

PGW 2200 Softswitch TCAP Version 9.3 und höher

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Hintergrundinformationen](#)

[TCAP-Auflösung](#)

[Sniffer-Ethernet-Leitung](#)

[Platform.log TCAP Trace](#)

[MDL Trace Tool](#)

[Anhang A: MDL-Tags](#)

[Anhang B: SS7-Punktcodes abmelden](#)

[Anhang C: SCCP-Nachrichtentypen](#)

[Unitdata \(UDT\)](#)

[Unitdata-Service \(UDTS\)](#)

[UDTS-Rückgabeursachen](#)

[Anhang D: MDL-Schnittstelle für TCAP-Nachricht](#)

[Anhang E: Interne MDL-Schnittstelle](#)

[Zugehörige Informationen](#)

Einführung

Transaction Capabilities Applications Part (TCAP) unterstützt interaktive Anwendungen in einer verteilten Umgebung. TCAP definiert ein End-to-End-Protokoll zwischen seinen Benutzern. Diese kann sich in einem SS7-Netzwerk oder einem anderen Netzwerk befinden, das TCAP (IP) unterstützt.

Voraussetzungen

Anforderungen

Die Leser dieses Dokuments sollten über Folgendes verfügen:

- [Cisco Media Gateway Controller Version 9](#)

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf dem Cisco PGW 2200 Softswitch.

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 in den [Cisco Technical Tips Conventions](#).

Hintergrundinformationen

Das TCAP-Protokoll besteht aus zwei Unterschichten:

- Komponenten-Unterschicht
- Transaktions-Unterschicht

Die Komponentenunterschicht ist mit der Konvertierungsengine verbunden. Das Konvertierungsmodul entspricht einer Service-Benutzer- oder Subsystemnummer (SSN). Die Komponenten-Unterschicht unterstützt folgende Services:

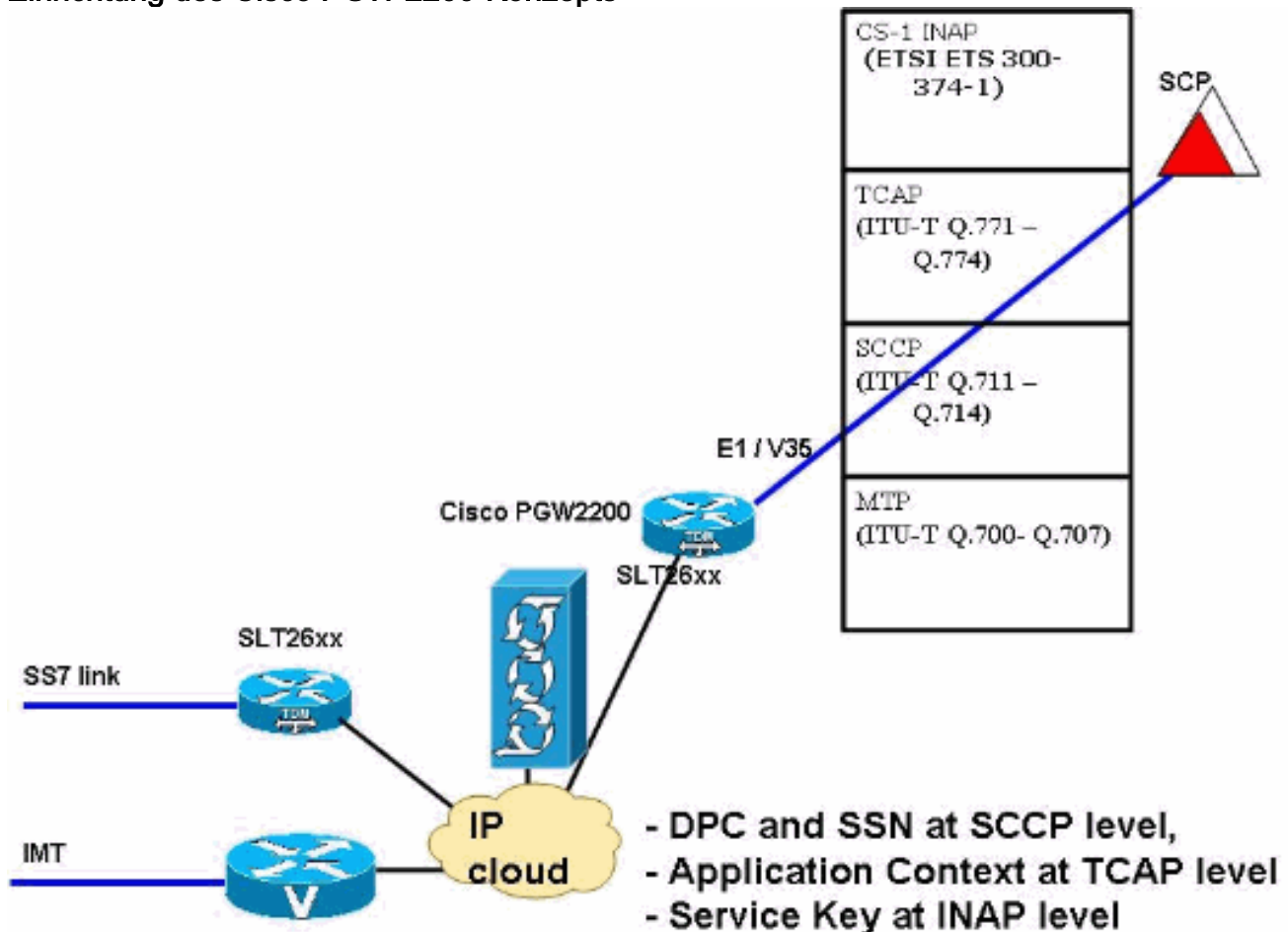
- Zuordnung von Operationen und Antworten.
- Ungewöhnliche Handhabung von Situationen.

Die Transaktions-Subschicht ist mit SCCP (Signaling Connection Control Part) verbunden. TCAP unterstützt nur einen verbindungslosen Netzwerkdienst. Die Transaktions-Subschicht kommuniziert über die verbindungslose Schnittstelle mit dem SCCP.

Die TCAP-Software nutzt die Dienste der SCCP-Software, um die Nachrichten an den TCAP-Benutzer im Zielknoten weiterzuleiten. Die Schnittstelle zwischen TCAP und der SCCP-Software ist eng miteinander verknüpft. Jede TCAP-Anfrage vom Motor enthält eine globale Titel- und Zielsubsystem-Nummer. TCAP stellt dem SCCP die Subsystemnummer für die Suche nach dem STP-Code (Signal Transfer Points) zur Verfügung. Wenn die SS7-Adressen und -Routen korrekt und vollständig betriebsbereit konfiguriert sind, führen Sie eine Fehlerbehebung für die SCCP- und TCAP-Informationen durch, die zwischen dem Cisco PGW 2200 und einem Remote-SCCP- oder TCAP-Peer weitergeleitet und empfangen wurden.

Der Cisco PGW 2200 kapselt mithilfe des SCCP TCAP-Abfragen für das Versenden von MTP (Message Transfer Part). Diese SCCP-Kommunikation zwischen Peers wird ohne Verbindung über MTP gesendet. Der Cisco PGW 2200 verwendet SCCP Unidata (UDT), um Daten zur verbindungslosen Kommunikation an den entfernten SCCP-Knoten zu senden. Der PGW 2200 erhält eine gültige Antwort, wenn die SCCP-UDT-Nachricht erfolgreich zugestellt wurde. Dies geschieht in der Regel in Form einer UDT-Nachricht. Der Austausch dieser UDT-Nachrichten erleichtert die verbindungslose Kommunikation zwischen dem PGW 2200 und dem Remote-SCCP-Peer (z. B. dem Service Control Point [SCP] für TCAP-Datenbankabfragen). Der PGW 2200 definiert ein optionales Feld im UDT, in dem festgelegt wird, dass der SCCP-Peer den Inhalt einer Nachricht, die er an den Remote-Knoten sendet, "bei Fehler zurückgeben" soll, wenn die UDT-Nachricht nicht zugestellt werden kann. Die Unidata Service (UDTS)-Nachricht wird verwendet, um diese Fehlerantwort zu vereinfachen. Die UDTS-Meldung weist an den PGW 2200 darauf hin, dass eine am Remote-Knoten empfangene UDT-Nachricht (z. B. STP oder SCP) nicht an das Ziel übermittelt werden kann.

Einrichtung des Cisco PGW 2200-Konzepts



TCAP-Auflösung

Das SCCP-Messaging (UDT/UDTS), das im Abschnitt [Hintergrundinformationen](#) behandelt wird, ist bei der Fehlerbehebung für TCAP-Services und -Funktionen von kritischer Bedeutung. Lösen Sie alle Probleme auf der SCCP-Ebene, bevor Sie die Fehler bei gesendeten oder empfangenen TCAP-Daten beheben. Das Format der UDT- und UDTS-Nachricht ist in [Anhang C](#) dargestellt.

Verwenden Sie die folgenden Cisco PGW 2200-Tools, um Anrufe zu debuggen, die die TCAP-Services (TCAP/SCCP) erfordern:

- [Schalten Sie die Ethernet-Leitung](#) mit Tools wie Ethereal, UNIX Snoop und Snoop aus.
- [Platform.log TCAP-Trace](#) auf dem PGW 2200.
- [MDL Trace Tool](#) für die Anrufverarbeitung auf Protokollebene.

Sniffer-Ethernet-Leitung

Der Cisco PGW 2200 verwendet Reliable UDP (RUDP), um MTP3- und SS7-Nachrichten auf höherer Ebene zwischen den lokalen MTP1- und MTP2-Geräten (z. B. einem Signaling Link Terminal [SLT]) zu senden. Diese Kommunikation erfolgt in der Regel über Port 700 an der lokalen Ethernet-Schnittstelle des Cisco PGW 2200. Diese Option kann konfiguriert werden. Weitere Informationen zur Konfiguration der PGW "stPort"-Ports in XECfgParm.dat finden Sie im [Konfigurationsleitfaden](#).

Sie können einen beliebigen Ethernet-Sniffer verwenden, um die zwischen dem Cisco PGW 2200

und seinem lokalen MTP2-Steuergerät gesendeten Pakete anzuzeigen. Allerdings unterstützen nicht alle das MTP- und SCCP-Protokoll, das zum Anzeigen einer dekodierten Nachricht verwendet wird. Wenn dem Kunden kein Ethernet-Sniffer zur Verfügung steht, führen Sie die Fehlerbehebung mit dem Befehl UNIX **snoop** durch. Die Ausgabe des Befehls **snoop** ist nicht benutzerfreundlich, aber im schlimmsten Fall hilfreich.

Ein Ethernet-Sniffer, der den SS7-Protokollstapel unterstützt, wird bevorzugt. Mit dieser Funktion können Sie die auf der Cisco PGW 2200 Ethernet-Schnittstelle erkannten Pakete dekodieren. Ein Open-Source-Sniffer wie [Ethereal](#) kann ebenfalls verwendet werden und ist online verfügbar.

