

# Commutateur logiciel PGW 2200 TCAP 9.3 et versions ultérieures

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Informations générales](#)

[Résolution TCAP](#)

[Renifleur la ligne Ethernet](#)

[Suivi de Platform.log TCAP](#)

[Outil de suivi de MDL](#)

[Annexe A : Balises de MDL](#)

[Annexe B : Codes de point de la déconnexion SS7](#)

[Annexe C : Types de message de SCCP](#)

[Unitdata \(UDT\)](#)

[Service d'Unitdata \(UDTS\)](#)

[Causes de retour UDTS](#)

[Annexe D : Interface de MDL pour le message TCAP](#)

[Annexe E : Interface interne de MDL](#)

[Informations connexes](#)

## [Introduction](#)

La pièce d'applications de capacités de transaction (TCAP) fournit le support pour des applications interactives dans un environnement distribué. Le TCAP définit un protocole de bout en bout entre ses utilisateurs. Ceci peut se trouver dans un réseau SS7 ou un réseau différent qui prend en charge TCAP (IP).

## [Conditions préalables](#)

### [Conditions requises](#)

Les lecteurs de ce document devraient avoir la connaissance de :

- [Version 9 de contrôleur de passerelle de médias de Cisco](#)

### [Composants utilisés](#)

Les informations dans ce document sont basées sur le Commutateur logiciel Cisco PGW 2200.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

## Conventions

Pour plus d'informations sur les conventions de documents, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

## Informations générales

Le protocole TCAP se compose de deux sous-couches :

- Sous-couche composante
- Sous-couche de transaction

La sous-couche de composant se connecte par interface à l'engine de conversion. L'engine de conversion est l'équivalent d'un utilisateur de services ou d'un nombre de sous-système (SSN). La sous-couche composante prend en charge ces services :

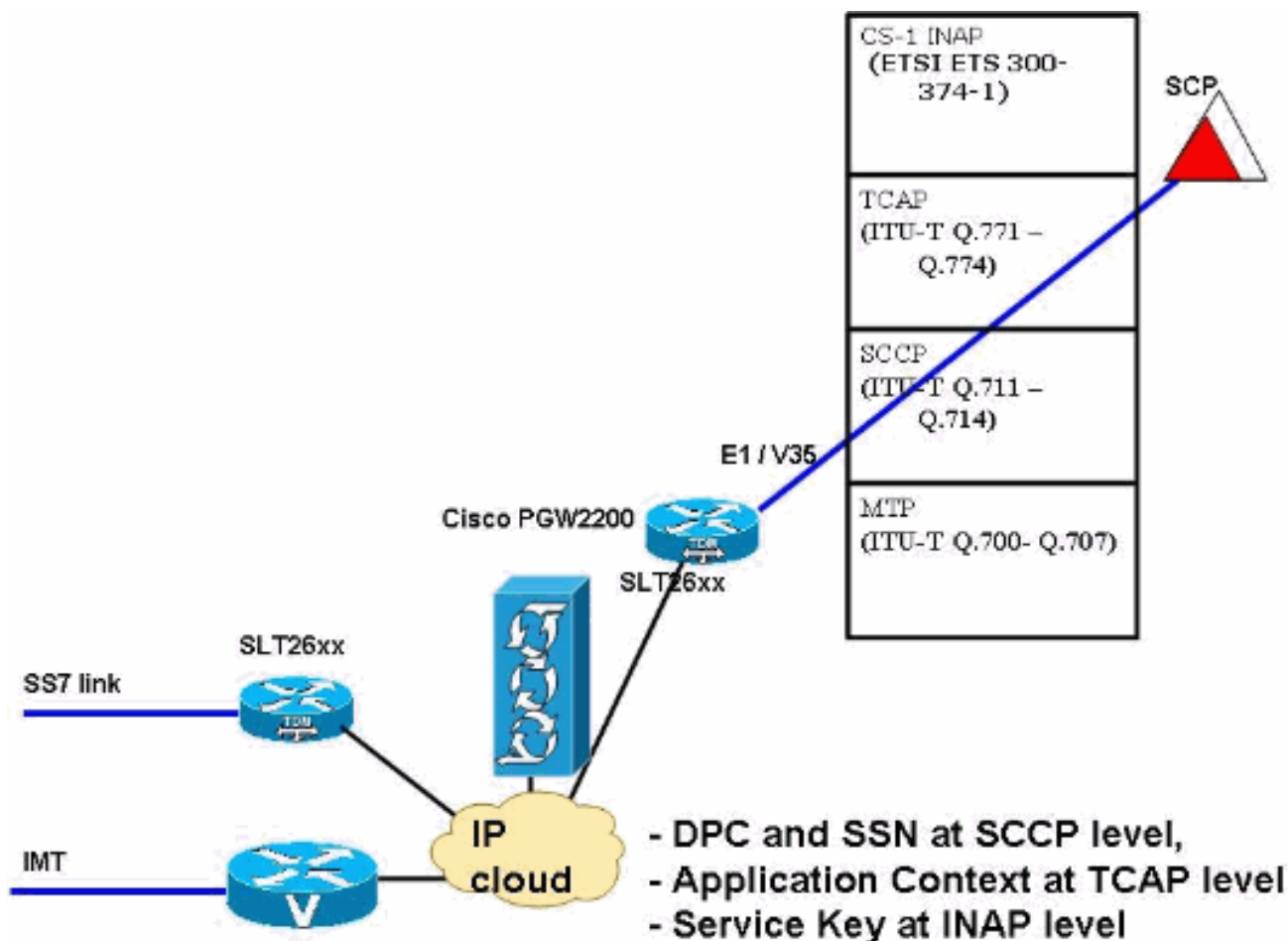
- Association des exécutions et des réponses.
- Manipulation anormale de situation.

