

PGW 2200 Softswitch TCAP release 9.3 e successive

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Convenzioni](#)

[Premesse](#)

[Risoluzione TCAP](#)

[Sniffer sulla linea Ethernet](#)

[Traccia TCAP Platform.log](#)

[Strumento di traccia MDL](#)

[Appendice A Tag MDL](#)

[Appendice B Disconnetti codici punto SS7](#)

[Appendice C: Tipi di messaggio SCCP](#)

[Unitdata \(UDT\)](#)

[Servizio Unitdata \(UDTS\)](#)

[Cause restituzione UDTS](#)

[Appendice D: Interfaccia MDL per messaggio TCAP](#)

[Appendice E: Interfaccia MDL interna](#)

[Informazioni correlate](#)

[Introduzione](#)

La parte TCAP (Transaction Capabilities Applications Part) fornisce il supporto per le applicazioni interattive in un ambiente distribuito. TCAP definisce un protocollo end-to-end tra gli utenti. Può trovarsi in una rete SS7 o in un'altra rete che supporta TCAP (IP).

[Prerequisiti](#)

[Requisiti](#)

I lettori di questo documento dovrebbero conoscere:

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

[Componenti usati](#)

Per la stesura del documento, è stato usato uno switch software Cisco PGW 2200.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento [Cisco sulle convenzioni nei suggerimenti tecnici](#).

Premesse

Il protocollo TCAP è costituito da due sottolivelli:

- Sottolivello componente
- Livello secondario transazione

Il sottolivello componente si interfaccia con il motore di conversione. Il motore di conversione è l'equivalente di un numero di utenti del servizio o di un numero di sottosistema (SSN). Il livello secondario del componente supporta i seguenti servizi:

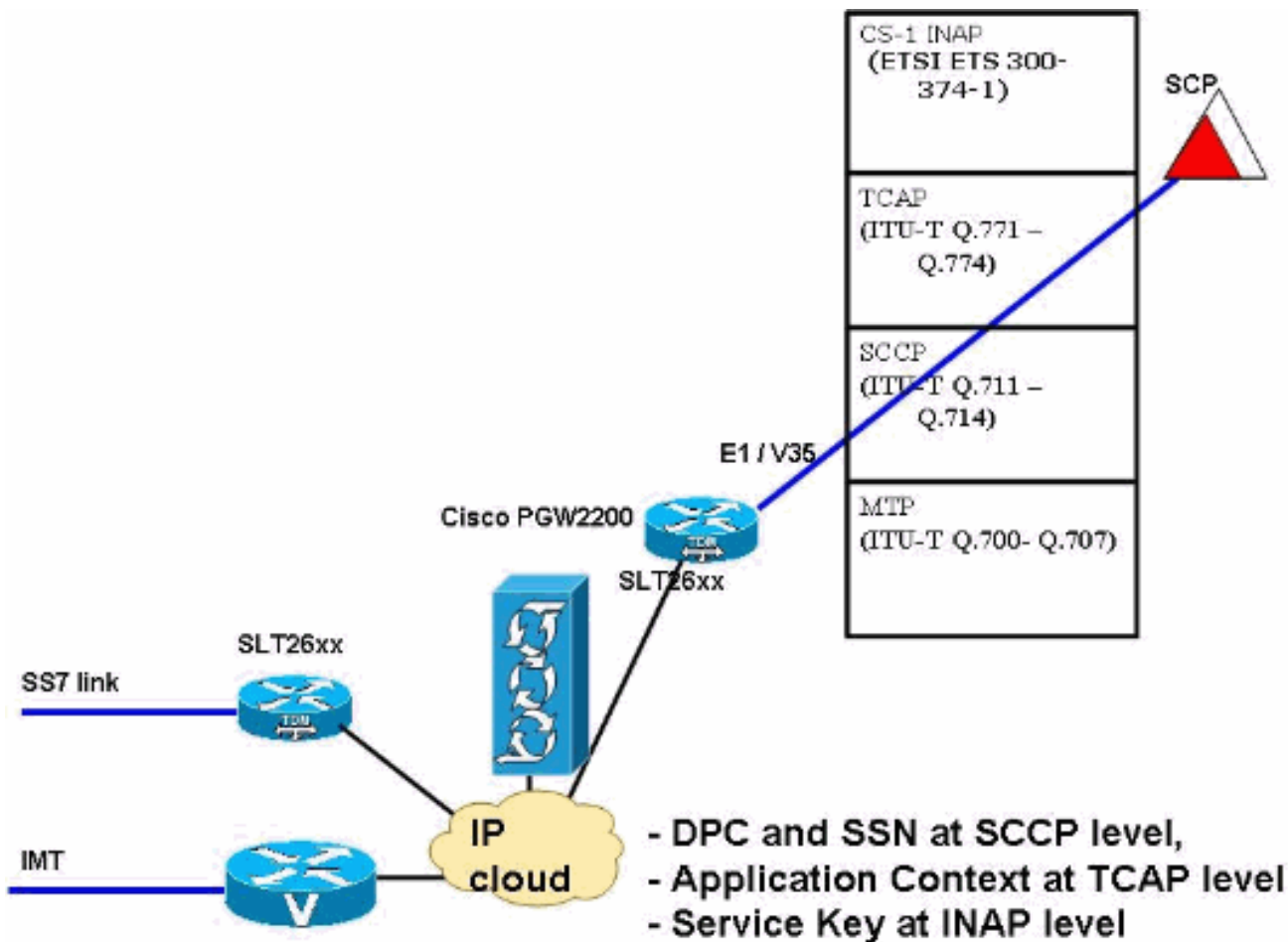
- Associazione di operazioni e risposte.
- Gestione della situazione anomala.