Wenn kein kommerzielles Sniffer-Dienstprogramm verfügbar ist, führen Sie den Befehl **snoop** auf dem Ziel-Cisco PGW 2200 aus, um die Hex-Datenausgabe der Nachrichten anzuzeigen, die an den und vom Cisco PGW 2200 gesendet wurden. Geben Sie mit der Root-Berechtigung auf dem Cisco PGW 2200 diesen Befehl aus, um die aus dem konfigurierten 'stPort' gesendeten Hexadezimaldaten anzuzeigen. Weitere Informationen zum Befehl **snoop** finden Sie in den 'snoop man pages' oder in den SUN Administrative Guides.

```
#snoop -d
```

Geben Sie diesen Befehl ein, um die Pakete zu snoop, die vom Ethernet-Gerät "hmeX" an Port 7000 gesendet wurden.

```
#snoop -d hmeX -x 42 port 7000
```

Dies ist die Beispielausgabe von erfassten SS7-Paketen mit dem Befehl **snoop**.

```
#snoop -d hme0 -x 42 port 7000
```

```
PGW2200 -> C2600.cisco.com UDP D=7000 S=7000 LEN=96
```

```
0: 4004 dcb5 0000 8000 0001 0000 0010 0000 @.....
```

```
16: 0000 0044 8321 4802 3209 8003 0d11 0a8b ...D.!H.2..... ← UDT (09) to SLT from PGW
```

```
32: 2108 3000 1838 3344 4404 c309 0865 2962 !.0.83DD....e)b
```

```
48: 2748 0102 6c22 a120 0201 0102 0100 3018 'H.I". .....0.
```

```
64: 8004 0000 0001 8207 0110 1838 3344 4483 .....83DD.
```

```
80: 0701 1107 1311 0010 .....  
.....
```

```
PGW2200 -> C2600.cisco.com UDP D=7000 S=7000 LEN=32
```

```
0: 4004 ddb5 0000 8000 0001 0000 0044 0000 @.....D..
```

```
16: 0000 0004 0000 0001 .....  
.....
```

```
C2600.cisco.com -> PGW2200 UDP D=7000 S=7000 LEN=144
```

```
0: 4004 b6dd 0000 8000 0001 0001 0045 0000 @.....E..
```

```
16: 0000 0074 0000 001e 0000 0000 0000 0000 ...t.....
```

```
32: 0000 0000 0000 0000 0000 0000 0000 0000 .....  
.....
```

```
48: 0000 0000 0000 0000 0000 0000 0000 0000 .....  
.....
```

```
0: 4004 b6dd 0000 8000 0001 0001 0045 0000 @.....E..
```

```
16: 0000 0074 0000 001e 0000 0000 0000 0000 ...t.....  
.....
```

```

16: 0000 0074 0000 001e 0000 0000 0000 0000  ...t.....
32: 0000 0000 0000 0000 0000 0000 0000 0000  .....
48: 0000 0000 0000 0000 0000 0000 0000 0000  .....
64: 0000 0000 0000 0003 0000 0000 0000 8571  .....q
80: 0000 0000 0000 0002 0000 0000 0000 000a  .....
96: 684f 3338 0000 0000 22b3 e70f 0003 598a  hO38....".....Y.
112: 0000 0001 0000 0000 0000 0000 0000 0000  .....
128: 0000 0000 0000 0005  .....

```

PGW2200 -> C2600.cisco.com UDP D=7000 S=7000 LEN=12

```
0: 4004 deb6 @...
```

C2600.cisco.com -> PGW2200 UDP D=7000 S=7000 LEN=96

```
0: 4004 b7dd 0000 8000 0001 0000 0011 0000 @.....
```

```
16: 0000 0044 8309 4808 a20a 0103 0d11 04c3  ...D.H..... ← UDTS (0A) from SLT to PGW
```

```
32: 0908 650a 8b21 0830 0018 3833 4444 2962  ..e.!0.83DD)b
```

```
48: 2748 0102 6c22 a120 0201 0102 0100 3018  'H.1". .....0.
```

```
64: 8004 0000 0001 8207 0110 1838 3344 4483  .....83DD.
```

```
80: 0701 1107 1311 0010  .....
```

Der Cisco Snooper kann auch verwendet werden (falls verfügbar), um den Hex-Dump der SCCP-Nachricht anzuzeigen. Der SCCP-Nachrichtenheader ist dekodiert, aber die Anzeige der Ausgabe hängt von der gewählten Snooper-Version ab. Der wichtige Punkt ist, dass der Meldungstyp sichtbar ist und angibt, wo die Fehlerbehebung für den Anrufverlauf gestartet werden soll. Das Hexadezimalsystem zeigt an, dass der Meldungstyp 09 eine UDT-Nachricht ist und der Meldungstyp 0a die UDTS-Dienstmeldung ist, die auf einen Fehler hinweist. Die Richtung des Nachrichtenflusses ist ebenfalls nützlich, da die SS7-PCs angezeigt werden. Wenn der Rest des Hex-Dump angezeigt wird (abhängig von der Snooper-Version), kann er verwendet werden, um die SCCP- und TCAP-Teile der Nachricht weiter zu decodieren. Dies basiert auf den Branchenstandards für SCCP und TCAP.

Dies ist die Snooper-Ausgabe der UDT SCCP-Nachricht mit TCAP-Daten (an das PSTN).

```

15:23:03.847052 1-001-1[02057] 1-004-1[02081] ITU SCCP. -> UDT (09) CGPA=0103TCAPMsgType= Pr:0 Ni:NTL

09 80 03 07 0b 04 c3 21 08 0c 04 c3 09 08 67 52 .....!.....gR
62 50 48 01 1f6b 22 28 20 06 07 00 11 86 05 01 bPH..k"( .....
01 01 a0 15 60 13 80 02 07 80 a1 0d 06 0b 2a 81 ....^.....*.
76 82 15 01 01 01 01 00 01 6c 27 a1 25 02 01 01 v.....f.%...
02 01 00 30 1d 80 04 00 01 5f91 82 08 83 10 65 ...0....._.....e
27 32 54 76 0f83 07 03 11 03 23 22 11 11 9a 02 '2Tv.....#* ....
20 00

```

Wenn eine nicht zustellbare SCCP-UDT-Nachricht vom Cisco PGW 2200 gesendet wird und/oder ein SCCP (auf dem Remote-Knoten) Probleme mit der Nachricht hat, erhält der Cisco PGW 2200 eine UDTS-Antwortmeldung. Diese Meldung weist auf eine "Rückgabeursache" hin, die bei der Fehlerbehebung sehr nützlich ist. Der UDTS ist der Meldungstyp 10 (oder 0a hex).

Dies ist ein Beispiel für eine UDTS-SCCP-Nachricht mit TCAP-Daten (vom PSTN).

Hinweis: Diese Nachricht ist nur ein Beispiel und spiegelt möglicherweise keine tatsächliche Abfrageantwortkombination/-sequenz wider. Das Format und die Anzahl der angezeigten Informationen variieren je nach Snooper-Version.

```

15:23:04.952706 1-004-1[02081] 1-001-1[02057] ITU SCCP. -> UDTS (0a) CGPA=0012TCAPMsgType=0a
Pr:0 Ni:NTL

0a 01 03 0d 11 04 c3 09 08 65 0a 8b 21 08 30 00 .....g.!..v
18 38 33 44 44 29 62 27 48 01 03 6c 22 a1 20 02 etH.P...l.k*{(
01 01 02 01 00 30 18 80 04 00 00 00 01 82 07 01 .....a.....
10 18 38 33 44 55 83 07 01 11 07 13 11 00 10 *.v.....

```

Diese Snooper-Ausgabe zeigt die IAM-, UDT-, UDTS- und REL-Sequenz an.

Hinweis: Diese Nachricht ist nur ein Beispiel und spiegelt möglicherweise keine tatsächliche Abfrageantwortkombination/-sequenz wider. Das Format und die Anzahl der angezeigten Informationen variieren je nach Snooper-Version.

```

10:49:37.940189 1-022-1[02225] 1-001-1[02057] ITU ISUP.-> IAM(01) CIC=00010 CDPN=8183334444 CGPN=7031110001
SLS=00 Pr:0 Ni:NTL

10:49:37.962583 1-001-1[02057] 1-004-1[02081] ITU SCCP.-> UDT (09) CGPA=0101TCAPMsgType=
Pr:0 Ni:NTL

10:49:38.034121 1-004-1[02081] 1-001-1[02057] ITU SCCP.-> UDTS (0a) CGPA=0068TCAPMsgType=
Pr:0 Ni:NTL

10:49:38.052539 1-001-1[02057] 1-022-1[02225] ITU ISUP.-> REL (0c) CIC=00010 Cause 31 = Normal, Unspecified
SLS=00 Pr:0 Ni:NTL

```

Dies ist ein SS7-Sniffer-Trace, der SS7 SCCP- und TCAP-Informationen enthält.

```

-----
SCP(IN)- 19/03/04 18:01:54:223      SCCP      SCP(IN)  UDT      SCP(IN)  BGN  INVK  IDP
-----
Octet001  ITU-T SS7                          Time=19/03/02 18:01:54:223
-----
11010011  BIB/BSN                               1/83
10010110  FIB/FSN                               1/22
..111111  SU type/length                        MSU63
00.....  Spare                                  0
-----
Octet004  Service information octet
-----
....0011  Service indicator                     SCCP Signalling Connection Control Part
..00....  Message priority                       0
10.....  Network indicator                     N National network
-----
Octet005  Routing label
-----
.....    DPC                                10337 SCP(IN)
.....    OPC                                10321
0001....  SLS                                    1
-----
Octet009  Message type
-----
00001001  Message type                          UDT  Unitdata
-----
Octet010  SCCP Protocol Class parameter
-----
....0001  Protocol class                         Class 1
0000....  Message handling                       No special options
00000011  Ptr -> Called number                   3
00000111  Ptr -> Calling #                       7
00001011  Pointer -> Data                         11
-----
Octet014  SCCP Called Party Address parameter
-----
00000100  Parameter length                       4
.....1   Sgnl pt code bit                       SPC present
.....1   Subsystem # bit                         SSN present
..0000..  Global title ind                       No global title included
.1.....  Routing bit                             DPC and SSN based routing
0.....  Reserved natl use                       0

```



```

..... Point code          10337 SCP(IN)
00..... Spare            0
11111100 Subsystem number  INAP      IN-CS1+
-----
Octet019  SCCP Calling Party Address parameter
-----
00000100 Parameter length    4
.....1  Sgnl pt code bit   SPC present
.....1. Subsystem # bit   SSN present
..0000.. Global title ind  No global title included
.1..... Routing bit      DPC and SSN based routing
0..... Reserved natl use  0
..... Point code         10321
00..... Spare            0
11111100 Subsystem number  INAP      IN-CS1+
-----
Octet024  SCCP Data parameter
-----
01100001 Parameter length    97
01100010 Tag                BGN Begin, constructor, application-wide
01011111 Length            95
-----
Octet027  Originating Transaction ID
-----
...01000 Tag                Originating Transaction ID
010..... Class and form    Application-wide, primitive
00000011 Length            3
..... Originating ID      F30051
-----
Octet032  TCAP Dialogue Portion
-----
...01011 Tag                TCAP Dialogue Portion
011..... Class and form    Application-wide, constructor
00100011 Length            35
-----
Octet034  TCAP External
-----
...01000 Tag                TCAP External
001..... Class and form    Universal, constructor
00100001 Length            33
-----
Octet036  Object identifier
-----
...00110 Tag                Object identifier
000..... Class and form    Universal, primitive
00000111 Length            7
00000000 Organization      itu-t recommendation
00010001 q                  Q
..... 773 (X'305)         773
00000001 as(1)             1
00000001 Protocol data unit dialogue PDU(1)
00000001 version(1)        1
10100000 Single-ASN.1-typeTag Parameter
00010110 Length            22
-----
Octet047  Dialogue request
-----
...00000 Tag                Dialogue request
011..... Class and form    Application-wide, constructor
00010100 Length            20
-----
Octet049  Protocol-version
-----
...00000 Tag                Protocol-version

```

100.....	Class and form	Context-specific, primitive
00000010	Length	2
00000111	Unused Bit	07
.0000000	Unused Bit	00
1.....	Protocol Version	Version 1

Octet053	Application-context-name	

...00001	Tag	Application-context-name
101.....	Class and form	Context-specific, constructor
00001110	Length	14

