

# PGW 2200 ソフトスイッチ TCAP リリース 9.3 以降

## 目次

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[背景説明](#)

[TCAP 解決](#)

[イーサネット ラインのスニファ](#)

[Platform.log TCAP トレース](#)

[MDL トレース ツール](#)

[付録 A : MDL タグ](#)

[付録 B : SS7 ポイント コードのログオフ](#)

[付録 C : SCCP メッセージ タイプ](#)

[ユニットデータ \( UDT \)](#)

[ユニットデータ サービス \( UDTS \)](#)

[UDTS リターンの原因](#)

[付録 D : TCAP メッセージ用 MDL インターフェイス](#)

[付録 E : 内部 MDL インターフェイス](#)

[関連情報](#)

## 概要

トランザクション機能応用部 ( TCAP ) は、分散環境のインタラクティブ アプリケーションをサポートします。 TCAP はユーザ間のエンドツーエンド プロトコルを定義します。これは、SS7 ネットワークまたは TCAP ( IP ) をサポートする別のネットワークに配置されている可能性があります。

## 前提条件

### 要件

このドキュメントを読むには、次の知識が必要です。

- [Cisco Media Gateway Controller リリース 9](#)

### [使用するコンポーネント](#)

このドキュメントの情報は、Cisco PGW 2200 ソフトスイッチに基づくものです。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。ネットワークが稼働中の場合は、コマンドが及ぼす潜在的な影響を十分に理解しておく必要があります。

## [表記法](#)

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

## [背景説明](#)

TCAP のプロトコルは、次の 2 つのサブレイヤで構成されます。

- コンポーネント サブレイヤ
- トランザクション サブレイヤ

コンポーネント サブレイヤは、変換エンジンとインターフェイスします。変換エンジンは、サービス ユーザまたはサブシステム番号 (SSN) に相当します。コンポーネント サブレイヤは、次のサービスをサポートします。