Il sottolivello della transazione si interfaccia con la parte SCCP (Signaling Connection Control Part). TCAP supporta solo un servizio di rete senza connessione. Il sottolivello della transazione comunica con il SCCP tramite l'interfaccia senza connessione.

Il software TCAP utilizza i servizi del software SCCP per instradare i messaggi all'utente TCAP nel nodo di destinazione. L'interfaccia tra il TCAP e il software SCCP è strettamente interconnessa. Ogni richiesta TCAP dal motore contiene un titolo globale e un numero di sottosistema di destinazione. TCAP fornisce il numero del sottosistema a SCCP per la ricerca del codice Signal Transfer Point (STP). Se gli indirizzi e le route SS7 sono configurati correttamente e completamente operativi, risolvere i problemi relativi alle informazioni SCCP e TCAP passate e ricevute tra il Cisco PGW 2200 e un peer SCCP o TCAP remoto.

Cisco PGW 2200 usa il SCCP per incapsulare le query TCAP per il trasferimento dei messaggi in uscita (MTP). Questa comunicazione SCCP tra peer viene inviata senza una connessione su MTP. Cisco PGW 2200 utilizza l'UDT (SCCP Unidata) per inviare i dati al nodo SCCP remoto per la comunicazione senza connessione. PGW 2200 riceve una risposta valida quando il messaggio UDT SCCP viene recapitato correttamente. Si tratta in genere di un messaggio UDT. Lo scambio di questi messaggi UDT facilita la comunicazione senza connessione tra PGW 2200 e il peer SCCP remoto (come Service Control Point [SCP] per le ricerche nel database TCAP). PGW 2200 definisce un campo opzionale nell'UDT che indica che il peer SCCP deve "restituire in caso di errore" il contenuto di qualsiasi messaggio inviato al nodo remoto se il messaggio UDT non è recapitato. Per facilitare questa risposta all'errore, viene utilizzato il messaggio del servizio Unidata (UDTS). Il messaggio UDTS indica a PGW 2200 che un messaggio UDT ricevuto sul nodo remoto (ad esempio STP o SCP) non può essere recapitato alla destinazione.

Cisco PGW 2200 Concept Setup



Risoluzione TCAP

La messaggistica SCCP (UDT/UDTS) descritta nella sezione [Informazioni di base](#) è fondamentale per la risoluzione dei problemi relativi ai servizi e alle funzionalità TCAP. Risolvere eventuali problemi a livello SCCP prima di risolvere i problemi relativi ai dati TCAP inviati o ricevuti. Il formato dell'UDT e del messaggio UDTS è illustrato nell'[appendice C](#).

Utilizzare questi strumenti Cisco PGW 2200 per eseguire il debug delle chiamate che richiedono i servizi TCAP (TCAP/SCCP):

- [Sniffate la linea Ethernet](#) con strumenti quali Etheral, UNIX snoop e Snooper.
- [Platform.log TCAP](#) su PGW 2200.
- [MDL Trace Tool](#) per l'elaborazione delle chiamate a livello di protocollo.

Sniffer sulla linea Ethernet

Cisco PGW 2200 utilizza un UDP (RUDP) affidabile per inviare messaggi MTP3 e SS7 di livello superiore tra i dispositivi MTP1 e MTP2 locali (ad esempio, Signaling Link Terminal [SLT]). Questa comunicazione viene in genere effettuata sulla porta 7000 sull'interfaccia Ethernet locale di Cisco PGW 2200. Questa operazione è configurabile. Fare riferimento alla [guida alla configurazione](#) per i dettagli sulla configurazione delle porte "stPort" PGW in XECfgParm.dat.

È possibile utilizzare uno sniffer Ethernet per visualizzare i pacchetti inviati tra il Cisco PGW 2200 e il relativo dispositivo di controllo MTP2 locale. Tuttavia, non tutti supportano il protocollo MTP e SCCP utilizzato per visualizzare un messaggio decodificato. Se il cliente non dispone di uno sniffer Ethernet, utilizzare il comando **snoop** UNIX per risolvere il problema. l'output del comando

snoop non è di facile utilizzo, ma è utile nel peggiore dei casi.

Si consiglia uno sniffer Ethernet che supporti lo stack di protocolli SS7. Consente di decodificare i pacchetti visualizzati sull'interfaccia Cisco PGW 2200 Ethernet. È inoltre possibile utilizzare un sniffer open source come [Ethereal](#), disponibile online.

Se non è disponibile un'utilità sniffer commerciale, usare il comando **snoop** sul Cisco PGW 2200 di destinazione per verificare l'output dei dati esadecimali dei messaggi inviati a e provenienti da Cisco PGW 2200. Con l'autorizzazione root su Cisco PGW 2200, eseguire questo comando per visualizzare i dati esadecimali inviati dalla 'porta st' configurata. Per ulteriori informazioni sul comando **snoop**, fare riferimento alle "pagine man snoop" o alle guide di amministrazione SUN.

```
#snoop -d
```

Eseguire questo comando per snoop dei pacchetti inviati dal dispositivo Ethernet, homeX, sulla porta 7000.