La sous-couche de transaction se connecte par interface à signaler la partie commande de connexion (SCCP). Le TCAP prend en charge seulement un service réseau sans connexion. La sous-couche de transaction communique avec le SCCP par l'interface sans connexion.

Le logiciel TCAP emploie les services du logiciel de SCCP pour conduire les messages à l'utilisateur TCAP dans le noeud destinataire. L'interface entre le TCAP et le logiciel de SCCP est étroitement accouplée. Chaque demande TCAP de l'engine contient un titre global et un nombre de sous-système de destination. Le TCAP fournit le nombre de sous-système au SCCP pour la consultation de code des points de transfert de signal (STP). Si le SS7 adresse et des artères sont configurées correctement et complètement opérationnel, dépannez les informations de SCCP et TCAP passées et reçues entre Cisco PGW 2200 et un pair distant de SCCP ou TCAP.

Cisco PGW 2200 emploie le SCCP pour encapsuler des requêtes TCAP pour la pièce de transfert des messages de transport (MTP). Cette transmission de SCCP entre les pairs est présentée sans connexion au-dessus de MTP. Cisco PGW 2200 emploie le SCCP Unidata (UDT) pour envoyer des données au noeud distant de SCCP pour la transmission sans connexion. Le PGW 2200 reçoit une réponse valide quand le message du SCCP UDT est fourni avec succès. C'est typiquement sous forme de message UDT. L'échange de ces messages UDT facilite la transmission sans connexion entre le PGW 2200 et le pair distant de SCCP (tel que point de contrôle des services [SCP] pour des consultations de base de données TCAP). Le PGW 2200 définit un champ facultatif dans l'UDT qui énonce le pair de SCCP si « renvoyez sur l'erreur » le contenu de n'importe quel message qu'elle envoie au noeud distant si le message UDT est non livrable. Le message du service d'Unidata (UDTS) est utilisé pour faciliter cette réponse d'erreur. Le message UDTS indique au PGW 2200 qu'un message UDT reçu au noeud distant (tel que STP ou SCP) ne peut pas être fourni à la destination.

## **Installation de concept de Cisco PGW 2200**



## Résolution TCAP

La Messagerie de SCCP (UDT/UDTS) discutée dans la [section Informations générales](#) est essentielle quand vous dépannez des services et la fonctionnalité TCAP. Résolez tous les problèmes à la couche de SCCP avant que vous dépannez des données transmises TCAP ou les ayez reçu. Le format de l'UDT et du message UDTS est affiché dans l'[annexe C](#).

Utilisez ces outils de Cisco PGW 2200 pour mettre au point les appels qui exigent les services TCAP (TCAP/SCCP) :

- [Renifleur la ligne Ethernet](#) avec des outils tels que le fureteur éthéré, UNIX, et le fouineur.
- [Suivi de Platform.log TCAP](#) sur le PGW 2200.
- [Outil de suivi de MDL](#) pour le Traitement des appels au niveau de protocole.

## Renifleur la ligne Ethernet

Cisco PGW 2200 emploie l'UDP fiable (RUDP) pour envoyer MTP3 et messages supérieurs de la couche SS7 entre les périphériques MTP1 et MTP2 de gens du pays (tels qu'un terminal de lien de signalisation [SLT]). Cette transmission est typiquement faite au-dessus du port 7000 sur l'interface Ethernet de gens du pays de Cisco PGW 2200. C'est configurable. Référez-vous au [guide de configuration](#) pour des détails sur configurer les ports de « stPort » PGW dans XECfgParm.dat.

Vous pouvez employer n'importe quel renifleur d'Ethernets pour visualiser les paquets envoyés entre Cisco PGW 2200 et son périphérique de contrôle des gens du pays MTP2. Cependant, pas tous prennent en charge le protocole MTP et de SCCP utilisé pour afficher un message décodé.

Si un renifleur d'Ethernets n'est pas à la disposition du client, utilisez l'ordre de **fureteur** UNIX de dépanner. La sortie de l'ordre de **fureteur** n'est pas conviviale, mais est utile dans un scénario de le pire des cas.

Un renifleur d'Ethernets qui prend en charge la pile de protocoles SS7 est préféré. Il te permet pour décoder des paquets vus sur l'interface Ethernet de Cisco PGW 2200. Un renifleur ouvert de source tel qu'[éthéré](#) peut également être utilisé et est accessible en ligne.

Si aucun utilitaire commercial de renifleur n'est disponible, émettez l'ordre de **fureteur** sur la cible Cisco PGW 2200 de voir les données produites hexadécimales des messages envoyés à et de Cisco PGW 2200. Avec l'autorisation de racine sur Cisco PGW 2200, émettez cette commande de voir les données transmises hexadécimales hors du « stPort configuré. » Pour des informations supplémentaires sur l'ordre de **fureteur**, référez-vous aux « man page de fureteur ou aux guides administratifs de SUN.

```
#snoop -d <ethernet device name> -x 42 port <stPort>
```

Émettez cette commande de piller les paquets a envoyé le périphérique Ethernet, hmeX, sur le port 7000.

