxxxxxxxx/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/xxxxxxxxxxxxxxxx/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/xxxxxxxxxxxxxxxx/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/xxxxxxxxxxxxxxxx/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/xxxxxxxxxxxxxxxx/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
```

## Zugehörige Informationen

- [Konfigurieren von Fax Relay mit VoIP \(T.38\)](#)
- [Leitfaden zur Fehlerbehebung für Fax-Relay](#)
- [Cisco AVVID Gateway-Unterstützung für Fax Relay und Fax Pass-Through](#)
- [Debugging für Sprachtelefonieanbieter](#)
- [Fehlerbehebung Fax](#)
- [Unterstützung von Sprachtechnologie](#)
- [Produkt-Support für Sprach- und Unified Communications](#)
- [Fehlerbehebung bei Cisco IP-Telefonie](#)
- [Technischer Support und Dokumentation - Cisco Systems](#)