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

Questo è l'output di esempio dei pacchetti SS7 acquisiti con il comando **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: 684f3338 0000 0000 22b3 e70f0003 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 .....
```

È possibile usare anche Cisco's Snooper (se disponibile) per visualizzare l'immagine esadecimale del messaggio SCCP. L'intestazione del messaggio SCCP viene decodificata, ma la visualizzazione dell'output dipende dalla versione di Snooper scelta. Il punto importante è che il tipo di messaggio è visibile e fornisce un'indicazione su dove iniziare a risolvere i problemi relativi al flusso di chiamata. Il dump esadecimale indica che il tipo di messaggio 09 è un messaggio UDT e il tipo di messaggio 0a è il messaggio del servizio UDTs che indica un errore. La direzione del flusso del messaggio è utile anche perché vengono mostrati i PC SS7. Se viene visualizzato il resto della dump esadecimale (a seconda della versione dello snooper), può essere utilizzato per decodificare ulteriormente le parti SCCP e TCAP del messaggio. Ciò si basa sugli standard di settore per SCCP e TCAP.

Questo è l'output Snooper del messaggio SCCP UDT con dati TCAP (su 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

```

Se esiste un messaggio UDT SCCP non recapitato inviato da Cisco PGW 2200 e/o un SCCP (sul nodo remoto) ha problemi con il messaggio, Cisco PGW 2200 riceve un messaggio di risposta UDTS. Questo messaggio indica una "causa di ritorno" molto utile per la risoluzione dei problemi. Il tipo di messaggio UDTS è 10 (o 0a hex).

Questo è un esempio di messaggio SCCP UDTS con dati TCAP (da PSTN).

Nota: questo messaggio è solo di esempio e potrebbe non riflettere una combinazione/sequenza di risposta all'interrogazione effettiva. Il formato e la quantità di informazioni visualizzate variano a seconda della versione di Snooper.

```

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.....

```

Questo output Snooper visualizza la sequenza IAM, UDT, UDTS e REL.

Nota: questo messaggio è solo di esempio e potrebbe non riflettere una combinazione/sequenza di risposta all'interrogazione effettiva. Il formato e la quantità di informazioni visualizzate variano a seconda della versione di Snooper.


```

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

```

Traccia di sniffer SS7 che include informazioni SS7 SCCP e TCAP.

```

-----
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
-----
-----
-----

```

Suggerimento per la risoluzione dei problemi: Causa restituzione UDTs

Per un messaggio UDTs, la 'causa del ritorno' è il primo byte dopo il tipo di messaggio ^{0a}. Questo valore aiuta a determinare il motivo per cui il protocollo STP/SCP invia una risposta di errore UDTs. Se queste informazioni non sono visibili nello sniffer, passare alla sezione [Platform.log TCAP Trace](#) per abilitare le tracce TCAP nel registro Cisco PGW 2200.

[Traccia TCAP Platform.log](#)

MML consente a un utente di avviare una traccia TCAP che esegue il dump dei messaggi <Trace> per il controller di canale TCAP in /opt/CiscoMGC/var/log/platform.log. Una traccia TCAP consente all'utente di visualizzare i messaggi TCAP/SCCP inviati al controller di canale SS7 per il routing allo switch SS7 tramite MTP3. Vedere [Appendice E](#) per il flusso di messaggi di una query TCAP tramite il software PGW 2200.

La traccia TCAP viene avviata tramite mml con il comando **sta-tcap-trc**. Per acquisire le informazioni pertinenti, abilitare la registrazione del debug per il controller di canale TCAP e SS7.

Questo è un esempio di come abilitare una traccia TCAP:

```
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 */
```

```
;
```

Nota: la registrazione del debug può influire sulle prestazioni del sistema e non deve essere utilizzata in un ambiente di produzione con un volume di chiamate elevato. Pianificare la finestra di manutenzione in modo appropriato.

Messaggi TCAP inviati da Cisco PGW 2200

Una volta inviato un `IN_TRIGGER` al motore, quest'ultimo invia il messaggio dal PGW 2200. Le informazioni passate dal livello di protocollo vengono inoltrate al controller di canale TCAP. La parte TCAP viene inviata al controller del canale SCCP. Inoltre, in platform.log viene creato un log per indicare che un messaggio TCAP è stato "trasmesso". Dal messaggio UDT precedente (mostrato nella parte sniffer di questo documento) è possibile vedere come PGW 2200 registra le

informazioni relative allo stesso messaggio nel file platform.log. Questo log della piattaforma corrisponde al contenuto dei dati mostrato nell'[analisi stratificata dei messaggi SCCP di esempio: Unitdata / Unitdata Service](#) tabella nell'[Appendice C](#). In questa tabella, il primo valore è il valore della lunghezza dei dati (52 esadecimali = 82 decimali). La parte dati TCAP effettiva segue la lunghezza del messaggio. Nel caso in cui sniffer o snooper non sia disponibile, questo platform.log può essere utilizzato per visualizzare / eseguire il debug di transazioni TCAP e SCCP.

Suggerimento per la risoluzione dei problemi: Se il messaggio TCAP non viene inviato a SCCP, è presente un problema a livello MDL o di motore. Risolvere i problemi relativi alla traccia MDL e osservare il segnale `Ltrigger` e `LTriggerRelease`.