Octet055	Object Identifier	

...00110	Tag	Object identifier
000.....	Class and form	Universal, primitive
00001100	Length	12
00101010	Protocol	ccitt identified-organization
10000110	SubProtocol	etsi
00111010	Domain	inDomain
00000000	Network	in-Network
10001001	AC Name	ac (application context)
01100001	Service	csl-ssp-to-scp(0)
00110011	Version	Reserved
.....	Contents	01 00 01 00 01

Octet069	TCAP Component Portion	

...01100	Tag	TCAP Component Portion
011.....	Class and form	Application-wide, constructor
10000000	Length	128

Octet071	Invoke component	

...00001	Tag	Invoke component
101.....	Class and form	Context-specific, constructor
00101111	Length	47

Octet073	Invoke ID	

...00010	Tag	Invoke ID
000.....	Class and form	Universal, primitive
00000001	Length	1
00000001	Invoke ID	01

Octet076	Operation Code	

...00010	Tag	Local
000.....	Class and form	Universal, primitive
00000001	Length	1
.....	Operation Code	IDP InitialDP

Octet079	Parameter Sequence	

...10000	Tag	Parameter Sequence
001.....	Class and form	Universal, constructor
00100111	Length	39

Octet081	ServiceKey	

...00000	Tag	ServiceKey
100.....	Class and form	Context-specific, primitive
00000001	Length	1
.....	Service key	94

```

-----
Octet084 CalledPartyNumber
-----
...00010 Tag CalledPartyNumber
100..... Class and form Context-specific, primitive
00000111 Length 7
.0000011 Nature of address National (significant) number( national use )
1..... Odd/even Odd number of address signals
....0000 Spare 00
.001.... Numbering plan ISDN (Telephony) numbering plan (Rec. E.164)
1..... Internal network # Routing to internal network number not allowed
..... Address signals 999956738
0000.... Filler 0
-----
Octet093 CallingPartyNumber
-----
...00011 Tag CallingPartyNumber
100..... Class and form Context-specific, primitive
00000111 Length 7
.0000011 Nature of address National (significant) number( national use )
1..... Odd/even Odd number of address signals
.....01 Screening Indicator User provided, verified and passed
....00.. Presentation? Presentation allowed
.001.... Numbering plan ISDN (Telephony) numbering plan (Rec. E.164)
0..... Number Incomplete? Complete
..... Address signals 2199997137
0000.... Filler 0
-----
Octet102 CallingPartysCategory
-----
...00101 Tag CallingPartysCategory
100..... Class and form Context-specific, primitive
00000001 Length 1
00001010 CallngPartyCategory Ordinary calling subscriber
-----
Octet105 ForwardCallIndicators
-----
...11010 Tag ForwardCallIndicators
100..... Class and form Context-specific, primitive
00000010 Length 2
.....0 Nat'l/International Call to be treated as a national call
....00. End-to-end method No end-to-end method available
...1... Interworking Interworking encountered
...0.... End-to-end info No end-to-end information available
..1..... ISUP indicator ISDN user part used all the way
01..... ISUP preference ISDN user part not required all the way
.....1 Orig ISDN access Originating access ISDN
....00. SCCP method No indication
...0... Spare 0
0000.... ReservedForNat'lUse 0
-----
Octet109 BearerCapability
-----
...11011 Tag BearerCapability
101..... Class and form Context-specific, constructor
00000101 Length 5
-----
Octet111 Bearer Cap
-----
...00000 Tag Bearer Cap
100..... Class and form Context-specific, primitive
-----
Octet112 User service information parameter
-----

```

```

00000011 Parameter length      3
-----
Octet113 User service info octet 3
-----
...00000 Transfer capability    Speech
.00..... Coding standard      CCITT standardized coding
1..... Extension bit          1
-----
Octet114 User service info octet 4
-----
...10000 Transfer rate          64 kbit/s
.00..... Transfer mode        circuit mode
1..... Extension bit          1
-----
Octet115 User service info octet 5
-----
...00011 Layer 1 protocol      Recommendation G.711 A-law
.01..... Layer 1 Identifier    User information layer 1 protocol
1..... Extension bit          1
-----
Octet116 CalledPartyNumber
-----
...00010 Tag                    CalledPartyNumber
110..... Class and form        Private use, primitive
00000010 Length                2
.0000000 Nature of address     Spare
0..... Odd/even               Even Number of Address signals
....1010 Spare                 0A
.000.... Numbering plan        Spare (no interpretation)
0..... Internal network #      Routing to internal network number allowed
-----
Octet120 End-of-contents
-----
00000000 Tag                    00
00000000 Length                00
-----
Checksum CRC16..... 0001011001110111 hex=1677
-----
-----
-----

```

SCP(IN)- 19/03/04 18:01:54:269 SCCP SCP(IN) UDT SCP(IN) CON INVK CUE

```

-----
Octet001 ITU-T SS7              Time=19/03/02 18:01:54:269
-----
10000001 BIB/BSN                1/1
10110010 FIB/FSN                1/50
..111111 SU type/length         MSU63
00..... Spare                  0
-----
Octet004 Service information octet
-----
...0011 Service indicator       SCCP Signalling Connection Control Part
..00.... Message priority       0
10..... Network indicator      N National network
-----
Octet005 Routing label
-----
..... DPC                       10321
..... OPC                       10337 SCP(IN)
1010.... SLS                    10
-----
-----

```

```

Octet009 Message type
-----
00001001 Message type          UDT   Unitdata
-----
Octet010 SCCP Protocol Class parameter
-----
....0001 Protocol class          Class 1
0000.... Message handling       No special options
00000011 Ptr -> Called number    3
00000111 Ptr -> Calling #       7
00001011 Pointer -> Data        11
-----
Octet014 SCCP Called Party Address parameter
-----
00000100 Parameter length        4
.....1 Sgnl pt code bit         SPC present
.....1. Subsystem # bit         SSN present
..0000.. Global title ind       No global title included
.1..... Routing bit            DPC and SSN based routing
0..... Reserved natl use       0
..... Point code                10321 Matinha
00..... Spare                   0
11111100 Subsystem number       INAP      IN-CS1+
-----
Octet019 SCCP Calling Party Address parameter
-----
00000100 Parameter length        4
.....1 Sgnl pt code bit         SPC present
.....1. Subsystem # bit         SSN present
..0000.. Global title ind       No global title included
.1..... Routing bit            DPC and SSN based routing
0..... Reserved natl use       0
..... Point code                10337 SCP(IN)
00..... Spare                   0
11111100 Subsystem number       INAP      IN-CS1+
-----
Octet024 SCCP Data parameter
-----
01001001 Parameter length        73
01100101 Tag                    CON Continue, constructor, application-wide
01000111 Length                  71
-----
Octet027 Originating Transaction ID
-----
...01000 Tag                    Originating Transaction ID
010..... Class and form         Application-wide, primitive
00000011 Length                  3
..... Originating ID           7A01B4
-----
Octet032 Destination Transaction ID
-----
...01001 Tag                    Destination Transaction ID
010..... Class and form         Application-wide, primitive
00000011 Length                  3
..... Destination ID           F30051
-----
Octet037 TCAP Dialogue Portion
-----
...01011 Tag                    TCAP Dialogue Portion
011..... Class and form         Application-wide, constructor
00101111 Length                  47
-----
Octet039 TCAP External
-----

```

...01000	Tag	TCAP External
001.....	Class and form	Universal, constructor
00101101	Length	45

Octet041	Object identifier	

...00110	Tag	Object identifier
000.....	Class and form	Universal, primitive
00000111	Length	7
00000000	Organization	itu-t recommendation
00010001	q	Q
.....	773 (X'305)	773
00000001	as(1)	1
00000001	Protocol data unit	dialogue PDU(1)
00000001	version(1)	1
10100000	Single-ASN.1-typeTag	Parameter
00100010	Length	34

Octet052	Dialogue response	

...00001	Tag	Dialogue response
011.....	Class and form	Application-wide, constructor
00100000	Length	32

Octet054	Protocol-version	

...00000	Tag	Protocol-version
100.....	Class and form	Context-specific, primitive
00000010	Length	2
00000111	Unused Bit	07
.0000000	Unused Bit	00
1.....	Protocol Version	Version 1

Octet058	Application-context-name	

...00001	Tag	Application-context-name
101.....	Class and form	Context-specific, constructor
00001110	Length	14

Octet060	Object Identifier	

...00110	Tag	Object identifier
000.....	Class and form	Universal, primitive
00001100	Length	12
00101010	Protocol	ccitt identified-organization
10000110	SubProtocol	etsi
00111010	Domain	inDomain
00000000	Network	in-Network
10001001	AC Name	ac (application context)
01100001	Service	csl-ssp-to-scp(0)
00110011	Version	Reserved
.....	Contents	01 00 01 00 01

Octet074	Result	

...00010	Tag	Result
101.....	Class and form	Context-specific, constructor
00000011	Length	3

Octet076	Integer	

...00010	Tag	Integer
000.....	Class and form	Universal, primitive
00000001	Length	1

```

..... Value                accepted
-----
Octet079 Result-source-diagnostic
-----
...00011 Tag                Result-source-diagnostic
101..... Class and form    Context-specific, constructor
00000101 Length            5
-----
Octet081 Dialogue service user
-----
...00001 Tag                Dialogue service user
101..... Class and form    Context-specific, constructor
00000011 Length            3
-----
Octet083 Integer
-----
...00010 Tag                Integer
000..... Class and form    Universal, primitive
00000001 Length            1
..... Value                Null
-----
Octet086 TCAP Component Portion
-----
...01100 Tag                TCAP Component Portion
011..... Class and form    Application-wide, constructor
10000000 Length            128
-----
Octet088 Invoke component
-----
...00001 Tag                Invoke component
101..... Class and form    Context-specific, constructor
00000110 Length            6
-----
Octet090 Invoke ID
-----
...00010 Tag                Invoke ID
000..... Class and form    Universal, primitive
00000001 Length            1
00000001 Invoke ID        01
-----
Octet093 Operation Code
-----
...00010 Tag                Local
000..... Class and form    Universal, primitive
00000001 Length            1
..... Operation Code      CUE Continue
-----
Octet096 End-of-contents
-----
00000000 Tag                00
00000000 Length            00
-----
Checksum CRC16..... 0011010011100010 hex=34E2
-----
-----
-----

```

Tipp zur Fehlerbehebung: UDTS-Rückgabequelle

Bei einer UDTS-Nachricht ist die 'return Cause' das erste Byte nach dem Nachrichtentyp ^{0a}. Dieser Wert hilft zu bestimmen, warum STP/SCP eine UDTS-Fehlerantwort sendet. Wenn diese Informationen im Sniffer nicht sichtbar sind, fahren Sie mit dem Abschnitt [Platform.log TCAP Trace](#) fort, um TCAP-Traces im Cisco PGW 2200-Protokoll zu aktivieren.

Platform.log TCAP Trace

Mit MML kann ein Benutzer eine TCAP-Ablaufverfolgung starten, die <Trace>-Nachrichten für den TCAP-Channel-Controller in /opt/CiscoMGC/var/log/platform.log ausgibt. Mit einer TCAP-Trace können die TCAP-/SCCP-Nachrichten angezeigt werden, die an den SS7-Channel-Controller gesendet werden, um über MTP3 zum SS7-Switch zu gelangen. Siehe [Anhang E](#) für den Nachrichtenfluss einer TCAP-Abfrage durch die PGW 2200-Software.

Die TCAP-Ablaufverfolgung wird mithilfe von MML mit dem Befehl **sta-tcap-trc** gestartet. Aktivieren Sie zur Erfassung der relevanten Informationen die Debug-Protokollierung für den TCAP- und SS7-Channel-Controller.