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

C'est exemple de sortie des paquets SS7 capturés avec l'ordre de **fureteur**.

```
#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 .....
```

Le fouineur de Cisco peut également être utilisé (si disponible) pour afficher le vidage hexadécimal du message de SCCP. L'en-tête de message de SCCP est décodée mais l'affichage de la sortie dépend de la version du fouineur choisie. Le point important est que le type de message est visible et donne une indication quant à où démarrer pour dépanner l'écoulement d'appel. Le vidage hexadécimal prouve que le type de message 09 est un message UDT et le type de message 0a est le message de service UDTS qui indique une erreur. La direction du flux des messages est également utile puisque les PC SS7 sont affichés. Si le reste du vidage hexadécimal est affiché que (dépend de la version de fouineur) elle peut être utilisée pour décoder plus loin les parties de SCCP et TCAP du message. Ceci est basé sur les standards de l'industrie pour le SCCP et le TCAP.

C'est la sortie de fouineur du message de SCCP UDT avec des données TCAP (au 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

```

S'il y a un message non livrable du SCCP UDT envoyé de Cisco PGW 2200 et/ou un SCCP (sur le noeud distant) a des problèmes avec le message, Cisco PGW 2200 reçoit un message de réponse UDTS. Ce message indique « une cause de retour » qui est très utile dans le dépannage. L'UDTS est hexa 10 (ou 0a) de type de message.

C'est un exemple d'un message de SCCP UDTS avec des données TCAP (du PSTN).

**Remarque:** Ce message est un exemple seulement et peut ne pas refléter une combinaison/ordre réels de réponse de requête. Le format et la quantité d'informations affichés varie selon la version de fouineur.

```

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.....

```

Cette sortie de fouineur affiche J'ordre SUIS, UDT, UDTS, et de VERSION.

**Remarque:** Ce message est un exemple seulement et peut ne pas refléter une combinaison/ordre réels de réponse de requête. Le format et la quantité d'informations affichés varie selon la version de fouineur.

```

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

```

C'est un tracé de renifleur SS7 qui inclut les informations du SCCP SS7 et 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	cs1-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

## Dépannez le conseil : Cause de retour UDTS

Pour un message UDTS, « la cause de retour » est le premier octet après le type de message 0a. Cette valeur aide à déterminer pourquoi le STP/SCP envoie une réponse d'erreur UDTS. Si ces informations ne sont pas visibles dans le renifleur, poursuivez à la section de [suivi de Platform.log TCAP](#) afin d'activer des suivis TCAP dans le log de Cisco PGW 2200.

## Suivi de Platform.log TCAP

MML permet à un utilisateur pour engager un suivi TCAP des messages de ce <Trace> de vidages mémoire pour le contrôleur de canal TCAP dans /opt/CiscoMGC/var/log/platform.log. Un suivi TCAP permet à l'utilisateur pour voir les messages TCAP/SCCP envoyés au contrôleur de canal SS7 pour conduire au commutateur SS7 au-dessus de MTP3. Voir l'[annexe E](#) pour le flux des messages d'une requête TCAP par le logiciel PGW 2200.

Le suivi TCAP est commencé par l'intermédiaire du mml avec la commande de **sta-TCAP-trc**. Afin de capturer le contrôleur des informations pertinentes, de debug logging d'enable pour le TCAP et de canal SS7.

C'est un exemple de la façon activer un suivi 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" ;
```

**Remarque:** Le debug logging peut exercer un effet sur la performance du système et ne devrait pas être utilisé dans un environnement de production sous le volume d'appels élevé. Veuillez prévoir votre fenêtre de maintenance en conséquence.

### Messages TCAP envoyés par Cisco PGW 2200

Une fois qu'un `IN_TRIGGER` est envoyé à l'engine, les êtres d'engine pour envoyer le message hors du PGW 2200. Les informations passées vers le bas du niveau de protocole sont transmises par relais au contrôleur de canal TCAP. La partie TCAP est envoyée vers le bas au contrôleur de canal de SCCP. En outre, un log est créé dans platform.log pour indiquer qu'un message TCAP « a été transmis ». Du message précédent UDT (affiché dans la partie de renifleur de ce document) vous pouvez voir comment le PGW 2200 se connecte relatif à l'information à ce même message dans platform.log. Ce log de plate-forme apparie le contenu de données affiché dans la [répartition de message de SCCP d'échantillon](#) : Table d'[Unitdata/service d'Unitdata](#) dans l'[annexe C](#). De cette table, la première valeur est la valeur de longueur des données (hexa 52 = décimale 82). La partie données de l'effectif TCAP suit la longueur de message. Au cas où le renifleur ou le fouineur ne serait pas disponible, ce platform.log peut être utilisé pour visualiser/met au point des transactions TCAP et de SCCP.

**Dépannez le conseil :** Si le message TCAP n'est pas envoyé vers le bas au SCCP, il y a un problème au MDL ou au niveau d'engine. Dépannez le suivi de MDL et regardez le signal de `Ltrigger` et de `LTriggerRelease`.

Cette sortie affiche que le log PGW 2200 envoyant le TCAP empilent vers le bas à 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 5f 91 82 8 83 10 65 27 32 54 76 f83 7 3
11 3 23 22 11 11 9a 2 20 0
```

Après que le TCAP envoie le message au SCCP, le contrôleur de canal SS7 lit le `MSG REÇU DU SCCP` et se connecte la représentation hexadécimale du message pour indiquer la réception du



message. Ce vidage hexadécimal inclut les parties de SCCP et TCAP suivant les indications de cette sortie.