Questo output mostra il log PGW 2200 che invia il TCAP allo stack giù per SCCP/MTP.

```
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
```

Dopo che TCAP ha inviato il messaggio a SCCP, il controller del canale SS7 riproduce il messaggio `RECEIVED MSG FROM SCCP` e registra la rappresentazione esadecimale del messaggio per indicare la ricezione del messaggio. Questa immagine esadecimale include le parti SCCP e TCAP, come mostrato nell'output.

```
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 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
```

Suggerimenti per la risoluzione dei problemi:

- Utilizzare il formato del messaggio SCCP mostrato nell'[Appendice C](#) per decodificare il tipo di messaggio, le informazioni dell'intestazione SCCP (mostrate nell'[output](#) in giallo) e l'inizio dei dati TCAP (mostrati nell'[output](#) in blu). L'`1e0002` nell'[output](#) rappresenta il codice del punto di destinazione da `dpc.dat` e il dump del messaggio SCCP inizia immediatamente dopo il tipo "1" (iniziando con il tipo di messaggio SCCP).
- Il PGW 2200 registra contatori e allarmi per eventi SCCP, TCAP e SS7. Se le misure sono abilitate, controllare i contatori per il messaggio TCAP. Verificare inoltre che SCCP, UDT e UDTs siano stati ricevuti e trasmessi. Consultare i seguenti documenti per le procedure operative MGC: [Gestione delle misurazioni del sistema](#) [Misurazioni Cisco MGC](#) [Recupero transazioni TCAP](#)
- Se il controller di canale SS7 non riceve il messaggio inviato da PGW 2200, verificare che TCAP abbia trasmesso un messaggio al CSCCP. Se il livello TCAP trasmette il messaggio

verso il basso, è possibile che il SCCP non disponga di informazioni sufficienti per creare il messaggio SCCP corretto. Ciò può anche indicare che il sottosistema SS7 non è stato predisposto correttamente o non è disponibile. Selezionare questo elenco per verificare: Configurazione e stato del codice punto SS7 Configurazione sottosistema SS7 Configurazione routing sottosistema SS7 Stato SSN locale e remoto In Configurazione servizio (trigger.dat) **Verifica del sistema**

```
mm1>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"
```

```
;
```

```
mm1> 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

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

"scpl: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 scpl

OPC = opcl

```

DPC = scp1

LNKSET = ls3

PRI = 1

*/

;

```

- Se tutte queste informazioni sembrano corrette (come mostrato nell'output mostrato sopra), verificare i valori con tag inviati dal livello del protocollo TCAP, ad esempio l'indirizzo SSN, SCCPCalledParty e/o SCCPCallingParty.

Messaggi TCAP che entrano in Cisco PGW 2200

La logica inversa può essere usata per tracciare un messaggio SS7 che arriva nel Cisco PGW 2200 e che è destinato al livello utente TCAP / SCCP dello stack SS7. I log di PGW 2200 mostrano il messaggio SS7 che arriva nel controller di canale SS7 (dalla linea SS7) e viene inviato a TCAP per l'elaborazione. Il messaggio viene suddiviso su ciascun layer dello stack SS7. Prendere nota anche di OPC/DPC, dell'indicatore di servizio (SIO) e della selezione del collegamento di segnalazione (SLS). OPC e DPC sono rappresentati in formato ITU (solo in questo esempio).

Suggerimento per la risoluzione dei problemi: Verificare il tipo di messaggio ricevuto dalla riga SS7. Se viene ricevuto un messaggio UDTs, selezionare la casella di controllo 'return cause'.

Questo output visualizza il log PGW 2200 quando riceve messaggi SCCP dalla riga SS7:

```

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 ffff3 09 08 67 04 ffff3 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 00 00 49 11 f6b 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 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

```

Suggerimento per la risoluzione dei problemi: Utilizzare il formato del messaggio SCCP mostrato nell'[Appendice C](#) per decodificare il tipo di messaggio, le informazioni dell'intestazione SCCP (mostrate nell'[output](#) in giallo) e l'inizio dei dati TCAP. L'1e0002 nell'output sopra riportato rappresenta l'indirizzo chiamante (OPC) per il messaggio ricevuto al PGW come rappresentato in dpc.dat. Il dump del messaggio SCCP inizia immediatamente dopo il valore "0" (iniziando con il tipo di messaggio SCCP).

Questo output viene generato dal log PGW 2200 quando riceve il TCAP UDTs su SCCP/MTP:

```
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>
```

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.

Questo output viene generato dal log PGW 2200 quando riceve un messaggio TCAP non valido su SCCP / MTP:

```
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
```

Strumento di traccia MDL

Cisco PGW 2200 utilizza trigger per avviare una transazione TCAP. Le transazioni del protocollo TCAP utilizzano il metodo `IN_TRIGGER` per inviare e ricevere messaggi da e verso il livello di controllo TCAP. Quando l'analisi delle chiamate raggiunge il tipo di risultato 22, il protocollo TCAP `IN_TRIGGER` viene inizializzato. Le informazioni/i messaggi TCAP vengono scambiati tra il livello del protocollo TCAP (ad esempio, i trigger scritti nel linguaggio MDL) e il processo del motore Cisco PGW 2200 utilizzando una tag, una lunghezza e un valore o una sintassi TLV. Il motore inoltra quindi le informazioni al controller di canale TCAP per un'ulteriore elaborazione.

Usare Cisco PGW 2200 MDL trace per visualizzare i dati inviati al controller TCAP (tramite il motore) dal e al livello del protocollo TCAP. Il controller di canale TCAP esegue l'elaborazione necessaria sui messaggi MDL ricevuti e li inoltra all'IOCC appropriato (TALI-IOCC, IP-IOCC o SS7-IOCC). Il modulo converte inoltre le informazioni sui messaggi TCAP ricevute dal controller di canale TCAP (tramite SCCP / MTP3) in un formato TLV che può essere passato al livello del protocollo TCAP, noto anche come `IN_TRIGGER`. Per tracciare una chiamata TCAP a livello di protocollo, attenersi alla seguente procedura:

1. Avvia una traccia MDL.

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

2. Effettua una chiamata che attiva un servizio TCAP (riscontra il tipo di risultati dell'analisi IN_TRIGGER).

3. Arrestare la traccia MDL.

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