Dies ist ein Beispiel für das Aktivieren einer TCAP-Ablaufverfolgung:

```
mml> set-log:TCAP-01:debug,confirm
```

```
MGC-01 - Media Gateway Controller 2004-03-26 11:17:31.503 EST
```

```
M COMPLD
```

```
"TCAP-01"
```

```
;
```

```
mml> set-log:ss7-i-1:debug,confirm
```

```
MGC-01 - Media Gateway Controller 2004-03-26 11:17:40.715 EST
```

```
M COMPLD
```

```
"ss7-i-1"
```

```
;
```

```
mml> sta-tcap-trc
```

```
MGC-01 - Media Gateway Controller 2004-03-26 11:05:27.040 EST
```

```
M RTRV
```

```
SROF
```

```
"TCAP-01"
```

```
/* Component already started */
```

```
;
```

Hinweis: Die Debug-Protokollierung kann die Systemleistung beeinflussen und sollte nicht in einer Produktionsumgebung unter hohem Anrufaufkommen verwendet werden. Bitte planen Sie Ihr Wartungsfenster entsprechend.

Vom Cisco PGW 2200 gesendete TCAP-Nachrichten

Sobald ein `IN_TRIGGER` an die Engine gesendet wurde, sendet die Engine die Nachricht aus dem PGW 2200. Die von der Protokollebene weitergeleiteten Informationen werden an den TCAP-Channel-Controller weitergeleitet. Die TCAP-Komponente wird an den SCCP-Channel-Controller weitergeleitet. Außerdem wird in platform.log ein Protokoll erstellt, das angibt, dass eine TCAP-

Nachricht 'übertragen' wurde. Aus der vorherigen UDT-Meldung (die im Sniffer-Teil dieses Dokuments angezeigt wird) können Sie sehen, wie der PGW 2200 Informationen zu dieser Meldung im platform.log protokolliert. Dieses Plattformprotokoll stimmt mit dem Dateninhalt überein, der in der [Beispiel-SCCP-Nachrichtenaufschlüsselung](#) angezeigt wird: [Unitdata / Unitdata Service](#)-Tabelle in [Anhang C](#). Aus dieser Tabelle ist der erste Wert der Datenlänge (52 Hexadezimal = 82 Dezimalstellen). Die eigentliche TCAP-Datenkomponente folgt der Nachrichtenlänge. Falls Sniffer oder Snooper nicht verfügbar ist, kann diese platform.log zum Anzeigen/Debuggen von TCAP- und SCCP-Transaktionen verwendet werden.

Tipps zur Fehlerbehebung: Wenn die TCAP-Nachricht nicht an SCCP gesendet wird, liegt ein Problem auf MDL- oder Engine-Ebene vor. Beheben Sie die MDL-Ablaufverfolgung, und überprüfen Sie das Signal `Ltrigger` und `LTriggerRelease`.

Diese Ausgabe zeigt das PGW 2200-Protokoll, das den TCAP-Down-Stack an SCCP/MTP sendet.

```
Thu Dec 4 15:23:03:837 2003 EST | TCAP (PID 9513) <Trace>
PROT_TRACE_TCAP_PDU_TX: Hex dump of TCAP message transmitted, SSN=103,
LEN=82,
62 50 48 1 1f 6b 22 28 20 6 7 0 11 86 5 1 1 1 a0 15 60 13 80 2 7 80 a1 d 6 b 2a 81 76 82 15
1 1 1 1 0 1 6c 27 a1 25 2 1 1 2 1 0 30 1d 80 4 0 1 5f91 82 8 83 10 65 27 32 54 76 f83 7 3
11 3 23 22 11 11 9a 2 20 0
```

Nachdem TCAP die Nachricht an SCCP gesendet hat, übernimmt der SS7-Channel-Controller EMPFANGENES MSG FROM SCCP und protokolliert die Hexadezimaldarstellung der Nachricht, um den Empfang der Nachricht anzuzeigen. Dieser Hex-Dump enthält die SCCP- und TCAP-Bereiche, wie in dieser Ausgabe gezeigt.

```
Thu Dec 4 15:23:03:846 2003 EST | ss7-i-1 (PID 9518) <Debug>
RECEIVED MSG FROM SCCP ← INDICATES MESSAGE WAS FROM SCCP (TCAP)
```

```
Thu Dec 4 15:23:03:846 2003 EST | ss7-i-1 (PID 9518) <Debug>
<<<< To: 821 from 809 (bytes 98) prior 0 sio 83 sls 8: ← DPC 1-004-1, OPC 1-001-1
```

```
Thu Dec 4 15:23:03:846 2003 EST | ss7-i-1 (PID 9518) <Trace>
PROT_TRACE_MTP3_PDU: Hex dump of MTP3 and UP messages 1e0002 1 09 80 03 07 0b 04 c3 21 08 0c 04 c3 09 08
67
52 62 50 48 01 1f 6b 22 28 20 06 07 00 11 86 05 01 01 01 a0 15 60 13 80 02 07 80 a1 0d 06 0b 2a 81 76 82 15 01 01 01 00
01 6c 27 a1 25 02 01 01 02 01 00 30 1d 80 04 0 0 01 5f91 82 08 83 10 65 27 32 54 76 0f 83 07 03 11 03 23 22 11 11 9a 02 20
00
```

Tipps zur Fehlerbehebung:

- Verwenden Sie das in [Anhang C](#) dargestellte SCCP-Nachrichtenformat, um den Meldungstyp, die SCCP-Headerinformationen (in der [Ausgabe](#) in gelb dargestellt) und den Anfang der TCAP-Daten (in der [Ausgabe](#) in blau dargestellt) zu decodieren. Die `1e0002` in der [Ausgabe](#) stellt den Zielpunktcode aus `dpc.dat` dar, und der SCCP-Nachrichtendump beginnt unmittelbar nach dem Typ "1" (beginnend mit dem SCCP-Meldungstyp).
- Der PGW 2200 protokolliert Zähler und Alarme für SCCP-, TCAP- und SS7-Ereignisse. Wenn Messwerte aktiviert sind, überprüfen Sie die Zähler für die TCAP-Nachricht. Überprüfen Sie außerdem die empfangenen und übertragenen SCCP-, UDT- und UDTS-Protokolle. Weitere Informationen zu den Betriebsverfahren für MGC finden Sie in diesen Dokumenten. [Verwalten](#)

[von SystemmessungenCisco MGC-MesswerteAbrufen von TCAP-Transaktionen](#)

- Wenn der SS7-Kanal-Controller die vom PGW 2200 gesendete Nachricht nicht empfängt, überprüfen Sie, ob TCAP eine Nachricht nach unten an das SCCP übertragen hat. Wenn der TCAP-Layer die Nachricht nach unten überträgt, kann dies daran liegen, dass das SCCP nicht über genügend Informationen zum Erstellen der richtigen SCCP-Nachricht verfügt. Dies kann auch ein Hinweis darauf sein, dass das SS7-Subsystem nicht ordnungsgemäß bereitgestellt wird oder nicht verfügbar ist. Überprüfen Sie diese Liste:Konfiguration und Status des SS7-Point-CodesSS7-SubsystemkonfigurationRouting-Konfiguration für SS7-SubsystemLokaler und Remote-SSN-StatusIN-Dienstkonfiguration

(trigger.dat)**Systemüberprüfung**

```
mml>rtrv-spc:all
```

```
MGC-01 - Media Gateway Controller 2004-03-26 13:22:05.492 EST
```

```
M RTRV
```

```
"ss7svc1:DPC=001.022.001,DNW=2:OPC=001.001.001:IS"
```

```
"ss7svc2:DPC=001.022.002,DNW=2:OPC=001.001.001:IS"
```

```
"itussn1:DPC=001.004.001,DNW=2:OPC=001.001.001:IS"
```

```
"itussn2:DPC=001.003.001,DNW=2:OPC=001.001.001:IS"
```

```
"itussn3:DPC=001.004.001,DNW=2:OPC=001.001.001:IS"
```

```
;
```

```
mml> prov-rtrv:ss7subsys:NAME="itussn1"
```

```
MGC-01 - Media Gateway Controller 2004-03-26 11:48:26.321 EST
```

```
M RTRV
```

```
"session=fix551tgp:ss7subsys"
```

```
/*
```

```
NAME = itussn1
```

```
DESC = pc_ssn rte-ssn 48
```

```
SVC = scp1
```

```
PRI = 1
```

```
MATEDAPC =
```

```
LOCALSSN = 101
```

```
PROTO = SS7-ITU
```

```
STPSCPIND = 1
```

```
TRANSPROTO = SCCP
```

```
OPC = opc1
```

```
SUAKEY =
```

REMOTESN = 48

*/

;

mml> **rtrv-lssn:all**

MGC-01 - Media Gateway Controller 2004-03-26 11:49:01.985 EST

M RTRV

"TCAP-01:SSN=12,PST=IS"

"TCAP-01:SSN=101,PST=IS"

"TCAP-01:SSN=102,PST=IS"

;

mml> **rtrv-rssn:all**

MGC-01 - Media Gateway Controller 2004-03-26 11:49:04.695 EST

M RTRV

"scp1:PC=001.004.001,SSN=12,PST=IS"

"scp1:PC=001.004.001,SSN=48,PST=IS"

;

mml> **prov-rtrv:inservice:name="finap-initdp"**

MGC-01 - Media Gateway Controller 2004-03-29 14:45:25.738 EST

M RTRV

"session=fix551tgp:inservice"

/* NAME = finap-initdp

SKORTCV = 90001

GTORSSN = ROUTEBYSSN

GTFORMAT = NOGT

MSNAME = finap-initdp

*/

;

mml> **prov-rtrv:SS7ROUTE:NAME="route4"**

MGC-01 - Media Gateway Controller 2004-03-30 11:53:08.493 EST

M RTRV

"session=fix551tgp:SS7ROUTE"

/*

```
NAME = route4
DESC = rte to 1.4.1 scp1
OPC = opcl
DPC = scp1
LNKSET = ls3
PRI = 1
*/
;
```

- Wenn alle diese Informationen korrekt angezeigt werden (wie in der oben angezeigten Ausgabe gezeigt), überprüfen Sie die markierten Werte, die von der TCAP-Protokollebene nach unten gesendet wurden, z. B. die SSN-, SCCPCalledParty- und/oder SCCPCallingParty-Adresse.

TCAP-Nachrichten, die in den Cisco PGW2200 eingegeben werden

Die umgekehrte Logik kann verwendet werden, um eine SS7-Nachricht zu verfolgen, die auf dem Cisco PGW 2200 eingeht und für die TCAP-/SCCP-Benutzerebene des SS7-Stacks bestimmt ist. Die PGW 2200-Protokolle zeigen die SS7-Nachricht an, die beim SS7-Kanal-Controller (von der SS7-Leitung) eingeht und zur Verarbeitung an TCAP gesendet wird. Die Nachricht wird auf jeder Ebene des SS7-Stacks unterbrochen. Beachten Sie außerdem OPC/DPC, Service Indicator (SIO) und SLS (Signaling Link Selection). OPC und DPC werden im ITU-Format dargestellt (nur in diesem Beispiel).

Tipps zur Fehlerbehebung: Überprüfen Sie den Meldungstyp, der von der SS7-Leitung empfangen wurde. Wenn eine UDTS-Nachricht empfangen wird, überprüfen Sie die 'return Cause'.