```
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 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
```

## Dépannez les conseils :

- Utilisez le format de message de SCCP affiché dans l'[annexe C](#) pour décoder le type de message, les informations d'en-tête de SCCP (affichées dans la [sortie](#) en jaune) et le début des données TCAP (affichées dans la [sortie](#) dans le bleu). Le 1e0002 dans la [sortie](#) représente le code de point de destination de dpc.dat et le vidage mémoire de message de SCCP commence juste après le type "1" (début avec le type de message de SCCP).
- Le compteur et les alarmes de logs PGW 2200 pour le SCCP, les événements TCAP et SS7. Si des mesures sont activées, vérifiez les compteurs pour le message TCAP. Vérifiez également le SCCP, l'UDT, et l'UDTS reçu et transmis. Référez-vous à ces documents pour des procédures opérationnelles MGC. [Gérer des mesures de système](#) [Mesures de Cisco](#) [MGCRécupérer des transactions TCAP](#)
- Si le contrôleur de canal SS7 ne reçoit pas le message envoyé hors du PGW 2200, vérifiez que le TCAP a transmis un message vers le bas au SCCP. Si la couche TCAP transmet le message vers le bas, elle peut être parce que le SCCP n'a pas assez d'informations pour établir le message approprié de SCCP. Ceci peut également être une indication que le sous-système SS7 pas provisionné correctement ou n'est pas disponible. Vérifiez cette liste pour vérifier : Configuration et état de code du point SS7 Configuration du sous-système SS7 Configuration de routage du sous-système SS7 État local et distant SSNDANS la configuration de service (trigger.dat) **Vérification de système**

```
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"  
"itusn1:DPC=001.004.001,DNW=2:OPC=001.001.001:IS"  
"itusn2:DPC=001.003.001,DNW=2:OPC=001.001.001:IS"  
"itusn3:DPC=001.004.001,DNW=2:OPC=001.001.001:IS" ; mml> prov-rtrv:ss7subsys:NAME="itusn1"  
MGC-01 - Media Gateway Controller 2004-03-26 11:48:26.321 EST M RTRV  
"session=fix551tgp:ss7subsys" ; 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" ; mml> prov-rtrv:SS7ROUTE:NAME="route4" MGC-01 - Media Gateway Controller 2004-03-30 11:53:08.493 EST M RTRV "session=fix551tgp:SS7ROUTE" ;
```
- Si toutes ces informations semblent être correctes (suivant les indications de la sortie affichée ci-dessus) vérifiez les valeurs étiquetées envoyées vers le bas du niveau de protocole TCAP tel que le SSN, l'adresse de SCCPCalledParty et/ou l'adresse de SCCPCallingParty.

## Messages TCAP qui entrent dans Cisco PGW 2200

La logique inverse peut être utilisée pour tracer un message SS7 qui entre dans Cisco PGW 2200 qui est destiné couche TCAP/SCCP à utilisateur de la pile SS7. Les logs PGW 2200 affichent le message SS7 qui entre dans le contrôleur de canal SS7 (de la ligne SS7) et est envoyé au TCAP pour le traitement. Le message est décomposé à chaque couche de la pile SS7. En outre, notez l'OPC/DPC, l'indicateur de service (SIO) et la sélection de lien de signalisation (SLS). L'OPC et le DPC est représenté dans le format ITU (dans cet exemple seulement).

**Dépannez le conseil :** Vérifiez le type de message reçu de la ligne SS7. Si un message UDTs est recevez le contrôle que les « de retour entraînent ».

Cette sortie affiche le log PGW 2200 quand elle reçoit des messages de SCCP de la ligne 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 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 00 00 04 91 1f 6b 2a 28 28 67 01 11 86 51 11 1a 01 d6 11 b a1 d 6 b
2a 81 76 82 15 11 11 01 a2 32 10 a3 5a 13 21 16 c3 d a1 17 21 42 11 73 0f a0 d
30 b8 01 a8 11 0a 23 80 11 a1 22 21 52 12 33 01 a8 01 03 0e a0 c a0 a a1 5a 03 81
1 68 21 a8 11 1a 23 80 11
```

**Dépannez le conseil :** Utilisez le format de message de SCCP affiché dans l'[annexe C](#) pour décoder le type de message, les informations d'en-tête de SCCP (affichées dans la [sortie](#) en jaune) et le début des données TCAP. Le 1e0002 dans la sortie ci-dessus représente l'adresse appelante (OPC) pour le message reçu au PGW comme représenté dans dpc.dat. Le vidage mémoire de message de SCCP commence juste après le "0" (début avec le type de message de SCCP).

Cette sortie est du log PGW 2200 quand elle reçoit UDTs TCAP au-dessus de 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 ffffffff3 09 08 65 0a ffffffff8b 21 08 30 00 18 38 33 44 44 29 62 27 48 01 02 6c  
22 fffffffa1 20 02 01 01 02 01 00 30 18 ffffffff80 04 00 00 00 01 fffffff82 07 01 10 18 38 33 44 44  
fffffff83 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.**

Cette sortie est du log PGW 2200 quand elle reçoit un message non valide TCAP au-dessus du 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**

## Outil de suivi de MDL

Cisco PGW 2200 emploie des déclencheurs pour initier une transaction TCAP. Les transactions de protocole TCAP emploient la méthode `IN_TRIGGER` pour envoyer et recevoir des messages à et de la couche de contrôle TCAP. Quand l'analyse d'appel frappe le type 22 de résultat, le protocole `IN_TRIGGER` TCAP est initialisé. Les informations/messages TCAP sont permutés entre la couche de protocole TCAP (par exemple, des déclencheurs écrits en langage de MDL) et le processus d'engine de Cisco PGW 2200 utilisant une balise, une longueur, et une valeur ou une syntaxe

TLV. L'engine puis en avant les informations au contrôleur de canal TCAP pour une transformation plus ultérieure.