```
MGC-01 - Media Gateway Controller 2004-03-24 17:41:04.702 EST
```

```
M COMPLD
```

```
"ALL:Trace stopped for the following files:
```

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

4. Eseguire `get_trc` per visualizzare la traccia MDL acquisita.

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

5. Eseguire l'opzione **S** per visualizzare una 'stampa simmetrica' della chiamata che indica il flusso di messaggi tra i processi interni PGW 2200.

6. Eseguire l'opzione **D** per visualizzare la traccia effettiva della chiamata tramite il codice PGW 2200. **Nota:** il contenuto visualizzato dalle opzioni **D** e **S** in `get_trc.sh` potrebbe non essere comprensibile in quanto i dati vengono visualizzati con i tipi di dati interni e i nomi delle variabili. Tuttavia, una descrizione degli elementi da cercare per eseguire il debug delle transazioni TCAP è disponibile nella sezione **Analisi traccia MDL per TCAP**.

Analisi di traccia MDL per TCAP

Utilizzare 'sim print' (opzione S di `get_trc.sh`) per visualizzare il flusso di chiamate complessivo a livello di protocollo Cisco PGW 2200. La stampa simmetrica è simile a quella mostrata nell'[Appendice D](#). In caso contrario, provare a prendere nota del punto in cui il flusso di chiamate derivato diverge e iniziare a risolvere il problema con l'evento. Per la risoluzione dei problemi TCAP, focalizzare l'attenzione su uno di questi eventi.

- LTrigger
- Informazioni sul trigger
- LTriggerSuccessivo
- LtriggerRelease

Questi sono gli eventi interni che guidano la macchina a stati IN_TRIGGER.

Utilizzare il trace Cisco PGW 2200 MDL per verificare il flusso di codice effettivo per ciascuno di questi eventi. LTrigger genera un OUTPUT IN_TRIGGER, mentre gli altri tre vengono inviati ricevuti da IN_TRIGGER tramite un messaggio INPUT IN_TRIGGER dal motore.

Messaggi TCAP in uscita

Per identificare i messaggi che entrano ed escono da MDL per TCAP, cercare IN_TRIGGER nella traccia MDL. La [sintassi di esempio IN_TRIGGER dell'immagine di traccia MDL](#) mostra un messaggio inviato e uno ricevuto in MDL da e verso il motore. L'OUTPUT indica che IN_TRIGGER ha inviato una richiesta al motore di inoltrare un messaggio TCAP.

Suggerimenti per la risoluzione dei problemi

- Utilizzare la traccia MDL per verificare che il messaggio `TRIGGER` sia stato inviato al motore se `IN_TRIGGER` o `OUTPUT` non è stato inviato.
- Controllare il dial plan per la configurazione dei risultati `IN_TRIGGER`.
- Controllare la configurazione `inservice` e / o `trigger.dat`.
- Verificare che il messaggio sia stato inviato dal controller di canale SS7. Se il messaggio non è mai uscito dal controller del canale SS7, è il risultato che il controller del canale SCCP non dispone di informazioni sufficienti per instradare la chiamata o generare un messaggio valido.
- Controllare la configurazione SCCP e la configurazione `SS7_SUBSYSTEM`.
- Controllare lo stato SSN.
- Controllare lo stato del PC.

Se l'output di `IN_TRIGGER` ha esito positivo, la traccia Cisco PGW 2200 MDL visualizza la risposta a tale messaggio come `INPUT` in `IN_TRIGGER`.

Esempio di sintassi `IN_TRIGGER` dalla traccia MDL

```
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 0c 00 01 03 00 0f  
00 01 01 00 13 00 0d 02 00 2a 81 76 82 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 1a 80 04 00 01 5f 91  
82 08 83 10 65 27 32 54 76 0f 83 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 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  
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 82 01 0a 81 01 01 a2 03 80 01 01 00 0a 00 00
```

Il messaggio `INPUT` è la risposta del motore in riferimento alla richiesta (o messaggio `OUTPUT`) inviata dal protocollo TCAP. Il motore può rispondere per conto proprio o per conto del livello TCAP.

Il messaggio `IN_TRIGGER` indica che MDL invia le informazioni TCAP/SCCP al motore e ai controller di canale per creare un messaggio UDT inviato sulla LINE all'SCP. Le informazioni inviate al motore derivano dal file `trigger.dat` e vengono visualizzate direttamente sopra l'output di questo messaggio. Per visualizzare il contenuto del messaggio generato da MDL, scorrere verso l'alto dal testo `IN_TRIGGER`. L'inizio della procedura di generazione del messaggio è indicato da `SendMessage()...`, come illustrato di seguito.

FUNCTION SendMessage() BEGIN

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

<messageData>.processId := bit(self, 32) -> '00000000000000000000000001101001'B

<messageData>.callRef := bit(CC.db.essentialData.releaseData.DATA.globalCallRefElem.DATA, 32)
-> '000000000000000000'

000000000000101'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

```

...