Diese Ausgabe zeigt das PGW 2200-Protokoll an, wenn es SCCP-Nachrichten von der SS7-Leitung empfängt:

Thu Dec 4 15:23:04:953 2003 EST | ss7-i-1 (PID 9518) <Debug>
CP Received PDU from ssetId 3, chan 0

Thu Dec 4 15:23:04:953 2003 EST | ss7-i-1 (PID 9518) <Trace>
PROT_TRACE_MTP3_PDU: Hex dump of MTP3 and UP
messages 1d0005 0 CP DATA IND len: 139 data: 83 09 48 08 02 09 ←msgtype 09= UDT

Thu Dec 4 15:23:04:953 2003 EST | ss7-i-1 (PID 9518) <Debug>
>>>> from: 821 to opc 809 (bytes 134) sio 83 sls 0: ← OPC 1-004-1, DPC 1-001-1

Thu Dec 4 15:23:04:953 2003 EST | ss7-i-1 (PID 9518) <Trace>
PROT_TRACE_MTP3_PDU: Hex dump of MTP3 and UP messages
1e0002 0 09 ffff80 03 07 0b 04 ffff c3 09 08 67 04 ffff c3 21 08 0c 7...<continues>

Thu Dec 4 15:23:04:953 2003 EST | ss7-i-1 (PID 9518) <Debug>
RECEIVED SCCP STACK MSG
<lines omitted>

Thu Dec 4 15:23:04:954 2003 EST | TCAP (PID 9513) <Trace>
PROT_TRACE_TCAP_PDU_RX: Hex dump of TCAP message received, SSN=103, LEN=118,
65 74 48 45 0 0 0 49 1 1f6b 2a 28 28 6 7 0 11 86 5 1 1 1 a0 1d 61 1b a1 d 6 b
2a 81 76 82 15 1 1 1 0 1 a2 3 2 1 0 a3 5 a1 3 2 1 1 6c 3d a1 17 2 1 4 2 1 17 30 f a0 d
30 b 80 1 a 81 1 0 a2 3 80 1 1 a1 22 2 1 5 2 1 23 30 1a 80 10 30 e a0 c a0 a a1 5 a0 3 81
1 6 82 1 a 81 1 1 a2 3 80 1 1

Tipps zur Fehlerbehebung: Verwenden Sie das in [Anhang C](#) dargestellte SCCP-Nachrichtenformat, um den Meldungstyp, die SCCP-Headerinformationen (in der [Ausgabe](#) gelb dargestellt) und den Start der TCAP-Daten zu decodieren. Die Nummer 1e0002 in der obigen Ausgabe stellt die Anruferadresse (OPC) für die am PGW empfangene Nachricht dar, die in dpc.dat dargestellt ist. Der SCCP-Nachrichtendump beginnt unmittelbar nach der "0" (beginnend mit dem SCCP-Nachrichtentyp).

Diese Ausgabe stammt aus dem PGW2200-Protokoll, wenn das UDTs TCAP über SCCP/MTP empfangen wird:

Thu Mar 25 18:35:35:385 2004 EST | ss7-i-1 (PID 27288) <Debug>
CP Received PDU from ssetId 3, chan 0

Thu Mar 25 18:35:35:385 2004 EST | ss7-i-1 (PID 27288) <Trace>
PROT_TRACE_MTP3_PDU: Hex dump of MTP3 and UP messages 1d0005 0
CP DATA IND len: 68 data: 83 09 48 08 a2 0a

Thu Mar 25 18:35:35:385 2004 EST | ss7-i-1 (PID 27288) <Debug>
>>>> from: 821 to opc 809 (bytes 63) sio 83 sls a:

Thu Mar 25 18:35:35:385 2004 EST | ss7-i-1 (PID 27288) <Trace>

PROT_TRACE_MTP3_PDU: Hex dump of MTP3 and UP messages 1e0002 0 0a 01 03 0d
11 04 ffffffff c3 09 08 65 0a ffffffff 8b 21 08 30 00 18 38 33 44 44 29 62
27 48 01 02 6c 22 ffffffff a1 20 02 01 01 02 01 00 30 18 ffffffff 80 04 00 00
00 01 ffffffff 82 07 01 10 18 38 33 44 44 ffffffff 83 07 01 11 07 13 11 00 10

Thu Mar 25 18:35:35:385 2004 EST | TCAP (PID 27283) <Debug>

Got 91 bytes from fifo /tmp/sccp_input (fd=16)

Thu Mar 25 18:35:35:385 2004 EST | ss7-i-1 (PID 27288) <Debug>

RECEIVED SCCP STACK MSG

!--- Indicates message is from MTP(SS7 stack). !--- Lines omitted. Thu Mar 25 18:35:35:385 2004
EST | TCAP (PID 27283) <Debug> 00 01 00 01 1E 00 15 00 00 00 1A 00 00 02 00 00 00 00 00 08 21
00 00 08 09 FFF0A 0A 01 03 0D 11 04 FFF09 08 65 0A FFF21 08 30 00 18 38 33 44 44 29 62 27 48 01
02 6C 22 FFF20 02 01 01 02 01 00 30 18 FFF04 00 00 00 01 FFF07 01 10 18 38 33 44 44 FFF07 01 11
07 13 11 00 10 Thu Mar 25 18:35:35:386 2004 EST | TCAP (PID 27283) <Debug>
ioTcSuIntfc::handleNotInd: **Cause =1**

Thu Mar 25 18:35:35:386 2004 EST | TCAP (PID 27283) <Debug>

Calling StUiStuDatReq(), spId = 1

Thu Mar 25 18:35:35:386 2004 EST | TCAP (PID 27283) <Debug>

Deleted spDlgEntry 2-69

Thu Mar 25 18:35:35:386 2004 EST | TCAP (PID 27283) <Debug>

Sending **msgType 15 to Engine**

!--- TCAP sends response to Engine which is translated into L.

Diese Ausgabe stammt aus dem PGW 2200-Protokoll, wenn eine ungültige TCAP-Nachricht über SCCP/MTP empfangen wird:

```
Tue Mar 23 16:24:51.565 2004 EST | ss7-i-1 (PID 22997) <Trace>
PROT_TRACE_MTP3_PDU: Hex dump of MTP3 and UP messages
1d0005 0 CP DATA IND len: 12 data: 83 09 48 08 02 0a ←msgtype 10= UDTS
```

```
Tue Mar 23 16:24:51.565 2004 EST | ss7-i-1 (PID 22997) <Debug>
>>>> from: 821 to opc 809 (bytes 7) sio 83 sls 0:
```

```
Tue Mar 23 16:24:51.565 2004 EST | ss7-i-1 (PID 22997) <Trace>
PROT_TRACE_MTP3_PDU: Hex dump of MTP3 and UP messages
1e0002 0 0a 03 00 00 00 00 00 ←Msg Type 10 (UDTS), Return cause = 03 =
<lines omitted>
```

```
Tue Mar 23 16:24:51.565 2004 EST | ss7-i-1 (PID 22997) <Debug>
RECEIVED SCCP STACK MSG
<lines omitted>
```

```
Tue Mar 23 16:24:51.566 2004 EST | TCAP (PID 22992) <Debug>
00 01 00 01 1E 00 15 00 00 00 1A 00 00 02 00 00 00 00 00 00 08 21 00 00 08
09 FFF00 0A 03 00 00 00 00 00 ← OA= dec (10) = UDTS message is
incorrect format missing parameters
```

```
Tue Mar 23 16:24:51.566 2004 EST | TCAP (PID 22992) <Error>
TIOS_ERR_SCCP_SYNTAX_ERR: Syntax error in SCCP switch 1 suId = 0
```

MDL Trace Tool

Der Cisco PGW 2200 initiiert mithilfe von Triggern eine TCAP-Transaktion. Bei TCAP-Protokolltransaktionen wird die `IN_TRIGGER`-Methode verwendet, um Nachrichten an die TCAP-Steuerungsebene zu senden und von dieser zu empfangen. Wenn die Anrufanalyse auf den Ergebnistyp 22 trifft, wird das `IN_TRIGGER` TCAP-Protokoll initialisiert. TCAP-Informationen/-Nachrichten werden zwischen der TCAP-Protokollschicht (z. B. in MDL-Sprache geschriebene Trigger) und dem Cisco PGW 2200-Engine-Prozess unter Verwendung eines Tags, einer Länge und eines Werts oder einer TLV-Syntax ausgetauscht. Die Engine leitet die Informationen dann zur weiteren Verarbeitung an den TCAP Channel Controller weiter.

Verwenden Sie die Cisco PGW 2200-MDL-Ablaufverfolgung, um die Daten anzuzeigen, die an die und von der TCAP-Protokollschicht an den TCAP-Controller (über die Engine) gesendet werden. Der TCAP-Channel-Controller verarbeitet die empfangenen MDL-Nachrichten und leitet sie an den entsprechenden IOCC weiter (entweder TALI-IOCC, IP-IOCC oder SS7-IOCC). Die Engine wandelt auch vom TCAP-Channel-Controller (über SCCP/MTP3) empfangene TCAP-Nachrichteninformatoren in ein TLV-Format um, das an die TCAP-Protokollschicht übergeben werden kann, die auch `IN_TRIGGER` genannt wird. Gehen Sie wie folgt vor, um einen TCAP-Aufruf auf Protokollebene zu verfolgen:

1. Starten Sie eine MDL-Ablaufverfolgung.

```
mm1> sta-sc-trc:ss7svc1:log="udts",confirm
```

2. Führen Sie einen Aufruf aus, der einen TCAP-Dienst auslöst (trifft auf den Ergebnistyp der Analyse `IN_TRIGGER`).

3. Stoppen Sie die MDL-Spur.

```
mm1> stp-sc-trc:all
```

M COMPLD

"ALL:Trace stopped for the following files:

```
../var/trace/udts_ss7svc2_20040324174103.btr
```

4. Führen Sie **get_trc aus**, um die erfasste MDL-Ablaufverfolgung anzuzeigen.

```
get_trc.sh udts_ss7svc2_20040324174103.btr
```

5. Führen Sie Option **S** aus, um einen "SIM-Ausdruck" des Anrufs anzuzeigen, der den Nachrichtenfluss zwischen den internen PGW 2200-Prozessen anzeigt.

6. Führen Sie Option **D aus**, um die tatsächliche Ablaufverfolgung des Anrufs durch den PGW 2200-Code anzuzeigen. **Hinweis:** Der von den Optionen **D** und **S** in **get_trc.sh** angezeigte Inhalt ist möglicherweise nicht offensichtlich, da die Daten mit internen Datentypen und Variablennamen angezeigt werden. Eine Beschreibung dessen, wonach TCAP-Transaktionen gedebuggt werden sollen, finden Sie jedoch im Abschnitt **MDL Trace Analysis for TCAP**.

MDL Trace Analysis für TCAP

Verwenden Sie "sim print" (Option S von **get_trc.sh**), um den gesamten Anruffluss auf der Cisco PGW 2200-Protokollebene anzuzeigen. Der SIM-Druck ähnelt dem in [Anhang D](#) dargestellten. Wenn dies nicht der Fall ist, versuchen Sie, die Abweichung des abgeleiteten Anrufablaufs zu notieren, und beginnen Sie, die Fehlerbehebung für dieses Ereignis durchzuführen. Konzentrieren Sie sich bei der TCAP-Fehlerbehebung auf eines dieser Ereignisse.

- LTigger
- LTriggerInformationen
- LTriggerWeiter
- LtriggerRelease

Dies sind die internen Ereignisse, die den Statuscomputer `IN_TRIGGER` steuern.

Verwenden Sie die Cisco PGW 2200-MDL-Ablaufverfolgung, um den tatsächlichen Codefluss für jedes dieser Ereignisse anzuzeigen. LTrigger führt zu einem OUTPUT `IN_TRIGGER`, und die anderen drei werden von `IN_TRIGGER` durch eine INPUT `IN_TRIGGER`-Nachricht vom Motor empfangen.

Ausgehende TCAP-Nachrichten

Um Nachrichten zu identifizieren, die in und aus MDL für TCAP kommen, suchen Sie nach `IN_TRIGGER` in der MDL-Ablaufverfolgung. Die [Sample IN_TRIGGER Syntax aus der MDL Trace-Grafik](#) zeigt eine Nachricht, die an den und vom Motor gesendet und in MDL empfangen wurde. Die Ausgabe gibt an, dass `IN_TRIGGER` eine Anforderung an die Engine gesendet hat, eine TCAP-Nachricht weiterzuleiten.