Employez le suivi de MDL de Cisco PGW 2200 pour voir les données qui sont envoyées à et de la couche de protocole TCAP au contrôleur TCAP (par l'intermédiaire de l'engine). Le contrôleur de canal TCAP fait le traitement nécessaire sur des messages de MDL reçus et en avant eux à l'IOCC approprié (TALI-IOCC, IP-IOCC ou SS7-IOCC). L'engine convertit également les informations de message TCAP reçues du contrôleur de canal TCAP (par l'intermédiaire de SCCP/MTP3) en format TLV qui peut être passé à la couche de protocole TCAP, également connu sous le nom d'`IN_TRIGGER`. Pour tracer un appel TCAP au niveau de protocole, terminez-vous ces étapes :

1. Commencez un suivi de MDL.

```
mml> sta-sc-trc:ss7svc1:log="udts",confirm
```
2. Faites un appel qui déclenche un service TCAP (type `IN_TRIGGER` de résultat d'analyse de hit).
3. Arrêtez le suivi de 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. Exécutez le `get_trc` pour visualiser le suivi capturé de MDL.

```
get_trc.sh udts_ss7svc2_20040324174103.btr
```
5. Exécutez l'option **S** de voir une « copie de sim » de l'appel qui affiche le flux des messages entre les processus internes PGW 2200.
6. Exécutez l'option **D** de voir le suivi réel de l'appel par le code PGW 2200.**Remarque:** Le contenu affiché par les options **D** et **S** dans `get_trc.sh` peut ne pas être évident pour comprendre pendant que les données sont affichées avec des types et des noms de la variable de données internes. Cependant une description de ce qu'à rechercher pour mettre au point des transactions TCAP est affichée dans l'**analyse de suivi de MDL pour la section TCAP**.

## Analyse de suivi de MDL pour le TCAP

Utilisation « copie de sim » (option S de `get_trc.sh`) de visualiser l'écoulement global d'appel au niveau de protocole de Cisco PGW 2200. La copie de sim ressemble à celui affiché dans l'[annexe D](#). S'il ne fait pas, l'essai pour noter où l'écoulement dérivé d'appel diverge et commencent à dépanner avec cet événement. Pour le dépannage TCAP, concentrez votre attention sur un de ces événements.

- LTrigger
- LTriggerInformation
- LTriggerNext
- LtriggerRelease

Ce sont les événements internes qui pilotent l'ordinateur d'état `IN_TRIGGER`.

Employez le suivi de MDL de Cisco PGW 2200 pour voir l'écoulement réel de code pour chacun de ces événements. LTrigger a comme conséquence un RÉSULTAT `IN_TRIGGER`, et les autres trois sont envoyés reçus par `IN_TRIGGER` par un message de l'ENTRÉE `IN_TRIGGER` de l'engine.

## Messages sortants TCAP

Pour identifier les messages qui sont livré dans et hors du MDL pour le TCAP, recherchez `IN_TRIGGER` dans le suivi de MDL. [La syntaxe de l'échantillon IN\\_TRIGGER du graphique de suivi de MDL](#) affiche un message envoyé et un reçu dans le MDL à et de l'engine. La `SORTIE` indique

qu'`IN_TRIGGER` a envoyé une demande de l'engine d'expédier un message TCAP.

## Conseils de dépannage

- Employez le suivi de MDL pour vérifier que le message de DÉCLENCHEUR a été envoyé à l'engine `s'IN_TRIGGER` ou `SORTIE` n'était pas envoyé.
- Vérifiez le dialplan pour la configuration de résultat `IN_TRIGGER`.
- Vérifiez l'inservice et/ou la configuration `trigger.dat`.
- Vérifiez que le message a été envoyé hors du contrôleur de canal SS7. Si le message ne le faisait jamais hors du contrôleur de canal SS7, c'est un résultat du contrôleur de canal de SCCP n'ayant pas assez d'informations pour conduire l'appel ou pour établir un message valide.
- Vérifiez la configuration de SCCP et la configuration `SS7_SUBSYSTEM`.
- Vérifiez l'état SSN.
- Vérifiez l'état PC.

Si la sortie de `IN_TRIGGER` est réussie, le suivi de MDL de Cisco PGW 2200 affiche la réponse à ce message comme `ENTRÉE` dans `IN_TRIGGER`.

## Syntaxe de l'échantillon `IN_TRIGGER` de suivi de 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 1d 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 e2 01 0a 81 01 01 a2 03 80 01 01 00 0a 00 00
```

Le message d'`ENTRÉE` est la réponse de l'engine en référence à la demande (ou au message de `SORTIE`) envoyé du protocole TCAP. L'engine peut répondre en son propre nom ou au nom de la couche TCAP.

Le message `IN_TRIGGER` indique que le MDL envoie les informations TCAP/SCCP vers le bas aux contrôleurs d'engine et de canal à utiliser pour construire un message UDT qui est envoyé sur la LIGNE au SCP. Les informations envoyées vers le bas à l'engine sont dérivées à partir du fichier `trigger.dat` et elles affichent directement au-dessus de la sortie de ce message. Pour voir le contenu de ce message car le MDL l'a établi, faites défiler du texte `IN_TRIGGER`. Le début de la procédure de bâtiment de message est indiqué par `SendMessage()`..., comme affiché ici.



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

```

...