Suggerimenti per la risoluzione dei problemi

- Se si invia una query TCAP da Cisco PGW 2200 con dati non corretti, è possibile utilizzare la traccia MDL per verificare esattamente dove Cisco PGW 2200 ha derivato le proprie informazioni. La maggior parte delle informazioni proviene dal file trigger.dat. Per individuare la posizione in cui Cisco PGW 2200 ha ricavato le informazioni relative al messaggio in uscita, cercare (da `IN_TRIGGER`) l'elemento TCAP in questione. Ad esempio, se il tipo TCAP non è codificato correttamente, cercare la stringa `tcapType` nella traccia MDL (intorno al campo di scrittura `tcapType`).
- Per verificare dove Cisco PGW 2200 legge trigger.dat per codificare il contenuto TCAP, cercare le stringhe mostrate in questa tabella. Queste stringhe rappresentano le chiamate di routine utilizzate per recuperare le informazioni trigger.dat. Queste chiamate di routine devono avvenire tra l'evento `INPUT LTrigger` e il messaggio `OUTPUT IN_TRIGGER` in questione.

Nome	Descrizione	Stringa di ricerca MDL
TT	Record tabella trigger	GetTT

MA	Record azione messaggio	GetMA
MS	Record invio messaggio	GetMS
SO	Operazione Invio	GetOS
PS	Record di invio parametri	GetPS
RR	Record risposta ricevuta	GetRR
SIG.	Record ricezione messaggio	OttieniMR
O	Ricezione operazione	OttieniOR
PR	Record ricezione parametri	GetPR
RA	Record azione risposta	GetRA
AD	Dati azione	GetAD

Messaggi TCAP in ingresso

Il messaggio `INPUT` è la risposta del motore in riferimento alla richiesta. Il motore può rispondere per conto proprio o per conto del livello TCAP. Il messaggio in arrivo viene identificato dalla stringa del messaggio `INPUT IN_TRIGGER` nella traccia Cisco PGW 2200 MDL, come mostrato nell'output di esempio. In questo esempio viene inoltre visualizzato il messaggio decodificato. Questa opzione è utile se è necessario identificare eventuali problemi che possono verificarsi con la risposta TCAP.

Per decodificare il messaggio del motore ricevuto da Cisco PGW 2200 MDL, utilizzare lo stesso formato TLV descritto in precedenza in questo documento. Questi messaggi vengono decodificati immediatamente dopo il testo, `INPUT IN_TRIGGER`.

```
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
```

Di seguito viene riportato un esempio di output di una risposta in arrivo a un messaggio UDTs:

```
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 Continuing State Machine: IN_TRIGGER (105) STATE * INPUT
Information AS <messageData>
```



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

Un'altra utile informazione che è possibile ottenere dalla traccia Cisco PGW 2200 MDL (per chiamate TCAP) è il valore della causa `LTriggerRelease`. L'oggetto `INErrorElem` codificato in `LTriggerRelease` fornisce inoltre informazioni dettagliate sul motivo per cui una chiamata o una transazione TCAP non funziona come previsto. Vedere questa immagine Cisco PGW 2200 MDL che mostra una `LTriggerRelease` inviata in risposta all'evento `LTrigger` iniziale ricevuto da `IN_TRIGGER`. Vedere [Appendice E](#) per i dettagli sugli eventi `IN_TRIGGER` e sui valori `INErrorElem`.

```
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
```

[Appendice A Tag MDL](#)

I tag Cisco PGW 2200 MDL vengono scambiati tra il Cisco PGW 2200 MDL e il motore. In questa appendice vengono descritti l'ordine, il contenuto e il formato di tutti i tag utilizzati nelle transazioni TCAP. Le informazioni utilizzate per popolare questi valori di tag vengono ricavate dal contesto di chiamata e dai valori inseriti nel file `trigger.dat`. Il file di trigger viene inoltre utilizzato per indicare gli elementi da inviare al/dal motore per la creazione dei messaggi TCAP e gli elementi da ricevere dal motore per l'elaborazione dei messaggi TCAP quando viene ricevuta una risposta.

Questi tag vengono utilizzati per l'elaborazione delle chiamate TCAP:

- **ID TAG 1 - Tipo TCAP**
Descrizione: Indicazione del tipo di TCAP MDL
Lunghezza dati: fisso(1)
Formato dati:

1 = ETSI 300 374-1

2 = Bell Core GR-1298-CORE

TR-NWT-001284

3 = Bell Core Pre AIN

GR-1428-CORE

- **ID TAG 2 - Destinazione di sistema**Descrizione: Destinazione interna dell'eventoLunghezza dati: fisso(1)Formato dati: OttettoSommaro: 0 = SCP interno, 1 = TCAP trillium
- **ID TAG 3 - Indirizzo chiamato SCCP**Descrizione: Dati SCCP richiesti da trillioLunghezza dati: VariabileFormato dati:

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

- **ID TAG 4 - Indirizzo chiamante SCCP**Descrizione: Dati SCCP richiesti da trillioLunghezza dati: VariabileFormato dati:

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

- **ID TAG 5 - Tipo componente TCAPDescrizione:** Tipo di componente TCAP**Lunghezza dati:** fisso(1)**Formato dati:**

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

- **ID TAG 6 - Codice operazione TCAPDescrizione:** Codice operazione messaggio TCAP**Lunghezza dati:** Variabile (sempre 4 per ANSI)**Formato dati:**

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)