Tipps zur Fehlerbehebung

- Verwenden Sie die MDL-Ablaufverfolgung, um zu überprüfen, ob die `TRIGGER`-Nachricht an

- den Motor gesendet wurde, wenn `IN_TRIGGER` oder `OUTPUT` nicht gesendet wurde.
- Überprüfen Sie den Nummernplan für die Ergebniskonfiguration `IN_TRIGGER`.
 - Überprüfen Sie die Konfiguration in Service und / oder `trigger.dat`.
 - Überprüfen Sie, ob die Nachricht vom SS7-Kanal-Controller gesendet wurde. Wenn die Nachricht nie aus dem SS7-Kanal-Controller stammt, ist dies darauf zurückzuführen, dass der SCCP-Channel-Controller nicht über genügend Informationen verfügt, um den Anruf weiterzuleiten oder eine gültige Nachricht zu erstellen.
 - Überprüfen Sie die SCCP-Konfiguration und die Konfiguration von `SS7_SUBSYSTEM`.
 - Überprüfen Sie den SSN-Status.
 - Überprüfen Sie den PC-Status.

Wenn die Ausgabe des `IN_TRIGGER` erfolgreich ist, zeigt die Cisco PGW 2200-MDL-Ablaufverfolgung die Antwort auf diese Nachricht als `INPUT` in den `IN_TRIGGER` an.

Beispiel `IN_TRIGGER`-Syntax aus MDL-Trace

```
OUTPUT 'IN_TRIGGER': 00 00 00 0e 00 00 00 69 00 01 0b 00 01 00 01 01 00 02 00 01 01 00 03 00 07 01 00 00 00 00 00 00 00 0e 00 01 03 00 0f
00 01 01 00 13 00 0d 02 00 2a b1 76 b2 15 01 01 01 01 00 01 00 05 00 01 01 00 06 00 03 01 02 00 00 07 00 01 01 00 09 00 1d 80 04 00 01 5f 91
82 08 83 10 65 27 32 54 76 0f83 07 03 11 03 23 22 11 11 9a 02 20 00 00 0a 00 00
```

```
INPUT 'IN_TRIGGER': 00 00 00 02 00 00 00 69 00 02 0d 00 12 00 04 00 00 08 21 00 11 00 04 00 00 00 02 00 10 00 12 00 00 00 08 21 0e 01 67
02 04 50 00 00 00 00 00 08 09 00 13 00 0d 03 00 2a b1 76 b2 15 01 01 01 01 00 01 00 05 00 01 01 00 06 00 03 01 00 17 00 07 00 01 04 00 09 00
0fa0 0d 30 0b 80 01 0a 81 01 00 a2 03 80 01 01 00 05 00 01 01 00 06 00 03 01 00 23 00 07 00 01 05 00 09 00 1a 80 10 30 0e a0 0c a0 0a a1 05 a0
03 81 01 06 b2 01 0a 81 01 01 a2 03 80 01 01 00 0a 00 00
```

Die `INPUT`-Nachricht ist die Antwort des Motors auf die vom TCAP-Protokoll gesendete Anforderung (oder `OUTPUT`-Nachricht). Die Engine kann im eigenen Namen oder im Namen der TCAP-Schicht reagieren.

Die `IN_TRIGGER`-Meldung gibt an, dass MDL TCAP-/SCCP-Informationen an die Engine und die Channel-Controller sendet, um eine UDT-Nachricht zu erstellen, die in der LINE an die SCP gesendet wird. Die an das Modul gesendeten Informationen werden von der Datei `trigger.dat` abgeleitet und direkt über der Ausgabe dieser Nachricht angezeigt. Um den Inhalt dieser Nachricht als MDL erstellt anzuzeigen, scrollen Sie im Text `IN_TRIGGER` nach oben. Der Start der Message-Building-Prozedur wird durch `SendMessage()...`, wie hier gezeigt, angegeben.

```
FUNCTION SendMessage() BEGIN
```

```
<messageData>.tagCount := bit(card(<messageData>.DATA), 8) -> '00001011'B
```

```
<messageData>.processId := bit(self(), 32) -> '00000000000000000000000000001101001'B
```

```
<messageData>.callRef := bit(CC.db.essentialData.releaseData.DATA.globalCallRefElem.DATA, 32)  
-> '00000000000000000000
```

```
0000000000000101'B
```

```
VAR inTable := GetTT(<trigger>, 2) -> 24 ← TRIGGER TABLE in trigger.dat (FINAP Initial DP)
```

```
VAR msTable := GetLN(inTable, 1) -> 24 ← IN Service Index (see figure 9)
```

```
SELECT GetMS(msTable, 3) -> 1 ← Msg type 1 = ITU BEGIN
```

```
OUTPUT Begin TO LINE AS <messageData> -> ELEMENT
```

```
SET TcapTimer := <defaultTimer> -> 5000
```

```
...<omitted lines>
```

```
NEXTSTATE <state> -> STATE_WaitResponse
```

```
END INPUT
```

```
END STATE
```

```
ok
```

writing message Begin

←TCAP MESSAGE TYPE

writing element _Begin

writing field callRef

← Identifies Call reference for MDL/engine Xaction.

'0000 0000 0000 0000 0000 0000 0000 0010'B

ok

writing field processed

← Identifies process ID for MDL/engine Xaction

'0000 0000 0000 0000 0000 0000 0110 1001'B

ok

writing field msgType

← Identifies Msg Type for MDL/engine Xaction

'0000 0000 0000 0001'B

← Msg type 1 = ITU BEGIN

ok

writing field tagCount

← Identifies number of tags included in this msg

'0000 1011'B 11 0x0b

ok

```

writing field DATA                ← beginning of tags

writing element TcapTypeElem       ← Tag element #1

writing field DATA                ← Tag element #1 data portion begins

writing field octet1               ← Tag element #1 field begins

writing field tcapType             ← Tag element #1 field, variable name

'0000 0001'B 1 0x01               ← Tag #1 VALUE; tcapType = 01

ok

ok

ok

writing field ieId                 ← Tag element #1 TAGID

'0000 0000 0000 0001'B

ok

writing field ieLength             ← Tag element #1 TAG LENGTH

'0000 0000 0000 0001'B

ok

ok

writing element TcapSystemDestElem ← Tag element #2

```

...

Tipps zur Fehlerbehebung

- Wenn vom Cisco PGW 2200 eine TCAP-Abfrage mit falschen Daten gesendet wird, kann mithilfe der MDL-Ablaufverfolgung genau ermittelt werden, wohin die Informationen vom Cisco PGW 2200 abgeleitet wurden. Die meisten Informationen stammen aus der Datei trigger.dat. Um zu sehen, wohin der Cisco PGW 2200 seine Informationen für die ausgehende Nachricht ableitet, suchen Sie (von `IN_TRIGGER`) nach dem betreffenden TCAP-Element. Wenn beispielsweise der TCAP-Typ falsch codiert ist, suchen Sie die Zeichenfolge `tcapType` in der MDL-Ablaufverfolgung (um das Schreibfeld `tcapType`).
- Um zu sehen, wo der Cisco PGW 2200 trigger.dat liest, um den TCAP-Inhalt zu codieren, suchen Sie nach den Zeichenfolgen in dieser Tabelle. Diese Zeichenfolgen stellen die Prozeduraufrufe dar, mit denen die Trigger.dat-Informationen abgerufen werden. Diese Prozeduraufrufe sollten zwischen dem `INPUT LTrigger`-Ereignis und der betreffenden `OUTPUT IN_TRIGGER`-Meldung erfolgen.

Name	Beschreibung	MDL-Suchzeichenfolge
------	--------------	----------------------

TT	Trigger-Tabellendatensatz	GetTT
MA	Datensatz der Nachrichtenaktion	GetMA
MS	Nachrichtensender Datensatz	GetMS
Betriebssystem	Operation Sending	GetOS
PS	Parameter-Sendedatensatz	GetPS
RR	Antwortdatensatz erhalten	GetRR
HERR	Empfangs-Datensatz	GetMR
ODER	Vorgang empfängt	GetOR
PR	Parameter Receiving Record	GetPR
RA	Antwortaktionsdatensatz	GetRA
AD	Aktionsdaten	GetAD

Eingehende TCAP-Nachrichten

Die `INPUT`-Meldung ist die Antwort des Motors auf die Anforderung. Die Engine kann im eigenen Namen oder im Namen der TCAP-Schicht reagieren. Die eingehende Nachricht wird durch die `INPUT IN_TRIGGER`-Meldungszeichenfolge in der MDL-Trace des Cisco PGW 2200 identifiziert, wie in dieser Beispielausgabe gezeigt. In diesem Beispiel wird auch die dekodierte Meldung angezeigt. Dies ist hilfreich, wenn Sie Probleme identifizieren müssen, die bei der TCAP-Antwort auftreten können.

Um die Engine-Nachricht zu decodieren, die der Cisco PGW 2200 MDL erhält, verwenden Sie das gleiche TLV-Format, wie bereits zuvor in diesem Dokument beschrieben. Diese Nachricht wird unmittelbar nach dem Text `INPUT IN_TRIGGER` dekodiert.

```
INPUT "IN_TRIGGER": 00 00 00 02 00 00 00 69 00 02 0d 00 12 00 04
00 00 08 21 00 11 00 04 00 00 00 02 00 10 00 12 00 00 00 08 21 0c 01 67 02
04 50 00 00 00 00 00
```

```
08 09 00 13 00 0d 03 00 2a 81 76 82 15 01 01 01 01 00 01 00 05 00 01 01
00 06 00 03 01 00 17 00 07 00 01 04 00 09 00 0f a0 0d 30 0b 80 01 0a 81
01 00 a2 03 80 0
```

```
1 01 00 05 00 01 01 00 06 00 03 01 00 23 00 07 00 01 05 00 09 00 1a 80
10 30 0e a0 0c a0 0a a1 05 a0 03 81 01 06 82 01 0a 81 01 01 a2 03 80 01
01 00 0a 00 00
```

```
reading element header: TcapMessageStyle
```

```
reading field callRef
```

```
!--- Identifies call reference for MDL / engine Xaction. '0000 0000 0000 0000 0000 0000 0000
0010'B ok reading field processed !--- Identifies process ID for MDL/engine Xaction. '0000 0000
```

```
0000 0000 0000 0000 0110 1001'B ok reading field msgType !--- Identifies message type for MDL/engine Xaction. '0000 0000 0000 0010'B !--- Message type 2 = ITU CONTINUE. ok reading field tagCount !--- Identifies the number of tags included in this message. '0000 1101'B 13 0x0d ok ok reading element _Continue !--- TCAP message type. reading field RAW 1136 bits read ok reading field DATA reading element header: TcapElementStyle !--- Tag element #1. reading field ieId !--- Tag element #1 TAG ID. '0000 0000 0001 0010'B ok reading field ieLength !--- Tag element #1 Tag Length. '0000 0000 0000 0100'B !--- 4 bytes. ok ok reading element TcapDatabaseIdElem reading field RAW 32 bits read ok reading field DATA !--- Tag element #1 data portion begins. '0000 0000'B 0 0x00 !--- Byte 1. '0000 0000'B 0 0x00 !--- Byte 1. '0000 1000'B 8 0x08 !--- Byte 1. '0010 0001'B 33 0x21 "!" !--- Byte 1. 'B ok ok reading element header: TcapElementStyle !--- Tag element #2. reading field ieId
```