## Conseils de dépannage

- Si une requête TCAP est envoyée hors de Cisco PGW 2200 avec des données incorrectes, le suivi de MDL peut être utilisé pour voir exactement où Cisco PGW 2200 a dérivé ses informations. La majeure partie des informations provient le fichier trigger.dat. Pour voir où Cisco PGW 2200 a dérivé ses informations pour le message sortant, recherchez (d'IN\_TRIGGER) l'élément TCAP en question. Par exemple, si le type TCAP est inexactement encodé, recherchez le tcapType en de chaîne le suivi de MDL (autour du tcapType de writingfield).
- Pour voir où Cisco PGW 2200 lit trigger.dat pour encoder le contenu TCAP, recherchez les chaînes affichées dans cette table. Ces chaînes représentent les appels de procédure utilisés pour récupérer les informations trigger.dat. Ces appels de procédure devraient se produire entre l'événement INPUTLTRIGGER et le message OUTPUTIN\_TRIGGER en question.

Nom	Description	Chaîne de recherche de MDL
-----	-------------	----------------------------



TTT	Enregistrement de Tableau de déclencheur	GetTT
MA	Enregistrement d'action de message	GetMA
MS	Message envoyant l'enregistrement	GetMS
SYSTÈME D'EXPLOITATION	Envoi d'exécution	GetOS
Picoseconde	Paramètre envoyant l'enregistrement	GetPS
Rr	Enregistrement reçu de réponse	GetRR
M.	Message recevant l'enregistrement	GetMR
OU	Réception d'exécution	GetOR
RP	Paramètre recevant l'enregistrement	GetPR
RA	Enregistrement d'action de réponse	GetRA
AD	Données d'action	GetAD

## Messages entrants TCAP

Le message d'**ENTRÉE** est la réponse de l'engine en référence à la demande. L'engine peut répondre en son propre nom ou au nom de la couche TCAP. Le message entrant est identifié par la chaîne de message de l'**ENTRÉE IN\_TRIGGER** dans le suivi de MDL de Cisco PGW 2200 suivant les indications de cet exemple de sortie. Cet exemple affiche également le message qui est décodé. C'est utile si vous devez identifier n'importe quels problèmes qui peuvent exister avec la réponse TCAP.

Pour décoder le message d'engine a reçu par MDL de Cisco PGW 2200, utilisent le même format TLV décrit plus tôt dans ce document. Ces le message sont décodés juste après le texte,

IN\_TRIGGER ENTRÉ.

```
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
```

C'est sortie témoin d'une réponse entrant à un message 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 ok Continuing State Machine: IN_TRIGGER (105) STATE * INPUT Information AS <messageData>
CC.db.nonEssentialData.TCAPTransactionUnixEndTimeElem.DATA :=
MGetTime(CC.db.nonEssentialData.TCAPTransactionMsecEndTimeElem.DATA) -> 1080257735
```

Une autre importante information que vous pouvez obtenir du suivi de MDL de Cisco PGW 2200 (pour des appels TCAP) est la valeur de cause de `LTriggerRelease`. `LINErrorElem` encodé dans le `LTriggerRelease` fournit également la vue dans pourquoi un appel ou la transaction TCAP ne fonctionne pas comme prévu. Voir le ce graphique de MDL de Cisco PGW 2200 qui affiche un `LTriggerRelease` qui est envoyé en réponse à l'événement de `LTrigger` d'initiale reçu par `IN_TRIGGER`. Voir l'[annexe E](#) pour des informations sur des événements `IN_TRIGGER` et des valeurs d'`INErrorElem`.