- **ID TAG 7 - ID chiamata TCAPDescrizione:** ID del componente**Lunghezza dati:** fisso(1)**Formato dati:** Ottetto
- **ID TAG 8 - ID correlazione TCAPDescrizione:** ID del componente a cui è correlato questo componente**Lunghezza dati:** fisso(1)**Formato dati:** Ottetto
- **ID TAG 9 - Componente finestra TCAP ANSIDescrizione:** Corpo di un messaggio TCAP dal primo parametro in poi**Lunghezza dati:** Variabile**Formato dati:** Ottetto
- **ID TAG 10 - Indicatore di fine finestra TCAPDescrizione:** Corpo di un messaggio TCAP dal primo parametro in poi (SEQUENCE)**Lunghezza dati:** fisso(0)**Formato dati:** Nessuna
- **ID TAG 11 - ErroreDescrizione:** Dati errore**Lunghezza dati:** fisso(1)**Formato dati:** Ottetto**Sommario:**
 - 1 = TCAP_ERROR_SSN_OOS
 - 2 = TCAP_ERROR_PC_UNAVAILABLE
 - 3 = TCAP_ERROR_SERVICE_NOT_RESPONDING
 - 4 = TCAP_TRIGGER_TIMEOUT
- **TAG ID 12 - Indice gruppo STP-SCPDDescrizione:** Indice gruppo STP-SCP, dati passati dall'analisi.**Lunghezza dati:** fisso(1)**Formato dati:** Ottetto**Sommario:** Valore di indice del gruppo STP-SCP.
- **TAG ID 13 - Protocollo di trasporto TCAPDescrizione:** Tipo di protocollo di trasporto**Lunghezza dati:** fisso(1)**Formato dati:** Ottetto**Sommario:**
 - 1 = TCAP_TRANSPORT_SCCP
 - 2 = TCAP_TRANSPORT_TCP_IP
- **ID TAG 14 - Errore esterno/problema TCAPDescrizione:** Valore di errore o problema ricevuto o inviato nei componenti Error & Result**Lunghezza dati:** Variabile**Formato dati:** Ottetto
- **ID TAG 15 - Tipo corpo TCAPDescrizione:** Tipo di corpo del componente**Lunghezza dati:**

fisso(1)Formato dati: OttettoSommario:

1 = TCAP_BODY_SEQUENCE

2 = TCAP_BODY_SET

- **ID TAG 16 - Informazioni finestra TCAP**Descrizione: TCAP include questo TAG in tutti i messaggi inviati a MDL. MDL dovrebbe memorizzare queste informazioni e inviarle al Trillium TCAP in tutti i messaggi successivi per la conversazione o i messaggi unidirezionali relativi alla chiamata.**Lunghezza dati:** Variabile**Formato dati:** Ottetto
- **ID TAG 17 - ID transazione TCAP**Descrizione: TCAP include questo TAG in tutti i messaggi inviati a MDL. MDL deve memorizzare queste informazioni per l'invio a CDB.**Lunghezza dati:** Variabile**Formato dati:** Ottetto
- **ID TAG 18 - ID database TCAP**Descrizione: TCAP Trillium includerà questo TAG in tutti i messaggi inviati a MDL. MDL deve memorizzare queste informazioni per l'invio a CDB.**Lunghezza dati:** Variabile**Formato dati:** Ottetto

Appendice B Disconnetti codici punto SS7

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.)

	Cluste r	Rete	Membr o	Codice punto
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 (senza spaziatura, solo 14 bit)	001	000 00001	001	001000 = 8 00000001 = 01
PC 1-4-1 (senza spaziatura, solo 14 bit)	001	0000010 0	001	001000 = 8 00100001 = 21
PC 3-3-3	011	0000001 1	011	011000 = 18 00011011 = 1B

Appendice C: Tipi di messaggio SCCP

Tipo di messaggio	Codice tipo messaggio
Richiesta di connessione CR	0000 0001
Conferma connessione CC	0000 0010

Connessione CREF rifiutata	0000 0011
RLSD rilasciato	0000 0100
Rilascio RLC completato	0000 0101
Modulo dati DT1 1	0000 0110
Modulo dati DT2 2	0000 0111
Conferma dati AK	0000 1000
UDT Unitdata	0000 1001
servizio Unitdata UDTS	0000 1010
Dati accelerati ED	0000 1011
Riconoscimento dati accelerato EA	0000 1100
RSR: Reimposta richiesta	0000 1101
RSC Conferma reimpostazione	0000 1110
Errore dell'unità dati del protocollo ERR	0000 1111
Test di inattività IT	0001 0000
XUDT Extended unitdata	0001 0001
Servizio Unity Data esteso XUDTS	0001 0010
LUDT Long unitdata	0001 0011
Servizio LUDTS Long Unit Data	0001 0100

Unitdata (UDT)

Il messaggio UDT contiene:

- Tre puntatori
- I parametri indicati nella tabella.

Parametro	Riferimento Q.713	Tipo (F V O)	Lunghezza (ottetti)
Tipo di messaggio	2.1	F	1
Classe Protocol	3.6	F	1
Indirizzo parte chiamata	3.4	V	minimo 3
Indirizzo del destinatario	3.5	V	minimo 3
Dati	3.16	V	2-X (Nota 1)

Nota: a causa degli studi in corso sull'indirizzo del destinatario della chiamata e su quello del destinatario della chiamata SCCP, la lunghezza massima di questo parametro richiede ulteriori studi. Si noti inoltre che il trasferimento di un massimo di 255 ottetti di dati utente è consentito quando il SCCP chiamato e l'indirizzo del chiamante non includono il titolo globale.