Dies ist die Beispielausgabe einer eingehenden Antwort auf eine UDTS-Nachricht:

```
INPUT "IN_TRIGGER": 00 00 00 02 00 00 00 69 00 0f 02 00 0b
00 01 01 00 0a 00 00
```

```
reading element header: TcapMessageStyle
```

```
reading field callRef
```

```
'0000 0000 0000 0000 0000 0000 0000 0010'B
```

```
ok
```

```
reading field processId
```

```
'0000 0000 0000 0000 0000 0000 0110 1001'B
```

```
ok
```

```
reading field msgType
```

```
!--- Message type - Information message. '0000 0000 0000 1111'B ok reading field tagCount '0000 0010'B 2 0x02 ok ok reading element _Information
```

```
reading field RAW
```

```
72 bits read
```

```
ok
```

```
reading field DATA
```

```
reading element header: TcapElementStyle
```

```
reading field ieId
```

```
'0000 0000 0000 1011'B
```

```
ok
```

```
reading field ieLength
```

```
'0000 0000 0000 0001'B
```

```
ok
```

```
ok
```

```
reading element TcapErrorElem
```

```
!--- TCAP error element. reading field RAW 8 bits read ok reading field DATA reading field
```

```
octet1 reading field error '0000 0001'B 1 0x01 !--- TCAP error element = 01 ->
TCAP_ERROR_SSN_OOS. ok ok ok ok ok ok Continuing State Machine: IN_TRIGGER (105) STATE * INPUT
Information AS <messageData>
```

```
CC.db.nonEssentialData.TCAPTransactionUnixEndTimeElem.DATA
:= MGetTime(CC.db.nonEssentialData.TCAPTransactionMsecEndTimeElem.DATA)
-> 1080257735
```

Eine weitere nützliche Information, die Sie aus der Cisco PGW 2200 MDL-Spur (für TCAP-Anrufe) erhalten können, ist der Ursachenwert `LTriggerRelease`. Das in `LTriggerRelease` codierte `INErrorElem` liefert auch Einblicke, warum ein Aufruf oder eine TCAP-Transaktion nicht wie erwartet funktioniert. In dieser MDL-Grafik des Cisco PGW 2200 wird eine `LTriggerRelease` gezeigt, die als Antwort auf das ursprüngliche `LTrigger`-Ereignis gesendet wird, das `IN_TRIGGER` empfangen hat. In [Anhang E](#) finden Sie Einzelheiten zu `IN_TRIGGER`-Ereignissen und `INErrorElem`-Werten.

```
OD
```

```
END FUNCTION
```

```
VAR iErrorElem := NULL
```

```
iErrorElem.DATA.error := 42 → TRIG_ERROR_UNKNOWN
```

```
INSERT iErrorElem INTO <signalData>
```

```
IF (<signalData>::INActionElem = NULL) -> FALSE
```

```
FI
```

```
OUTPUT LTriggerRelease TO <callingProcess> -> 3 AS <signalData> -> ELEMLIST
```

```
NEXTSTATE <state> -> STATE_WaitResponse
```

```
END INPUT
```

```
END STATE
```

[Anhang A: MDL-Tags](#)

Die Cisco PGW 2200-MDL-Tags werden zwischen dem Cisco PGW 2200-MDL und dem Motor ausgetauscht. Dieser Anhang beschreibt die Reihenfolge, den Inhalt und das Format aller Tags, die in TCAP-Transaktionen verwendet werden. Die zum Auffüllen dieser Tagwerte verwendeten Informationen werden aus dem Anruferkontext und den in der Datei `trigger.dat` enthaltenen Werten abgerufen. Die Triggerdatei wird auch verwendet, um anzugeben, was an / von der Engine für das Erstellen von TCAP-Nachrichten gesendet werden soll und was vom Modul für die TCAP-Nachrichtenverarbeitung empfangen werden sollte, wenn eine Antwort empfangen wird.

Diese Tags werden für die TCAP-Anrufverarbeitung verwendet:

- **TAG-ID 1 - TCAP-TypBeschreibung:** Angabe des Typs der TCAP MDL Datenlänge:
fixed(1) Datenformat:
1 = ETSI 300 374-1

2 = Bell Core GR-1298-CORE
TR-NWT-001284
TR-NWT-001285

3 = Bell Core Pre AIN
GR-1428-CORE

- **TAG-ID 2 - SystemzielBeschreibung:** Internes Ziel des Ereignisses**Datenlänge:** fixed(1)**Datenformat:** Oktett**Inhalt:** 0 = Internes SCP, 1 = Trillium-TCAP
- **TAG-ID 3 - Angerufene SCCP-AdresseBeschreibung:** SCCP-Daten erforderlich für Trillionen**Datenlänge:** variabel**Datenformat:**

Octet 1 Routing Indicators

Bit A 0 - Route by GT, 1 - Route by SSN
Bit B DPC is present (Octets 2 to 4 have valid data)
Bit C SSN is present (Octet 5 has valid data)

Octet 2 DPC Network

Octet 3 DPC Cluster

Octet 4 DPC Member

Octet 5 Called SSN

Octet 6 GTFormat

0 - No global Title Included
1 - Global Title includes nature of address indicator only (ITU)
- Global title includes translation type,
numbering plan and encoding scheme.(ANSI)
2 - Global Title Includes translation type only.(ITU/ANSI)
3 - Global title includes translation type,
numbering plan and encoding scheme.
(ITU). - not used in ANSI.
4 - Global Title includes translation type, numbering plan,
encoding scheme and nature of address digits.
(ITU). - Not used in ANSI.

Octet 7 Translation Type Value

Octet 8 Numbering Plan

0 - Unknown
1 - ISDN Telephony
2 - Telephony
3 - Data
4 - Telex
5 - Maritime Mobile
6 - Land Mobile
7 - ISDN Mobile

Octet 9 Nature Of Number

1 - Subscriber Number
2 - National Number
3 - International Number

Octet 10 Number Of Digits in octets 11 to 43

Octet 11 to 43

Digits in IA5 format

- **TAG-ID 4 - SCCP-AnruferadresseBeschreibung:** SCCP-Daten erforderlich für Trillionen**Datenlänge:** variabel**Datenformat:**

Octet 1 Routing Indicators

Bit A 0 - Route by GT, 1 - Route by SSN

Bit B DPC is present (Octets 2 to 4 have valid data)
Bit C SSN is present (Octet 5 has valid data)

Octet 2 DPC Network

Octet 3 DPC Cluster

Octet 4 DPC Member

Octet 5 Calling SSN

- **TAG-ID 5 - Typ der TCAP-Komponente****Beschreibung:** Typ der TCAP-Komponente**Datenlänge:** fixed(1)**Datenformat:**

Octet

0 = Unknown
1 = Invoke
2 = Return Result Last
3 = Return Error
4 = Reject
5 = Return Result Not Last
6 = Invoke Last
7 = Invoke Not Last

- **TAG-ID 6 - TCAP-Vorgangscod****Beschreibung:** TCAP-Nachrichtenoperationscode**Datenlänge:** Variable (immer 4 für ANSI)**Datenformat:**

Octet 1 Flag

0 = None
1 = Local
2 = Global
3 = National
4 = Private

Octet 2 Operation Class

Octet 3 Op Code Highest byte (ITU) Family (ANSI)

Octet 4 Op Code Next byte (ITU) Specifier (ANSI)

Octet n Op Code Least byte (ITU)

- **TAG-ID 7 - TCAP-Invoke-ID****Beschreibung:** ID der Komponente**Datenlänge:** fixed(1)**Datenformat:** Oktett
- **TAG-ID 8 - TCAP Correlation-ID****Beschreibung:** ID der Komponente, der diese Komponente zugeordnet ist**Datenlänge:** fixed(1)**Datenformat:** Oktett
- **TAG-ID 9 - TCAP Dialog Component ANSI****Beschreibung:** Text einer TCAP-Nachricht ab dem ersten Parameter**Datenlänge:** variabel**Datenformat:** Oktett
- **TAG-ID 10 - Endmarker für den TCAP-Dialog****Beschreibung:** Haupttext einer TCAP-Nachricht ab dem ersten Parameter (SEQUENCE)**Datenlänge:** fixed(0)**Datenformat:** Keine
- **TAG-ID 11 - Fehler****Beschreibung:** Fehlerdaten**Datenlänge:** fixed(1)**Datenformat:** Oktett**Inhalt:**
 - 1 = TCAP_ERROR_SSN_OOS
 - 2 = TCAP_ERROR_PC_UNAVAILABLE
 - 3 = TCAP_ERROR_SERVICE_NOT_RESPONDING
 - 4 = TCAP_TRIGGER_TIMEOUT
- **TAG-ID 12 - STP-SCP-Gruppenindex****Beschreibung:** STP-SCP-Gruppenindex, Daten, die von der Analyse übergeben wurden.**Datenlänge:** fixed(1)**Datenformat:** Oktett**Inhalt:** STP-SCP Group Index-Wert.
- **TAG-ID 13 - TCAP Transport Protocol****Beschreibung:** Art des Transportprotokolls**Datenlänge:** fixed(1)**Datenformat:** Oktett**Inhalt:**
 - 1 = TCAP_TRANSPORT_SCCP
 - 2 = TCAP_TRANSPORT_TCP_IP

- **TAG-ID 14 - Externer TCAP-Fehler/-ProblemBeschreibung:** Fehler- oder Problemwert, der in Fehler- und Ergebniskomponenten empfangen oder versendet wurde
Datenlänge: variabel
Datenformat: Oktett
- **TAG-ID 15 - TCAP-Body-TypBeschreibung:** Art des Bauteils
Datenlänge: fixed(1)
Datenformat: Oktett
Inhalt:
1 = TCAP_BODY_SEQUENCE

2 = TCAP_BODY_SET
- **TAG-ID 16 - Informationen zum TCAP-DialogBeschreibung:** Trillium TCAP enthält dieses TAG in allen Nachrichten, die an MDL gesendet werden. MDL sollte diese Informationen speichern und an das Trillium TCAP in allen nachfolgenden Nachrichten für den Dialog oder unidirektionale Nachrichten im Zusammenhang mit dem Anruf senden.
Datenlänge: variabel
Datenformat: Oktett
- **TAG-ID 17 - TCAP-Transaktions-IDBeschreibung:** Trillium TCAP enthält dieses TAG in allen Nachrichten, die an MDL gesendet werden. MDL sollte diese Informationen zum Senden an CDB speichern.
Datenlänge: variabel
Datenformat: Oktett
- **TAG-ID 18 - TCAP-Datenbank-IDBeschreibung:** Trillium TCAP wird dieses TAG in alle Nachrichten einschließen, die an MDL gesendet werden. MDL sollte diese Informationen zum Senden an CDB speichern.
Datenlänge: variabel
Datenformat: Oktett

Anhang B: SS7-Punktcodes abmelden

ETSI PC 1-1-1 (padded to 16 bits) =
00001000 00001001 = 08 09 = 809 (shown in log)

ETSI PC 1-4-1 (padded to 16 bits) = 00001000 00100001 =
08 21 = 821 (shown in log)

ETSI PC 3-3-3 (padded to 16 bits) - 00011000 00011011 =
18 1B = 181b (another ex.)

	Cluster	Netzwerk	Mitglied	Punktcode
ESTI (14 Bit)	3 Bit	8 Bit	3 Bit	14 Bit
ANSI (24 Bit)	8 Bit	8 Bit	8 Bit	24 Bit
PC 1-1-1 (kein Padding, nur 14 Bit)	001	000 0001	001	001000 = 8 00000001 = 01
PC 1-4-1 (kein Padding, nur 14 Bit)	001	00000100	001	001000 = 8 00100001 = 21
PC 3-3-3	011	00000011	011	011000 = 18 00011011 = 1 B

Anhang C: SCCP-Nachrichtentypen

Nachrichtentyp	Nachrichtentyp
CR Connection-Anfrage	0000 001
CC Connection bestätigen	0000 0010
CREF Connection verweigert	0000 0011
RLSD veröffentlicht	0000 0100
RLC-Version abgeschlossen	0000 0101
DT1-Datenformular 1	0000 0110
DT2-Datenformular 2	0000 0111
AK-Datenbestätigung	0000 1000
UDT-Unitdaten	0000 1001
UDTS Unitdata-Service	0000 1010
Beschleunigte Daten von ED	0000 1011
EA Schnellere Datenbestätigung	0000 1100
RSR Reset-Anforderung zurücksetzen	0000 1101
RSC-Reset-Bestätigung	0000 1110
Fehler der ERR Protocol-Dateneinheit	0000 111
IT-Inaktivitätstest	0001 0000
XUDT Extended Unitdata	0001 0001
XUDTS Extended Unitdata Service	0001 0010
LUDT Long Unitdata	0001 0011
LUDTS Long Unitdata Service	0001 0100

Unitdata (UDT)

Die UDT-Nachricht enthält:

- Drei Zeiger
- Die in dieser Tabelle angegebenen Parameter.