```

OD

END FUNCTION

VAR iNErrElem := NULL

iNErrElem.DATA.error := 42      → TRIG_ERROR_UNKNOWN

INSERT iNErrElem INTO <signalData>

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

FI

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

NEXTSTATE <state> -> STATE_WaitResponse

END INPUT

END STATE

```

## Annexe A : Balises de MDL

Les balises de MDL de Cisco PGW 2200 sont permutées entre le MDL de Cisco PGW 2200 et l'engine. Cette annexe décrit la commande, le contenu, et le format de toutes les balises utilisées dans des transactions TCAP. Les informations utilisées pour remplir ces valeurs de balise sont obtenues du contexte d'appel et les valeurs remplies dans le trigger.dat classent. Le fichier de déclencheur est également utilisé pour indiquer ce qui devrait être envoyé à/de l'engine pour le bâtiment de message TCAP et ce qui devrait être reçu de l'engine pour le traitement de messages TCAP quand une réponse est reçue.

Ces balises sont utilisées pour le Traitement des appels TCAP :

- **ID 1 de BALISE – Type TCAPDescription** : Indication du type de MDL TCAP  
**Longueur des données** : fixed(1)  
**Format des données** : 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
- **ID 2 de BALISE – Destination de systèmeDescription** : Destination interne d'événement  
**Longueur des données** : fixed(1)  
**Format des données** : Octet  
**Contenu** : 0 = SCP interne, 1 = Trillium TCAP
- **ID 3 de BALISE – Adresse appelée de SCCPDescription** : SCCP eus besoin par le trillium  
**Longueur des données** : Variable  
**Format des données** : 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 4 de BALISE – Adresse appelante de SCCP**Description : SCCP eus besoin par le trilliumLongueur des données : VariableFormat des données :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 5 de BALISE – Type de composant TCAP**Description : Type de composant TCAPLongueur des données : fixed(1)Format des données :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 6 de BALISE – Code opération TCAPDescription** : Code opération de message  
**TCAPLongueur des données** : Variable (toujours 4 pour l'ANSI)**Format des données** : 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 7 de BALISE – Le TCAP appellent l'IDDescription** : ID du composant  
**Longueur des données** : fixed(1)**Format des données** : Octet
- **ID 8 de BALISE – ID de corrélation TCAPDescription** : ID du composant au lequel ce composant le corrèle  
**Longueur des données** : fixed(1)**Format des données** : Octet
- **ID 9 de BALISE – ANSI de composant de dialogue TCAPDescription** : Corps d'un message TCAP de premier paramètre en avant  
**Longueur des données** : Variable**Format des données** : Octet
- **ID 10 de BALISE – Repère d'extrémité de dialogue TCAPDescription** : Corps d'un message TCAP de premier paramètre en avant (ORDRE)  
**Longueur des données** : fixed(0)**Format des données** : Aucun
- **ID 11 de BALISE – ErreurDescription** : Données d'erreur  
**Longueur des données** : fixed(1)**Format des données** : Octet  
**Contenu** : 1 = TCAP\_ERROR\_SSN\_OOS  
2 = TCAP\_ERROR\_PC\_UNAVAILABLE  
3 = TCAP\_ERROR\_SERVICE\_NOT\_RESPONDING  
4 = TCAP\_TRIGGER\_TIMEOUT
- **ID 12 de BALISE – Index de groupe STP-SCPDescription** : L'index de groupe STP-SCP, des données a passé de l'analyse.  
**Longueur des données** : fixed(1)**Format des données** : Octet  
**Contenu** : Valeur de l'indice de groupe STP-SCP.
- **ID 13 de BALISE – Protocole de transport TCAPDescription** : Type de protocole de transport  
**Longueur des données** : fixed(1)**Format des données** : Octet  
**Contenu** : 1 = TCAP\_TRANSPORT\_SCCP  
2 = TCAP\_TRANSPORT\_TCP\_IP
- **ID 14 de BALISE – Erreur externe/problème TCAPDescription** : Valeur d'erreur ou de problème reçue ou introduite des composants d'erreurs et de résultat  
**Longueur des données** : Variable  
**Format des données** : Octet
- **ID 15 de BALISE – Type de corps TCAPDescription** : Type de corps de composant  
**Longueur des données** : fixed(1)**Format des données** : Octet  
**Contenu** : 1 = TCAP\_BODY\_SEQUENCE  
2 = TCAP\_BODY\_SET
- **ID 16 de BALISE – Les informations de dialogue TCAPDescription** : Le Trillium TCAP inclut cette BALISE dans tous les messages envoyés au MDL. Le MDL devrait stocker ces informations et les envoyer au Trillium TCAP dans tous les messages ultérieurs pour le dialogue ou messages unidirectionnels liés à l'appel.  
**Longueur des données** : Variable  
**Format des données** : Octet

- **ID 17 de BALISE – Id de transaction TCAPDescription** : Le Trillium TCAP inclut cette BALISE dans tous les messages envoyés au MDL. Le MDL devrait stocker ces informations pour envoyer à la BDC.  
**Longueur des données** : Variable  
**Format des données** : Octet
- **ID 18 de BALISE – Id de base de données TCAPDescription** : Le Trillium TCAP inclura cette BALISE dans tous les messages envoyés au MDL. Le MDL devrait stocker ces informations pour envoyer à la BDC.  
**Longueur des données** : Variable  
**Format des données** : Octet

## [Annexe B : Codes de point de la déconnexion 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.)

	Batterie	Réseau	Membre	Code de point
ESTI (14 bits)	3 bits	8 bits	3 bits	14 bits
ANSI (24 bits)	8 bits	8 bits	8 bits	24 bits
PC 1-1-1 (aucune remplissage, bit 14 seulement)	001	000 00001	001	001000 = 8 00000001 = 01
PC 1-4-1 (aucune remplissage, bit 14 seulement)	001	0000010 0	001	001000 = 8 00100001 = 21
PC 3-3-3	011	0000001 1	011	011000 = 18 00011011 = 1B

## [Annexe C : Types de message de SCCP](#)

Type de message :	Type de message code
Demande de connexion de CR	0000 0001
Les cc de connexion confirment	0000 0010
Connexion CREF refusée	0000 0011