Servizio Unitdata (UDTS)

Il messaggio UDTS contiene:

- Tre puntatori.
- I parametri indicati nella tabella.

Parametro	Riferimento Q.713	Tipo (F V O)	Lunghezza (ottetti)
Tipo di messaggio	2.1	F	1
Causa del reso	3.12	F	1
Indirizzo parte chiamata	3.4	V	minimo 3
Indirizzo del destinatario	3.5	V	minimo 3
Dati	3.16	V	2-X (Nota)

Nota: a causa degli studi in corso sull'indirizzo del destinatario della chiamata e su quello del destinatario della chiamata SCCP, la lunghezza massima di questo parametro richiede ulteriori studi. Si noti inoltre che il trasferimento di un massimo di 255 ottetti di dati utente è consentito quando il SCCP chiamato e l'indirizzo del chiamante non includono il titolo globale.

Nella tabella seguente viene illustrato un esempio di suddivisione dei messaggi SCCP per il servizio Unitdata/Unitdata:

Parametro	Tipo (F V O)	Lunghezza (ottetti)	Messaggio di correlazione in uscita	Messaggio di correlazione in ingresso
Tipo di messaggio	F	1	09	0a
Classe Protocol	F	1	80	01
Puntatore indirizzo parte chiamata	F	1	03	03
Puntatore indirizzo destinatario chiamata	F	1	07	0g
Puntatore dati	F	1	0b	11
Indirizzo parte chiamata	V	minimo 3	04 c3 21 08 0c	04 c3 ... 30 00
Indirizzo del destinatario	V	minimo 3	04 c3 09 08 67	18 38 33 44 44
Dati (DATI TCAP)	V	04 c3 09 08 67 18 38 33 44 44 Dati (dati)	52 62 ... 20 00	29 62 ... 00 10

Nota: questi messaggi sono solo esempi e potrebbero non riflettere una combinazione/sequenza di risposta all'interrogazione effettiva.

Cause restituzione UDTs

Nel messaggio del servizio Unitdata, del servizio Extended Unitdata o del servizio Long Unitdata, il campo del parametro "return cause" è un campo di un ottetto che contiene il motivo per cui è stato restituito un messaggio. I bit da 1 a 8 sono codificati come mostrato di seguito:

Value	Bits	
0	0 0 0 0 0 0 0 0	no translation for an address of such nature
1	0 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

Appendice D: Interfaccia MDL per messaggio TCAP

Tutti i messaggi rispettano un formato TLV comune:

- **Chiamare Instance e ProcessId** - Lunghezza di 8 byte. Deve essere ricevuta dal motore e restituita nel messaggio di risposta dal motore inalterato.
- **ID messaggio:** identifica il messaggio inviato o ricevuto dal livello di protocollo TCAP (i valori sono mostrati in questa [tabella](#)).
- **ID tag** - Il numero di tag e i dati dei tag (ID tag, lunghezza dati e dati) determinano ciò che viene inviato nel messaggio TCAP alla destinazione remota. Tutte le dimensioni dei campi sono fisse, ad eccezione del campo dati di un elemento tag la cui lunghezza è variabile e viene definito (in ottetti) in base alla lunghezza dei dati. Ogni campo Lunghezza totale, Istanza di chiamata e ID processo, ID messaggio, ID tag e Lunghezza dati viene trasmesso per primo dal byte più significativo.

Appendice E: Interfaccia MDL interna

Internamente, la comunicazione con gli SMO (State Machine Objects) TCAP avviene tramite segnali con dati. Con il segnale è possibile inviare qualsiasi tipo di dati MDL. I nomi e il significato dei segnali e dei dati sono elencati qui.

- **LTriggerDescrizione:** Questo è il primo segnale inviato da LCM a TCAP per avviare la finestra di dialogo. In Elan, `INTriggerElem` contiene anche `stpScpGroupIndex`. Per utilizzare questa opzione, è necessario impostare `MSG_ACTION_COPY_STP_SCP_INDEX_FROM_SIGNAL_DATA` nella tabella MA.**Componenti:** `INTriggerElem`, `BNumberElem`, `BNumberDataElem`
- **Informazioni sul triggerDescrizione:** Questo segnale viene inviato da TCAP a LCM in risposta a `LTrigger`, quando il dialogo continua.**Componenti:** `INTriggerElem`, `BNumberElem`, `BNumberDataElem`
- **LTriggerSuccessivoDescrizione:** Questo segnale viene inviato da LCM a TCAP come successiva richiesta di attivazione in una finestra di dialogo esistente.**Componenti:** `INTriggerElem`, `BNumberElem`, `BNumberDataElem`
- **RilascioTriggerDescrizione:** Questo segnale è l'ultimo ad essere inviato da LCM o TCAP e può essere inviato da TCAP in risposta a `LTrigger` dopo la ricezione di una risposta da SCP.**Componenti:** `IErrorElem`, `BNumberElem`, `BNumberDataElem`In `INErrorElem` sono disponibili i seguenti valori:

- 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

Informazioni correlate

- [Cisco PGW 2200 Softswitch - Note tecniche](#)
- [Supporto alla tecnologia vocale](#)
- [Supporto ai prodotti voce e Unified Communications](#)
- [Risoluzione dei problemi di Cisco IP Telephony](#)
- [Documentazione e supporto tecnico – Cisco Systems](#)