Parameter	Referenz Q.713	Typ (F V O)	Länge (Oktette)
Meldungstyp	2,1	F	1
Protokollklasse	3,6	F	1
Adresse des angerufenen Teilnehmers	3,4	V	Mindestens 3
Adresse des anrufenden Teilnehmers	1,5	V	Mindestens 3
Daten	3,16	V	2-X (Hinweis 1)

Hinweis: Aufgrund der laufenden Studien zur SCCP-Adresse für Angerufene und Anrufer muss die

maximale Länge dieses Parameters weiter untersucht werden. Es ist auch zu beachten, dass die Übertragung von bis zu 255 Oktetten von Benutzerdaten zulässig ist, wenn die SCCP-Adresse für Angerufene und Anrufer keinen globalen Titel enthält.

Unitdata-Service (UDTS)

Die UDTS-Nachricht enthält:

- Drei Zeiger.
- Die in dieser Tabelle angegebenen Parameter.

Parameter	Referenz Q.713	Typ (F V O)	Länge (Oktette)
Meldungstyp	2,1	F	1
Rückgabeursache	3,12	F	1
Adresse des angerufenen Teilnehmers	3,4	V	Mindestens 3
Adresse des anrufenden Teilnehmers	1,5	V	Mindestens 3
Daten	3,16	V	2-X (Hinweis)

Hinweis: Aufgrund der laufenden Studien zur SCCP-Adresse für Angerufene und Anrufer muss die maximale Länge dieses Parameters weiter untersucht werden. Es ist auch zu beachten, dass die Übertragung von bis zu 255 Oktetten von Benutzerdaten zulässig ist, wenn die SCCP-Adresse für Angerufene und Anrufer keinen globalen Titel enthält.

Diese Tabelle zeigt eine Beispiel-Aufschlüsselung der SCCP-Nachrichten für den Unitdata/Unitdata-Dienst:

Parameter	Typ (F V O)	Länge (Oktette)	Korrelation ausgehen der Nachricht en	Korrelation eingehen der Nachricht en
Meldungstyp	F	1	09	0 A
Protokollklasse	F	1	80	01
Adresszeiger des angerufenen Teilnehmers	F	1	03	03
Adresszeiger des Anrufers	F	1	07	0 d

Datenzeiger	F	1	0 b	11
Adresse des angerufenen Teilnehmers	V	Mindestens 3	04 c3 21 08 0c	04 c3 ... 30 00
Adresse des anrufenden Teilnehmers	V	Mindestens 3	04 c3 09 08 67	18 38 33 44 44
Daten (TCAP-DATEN)	V	04 c3 09 08 67 18 38 33 44 44 Daten (TCAP-DATEN) V	52 62 ... 20 00	29 62 ... 00 10

Hinweis: Diese Meldungen sind nur Beispiele und stellen möglicherweise keine tatsächliche Abfrageantwort-Kombination/-Sequenz dar.

UDTS-Rückgabeursachen

Im Unitdata-Dienst, im Extended Unitdata-Dienst oder in der Long Unitdata-Dienstmeldung ist das Parameterfeld "return Cause" ein Oktettfeld, das den Grund für die Rückgabe einer Nachricht enthält. Bits 1 bis 8 werden wie folgt codiert:

```
Value Bits
0 0 0 0 0 0 0 0 no translation for an address of such nature
1 0 0 0 0 0 0 1 no translation for this specific address
2 0 0 0 0 0 0 1 0 subsystem congestion
3 0 0 0 0 0 0 1 1 subsystem failure
4 0 0 0 0 0 1 0 0 unequipped user
5 0 0 0 0 0 1 0 1 MTP failure
6 0 0 0 0 0 1 1 0 network congestion
7 0 0 0 0 0 1 1 1 unqualified
8 0 0 0 0 1 0 0 0 error in message transport (Note)
9 0 0 0 0 1 0 0 1 error in local processing (Note)
10 0 0 0 0 1 0 1 0 destination cannot perform reassembly (Note)
11 0 0 0 0 1 0 1 1 SCCP failure
12 0 0 0 0 1 1 0 0 hop counter violation
13 0 0 0 0 1 1 0 1 segmentation not supported
14 0 0 0 0 1 1 1 0 segmentation failure
15 0 0 0 0 1 1 1 1
```

to

```
228 1 1 1 0 0 1 0 0 Reserved for International Use
229 1 1 1 0 0 1 0 1
```

to

```
254 1 1 1 1 1 1 1 0 Reserved for National Networks
255 1 1 1 1 1 1 1 1 Reserved
```

Anhang D: MDL-Schnittstelle für TCAP-Nachricht

Alle Nachrichten haben ein gemeinsames TLV-Format:

- **Call Instance und ProcessId:** 8 Byte lang, die vom Engine empfangen und in der

Antwortmeldung vom Engine unverändert zurückgegeben werden sollen.

- **Message ID** - Identifiziert die Nachricht, die von der Protokollschicht des TCAP gesendet oder empfangen wird (Werte in dieser [Tabelle](#)).
- **Tag-ID** - Die Anzahl der Tags und Tag-Daten (Tag-ID, Datenlänge und Daten) gibt an, was in der TCAP-Nachricht an das Remote-Ziel gesendet wird. Alle Feldgrößen sind festgelegt, mit Ausnahme des Datenfelds eines Tagelements, dessen Länge variabel ist und (in Oktetten) durch die Datenlänge definiert wird. Jedes der Felder Gesamtlänge, Anrufinstanz und Prozess-ID, Nachrichten-ID, Tag-ID und Datenlänge wird zuerst durch das signifikanteste Byte übertragen.

Anhang E: Interne MDL-Schnittstelle

Intern erfolgt die Kommunikation mit TCAP State Machine Objects (SMOs) über Signale mit Daten. Jeder MDL-Datentyp kann mit dem Signal gesendet werden. Die Namen und Bedeutungen der Signale und Daten sind hier aufgelistet.

- **LTriggerBeschreibung:** Dies ist das erste Signal, das der LCM an TCAP sendet, um den Dialog zu starten. In Elan enthält `INTriggerElem` auch den `stpScpGroupIndex`.
`MSG_ACTION_COPY_STP_SCP_INDEX_FROM_SIGNAL_DATA` muss in der MA-Tabelle festgelegt werden, damit dies verwendet werden kann.**Komponenten:** `INTriggerElem`, `BNumberElem`, `BNumberDataElem`
- **LTriggerInformationenBeschreibung:** Dieses Signal wird als Antwort auf `LTrigger` von TCAP an LCM gesendet, wenn der Dialog fortgesetzt wird.**Komponenten:** `INTriggerElem`, `BNumberElem`, `BNumberDataElem`
- **LTriggerWeiterBeschreibung:** Dieses Signal wird von LCM an TCAP als nachfolgende Trigger-Anforderung in einem vorhandenen Dialog gesendet.**Komponenten:** `INTriggerElem`, `BNumberElem`, `BNumberDataElem`
- **LTriggerReleaseBeschreibung:** Dieses Signal ist das letzte, das entweder von LCM oder von TCAP gesendet wird und als Antwort an `LTrigger` gesendet werden kann, nachdem die SCP eine Antwort erhalten hat.**Komponenten:** `INErrorElem`, `BNumberElem`, `BNumberDataElem`
`INErrorElem` hat folgende Werte:

- 1 `TRIG_ERROR_NONE,`
- 2 `TRIG_EXIT_UNABLE_TO_COMPLETE_MA_IS_LNP_M_BIT_CLEAR,`
- 3 `TRIG_ERROR_NULL_TRIGGER,`
- 4 `TRIG_ERROR_TRIGGER_TABLE_NOT_FOUND,`
- 5 `TRIG_ERROR_UNKNOWN_MESSAGE_ACTION,`
- 6 `TRIG_ERROR_UNKNOWN_RESPONSE_ACTION,`
- 7 `TRIG_ERROR_UNKNOWN_PARAMETER_ACTION,`
- 8 `TRIG_ERROR_MESSAGE_ACTION_FAILED,`
- 9 `TRIG_ERROR_UNABLE_TO_LOAD_DIALOGUE_COMPONENT,`
- 10 `TRIG_ERROR_UNABLE_TO_LOAD_TAG,`
- 11 `TRIG_ERROR_READING_TT,`

12 TRIG_ERROR_READING_MA,
13 TRIG_ERROR_READING_PS,
14 TRIG_ERROR_READING_RR,
15 TRIG_ERROR_READING_PR,
16 TRIG_ERROR_READING_RA,
17 TRIG_ERROR_ACTION_NOT_COMPATIBLE_IN_PR,
18 TRIG_ERROR_NO_ACTION_DATA_FOR_ACTION_RE_TRIGGER,
19 TRIG_ERROR_NO_ACTION_DATA_FOR_ACTION_SEND_ACTION_TO_LCM,
20 TRIG_ERROR_UNKNOWN_MESSAGE_IN_MS,
21 TRIG_ERROR_UNKNOWN_PR_ACTION,
22 TRIG_ERROR_UNABLE_TO_COMPLETE_MA_COPY_SCCP_GT_FROM_BNUMBER,
23 TRIG_ERROR_UNABLE_TO_COMPLETE_MA_COPY_STP_SCP_INDEX_FROM_SIGNAL_DATA,
24 TRIG_ERROR_UNKNOWN_DIALOGUE_COMPONENT,
25 TRIG_ERROR_SIGNAL_IN_WRONG_STATE,
26 TRIG_ERROR_SCCP_TIMEOUT,
27 TRIG_ERROR_IN_RESPONSE_OPERATION_CODE_MISSING,
28 TRIG_ERROR_IN_RESPONSE_INVOKE_ID_IN_USE,
29 TRIG_ERROR_IN_RESPONSE_INVOKE_ID_NOT_FOUND,
30 TRIG_ERROR_IN_RESPONSE_CORROLATION_ID_NOT_FOUND,
31 TRIG_ERROR_IN_RESPONSE_UNEXPECTED_CORROLATION_ID,
32 TRIG_ERROR_IN_RESPONSE_NO_COMPONENT_CONTENTS,
33 TRIG_ERROR_IN_RESPONSE_INVALLID_COMPONENT_CONTENTS,
34 TRIG_ERROR_IN_RESPONSE_UNEXPECTED_INVOKE_ID,
35 TRIG_ERROR_IN_RESPONSE_EXTERNAL_ERROR_NOT_FOUND,
36 TRIG_ERROR_ABORT,
37 TRIG_ERROR_USER_ABORT,
38 TRIG_ERROR_PROTOCOL_ABORT,
39 TRIG_ERROR_UNKNOWN

Zugehörige Informationen

- [Cisco PGW 2200 Softswitch - Technische Hinweise](#)
- [Unterstützung von Sprachtechnologie](#)
- [Produkt-Support für Sprach- und Unified Communications](#)

- [Fehlerbehebung bei Cisco IP-Telefonie](#)
- [Technischer Support und Dokumentation - Cisco Systems](#)