RLSD libéré	0000 0100
Release RLC complète	0000 0101
Forme de données DT1 1	0000 0110
Forme de données DT2 2	0000 0111
Accusé de réception de données AK	0000 1000
UDT Unitdata	0000 1001
Service UDTS Unitdata	0000 1010
Données exprès ED	0000 1011

Accusé de réception de données exprès ea	0000 1100
Demande de remise RSR	0000 1101
Confirmation de remise RSC	0000 1110
ERRENT l'erreur de Protocol Data Unit	0000 1111
Test informatique d'inactivité	0001 0000
Unitdata étendu XUDT	0001 0001
Service étendu d'unitdata XUDTS	0001 0010
Long unitdata LUDT	0001 0011
Long service d'unitdata LUDTS	0001 0100

## Unitdata (UDT)

Le message UDT contient :

- Trois pointeurs
- Les paramètres indiqués dans cette table.

Paramètre	Référence Q.713	Type (F V O)	Longueur (octets)
Type de message	2.1	F	1
Classe de protocole	3.6	F	1
Adresse d'appelé	3.4	V	3 minimum
Adresse d'appelant	3.5	V	3 minimum
Données	3.16	V	2-X (note 1)

**Remarque:** En raison des études actuelles sur le SCCP appelé et l'adresse d'appelant, la longueur maximale de ce paramètre nécessite davantage d'étude. On le note également qu'on permet le transfert de jusqu'à 255 octets de données d'utilisateur quand le SCCP appelé et l'adresse d'appelant n'incluent pas le titre global.

## Service d'Unitdata (UDTS)

Le message UDTS contient :

- Trois pointeurs.
- Les paramètres indiqués dans cette table.

Paramètre	Référence Q.713	Type (F V O)	Longueur (octets)
Type de message	2.1	F	1
Renvoyez la cause	3.12	F	1

Adresse d'appelé	3.4	V	3 minimum
Adresse d'appelant	3.5	V	3 minimum
Données	3.16	V	2-X (note)

**Remarque:** En raison des études actuelles sur le SCCP appelé et l'adresse d'appelant, la longueur maximale de ce paramètre nécessite davantage d'étude. On le note également qu'on permet le transfert de jusqu'à 255 octets de données d'utilisateur quand le SCCP appelé et l'adresse d'appelant n'incluent pas le titre global.

Cette table affiche une répartition de message de SCCP d'échantillon pour le service d'Unitdata/Unitdata :

Paramètre	Type (F V O)	Longueur (octets)	Message sortant de corrélation	Message entrant de corrélation
Type de message	F	1	09	0a
Classe de protocole	F	1	80	01
Pointeur d'adresse d'appelé	F	1	03	03
Pointeur d'adresse d'appelant	F	1	07	0d
Pointeur de données	F	1	0b	11
Adresse d'appelé	V	3 minimum	04 c3 21 08 0c	04 c3... 30 00
Adresse d'appelant	V	3 minimum	04 c3 09 08 67	18 38 33 44 44
Données (DONNÉES TCAP)	V	04 c3 09 08 67 18 38 33 44 44 données (DONNÉES TCAP)	52 62... 20 00	29 62... 00 10

**Remarque:** Ces messages sont des exemples seulement et peuvent ne pas refléter une combinaison/ordre réels de réponse de requête.



## Causes de retour UDTs

Dans le service d'Unitdata, le service étendu d'Unitdata, ou le long message de service d'Unitdata, le champ de paramètre « de cause de retour » est un champ d'un octet qui contient la raison pour un retour de message. Les bits 1 à 8 sont codés comme affiché ici :

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

## Annexe D : Interface de MDL pour le message TCAP

Tous les messages adhèrent à un format commun TLV :

- **L'exemple et le ProcessId d'appel** - 8 octets de long et devraient être reçus par l'engine et être retournés dans le message de réponse de l'engine inchangée.
- **ID de message** - Identifie le message qui est envoyé ou reçu par la couche de protocole TCAP (valeurs affichées dans cette [table](#)).
- **Id étiqueté** - Le nombre de balises et de données de balise (ID, longueur des données et données de balise) dictent ce qui est envoyé dans le message TCAP à la destination distante. Toutes les tailles de champ sont réparées excepté la zone d'information d'un élément de balise dont la longueur est variable et est définie (dans les octets) par la longueur des données. Chacun des champs longueur totale, exemple d'appel et identificateur de processus, id de message, id de balise et longueur des données est transmis par l'octet de poids fort d'abord.

## Annexe E : Interface interne de MDL

Intérieurement, la transmission avec des objets d'ordinateur d'état TCAP (SMOs) est par des signaux avec des données. N'importe quel type de données de MDL peut être envoyé avec le signal. Les noms et les significations des signaux et des données sont répertoriés ici.

- **LTriggerDescription** : C'est le premier signal que LCM envoie au TCAP pour commencer le dialogue. Dans l'elan, INTriggerElem contient également le stpScpGroupIndex. MSG\_ACTION\_COPY\_STP\_SCP\_INDEX\_FROM\_SIGNAL\_DATA doit être placé dans la table mA pour que ceci soit utilisé. **Composants** : INTriggerElem, BNumberElem, BNumberDataElem
- **LTriggerInformationDescription** : Ce signal est envoyé du TCAP à LCM en réponse à LTrigger, quand le dialogue continue. **Composants** : INTriggerElem, BNumberElem, BNumberDataElem
- **LTriggerNextDescription** : Ce signal est envoyé de LCM au TCAP comme demande ultérieure de déclencheur dans un dialogue existant. **Composants** : INTriggerElem, BNumberElem, BNumberDataElem
- **LTriggerReleaseDescription** : Ce signal est le bout à envoyer de LCM ou de TCAP et peut être envoyé du TCAP en réponse à LTrigger après qu'une réponse ait été reçue du SCP. **Composants** : INErrorElem, BNumberElem, BNumberDataElem

INErrorElem a ces valeurs

```

: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\_INVALLD\_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

## **Informations connexes**

- [Notes en tech de Commutateur logiciel Cisco PGW 2200](#)
- [Assistance technique concernant la technologie vocale](#)
- [Assistance concernant les produits vocaux et de communications unifiées](#)
- [Dépannage des problèmes de téléphonie IP Cisco](#)
- [Support et documentation techniques - Cisco Systems](#)