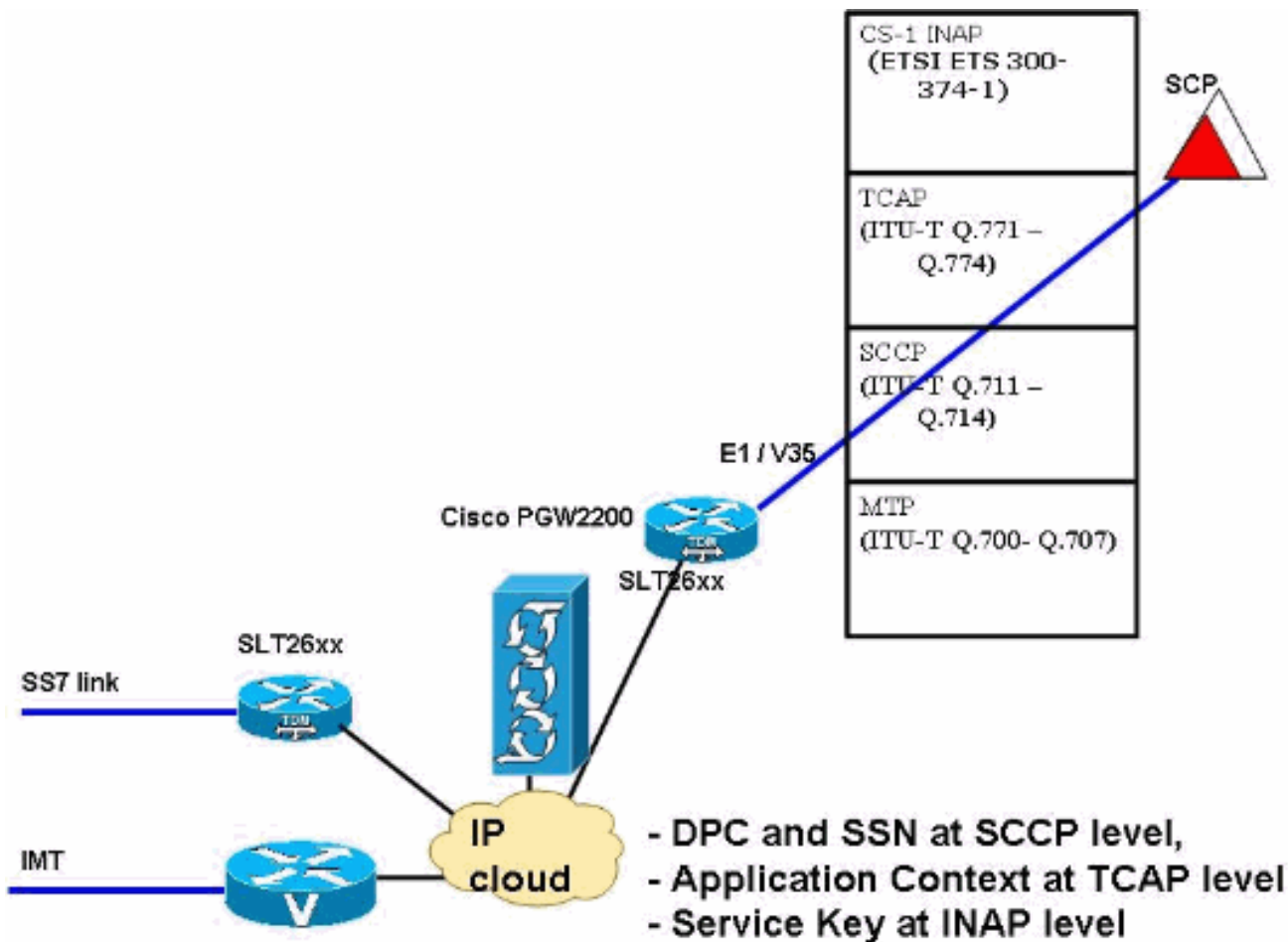
- 操作と応答の関連付け。
- 異常事態の処理。

トランザクション サブレイヤは、信号接続制御部 (SCCP) とインターフェイスします。TCAP はコネクションレス型ネットワーク サービスのみをサポートします。トランザクション サブレイヤは、コネクションレス型インターフェイス経由で SCCP と通信します。

TCAP ソフトウェアは、SCCP ソフトウェアのサービスを使用して宛先ノードの TCAP のユーザにメッセージをルーティングします。TCAP と SCCP ソフトウェア間のインターフェイスは強固に結び付けられています。エンジンからの各 TCAP 要求には、グローバル タイトルと宛先のサブシステム番号が含まれています。TCAP は信号中継局 (STP) のコード検索のため、SCCP にサブシステム番号を提供します。SS7 アドレスとルートが正しく設定されていて完全に動作可能である場合は、Cisco PGW 2200 とリモート SCCP または TCAP のピア間でやり取りされた SCCP および TCAP 情報をトラブルシューティングします。

Cisco PGW 2200 は、メッセージ転送部 (MTP) から転送される TCAP クエリーのカプセル化に SCCP を使用します。このピア間の SCCP 通信は、MTP 経由でコネクションレスで送信されます。Cisco PGW 2200 は SCCP ユニットデータ (UDT) を使用し、コネクションレス型通信のためにリモート SCCP ノードにデータを送信します。PGW 2200 は、SCCP UDT メッセージが正常に配信されると有効な応答を受信します。これは通常、UDT メッセージの形式になります。これらの UDT メッセージの交換は、PGW 2200 とリモート SCCP ピア (TCAP データベース 検索用のサービス コントロール ポイント (SCP) など) との間のコネクションレス型通信を容易にします。PGW 2200 は UDT メッセージが配信不能な場合、SCCP ピアがリモート ノードに送信するすべてのメッセージの内容を「エラー時に返す」ように指定する UDT のオプション フィールドを定義します。ユニットデータ サービス (UDTS) メッセージは、このエラー応答を容易に実行できるようにするために使用されます。UDTS メッセージは、リモート ノードで受信される UDT メッセージ (STP または SCP など) が宛先に配信できないことを PGW 2200 に示します。

## Cisco PGW 2200 概念の設定



## TCAP 解決

「[背景説明](#)」セクションに記載されている SCCP メッセージ ( UDT/UDTS ) は、TCAP のサービスと機能のトラブルシューティングにおいて重要です。送受信した TCAP データのトラブルシューティングを行う前に、SCCP レイヤの問題を解決してください。UDT と UDTS メッセージの形式は、「[付録 C](#)」に示すとおりです。

TCAP ( TCAP/SCCP ) サービスを必要とするコールをデバッグするには、次の Cisco PGW 2200 ツールを使用してください。

- Ethereal、UNIX snoop、Snooper などのツールを使用した[イーサネットラインのスニファ](#)。
- PGW 2200 上の [Platform.log TCAP トレース](#)。
- プロトコルレベルのコール処理に向けた [MDL トレース ツール](#)。

## イーサネットラインのスニファ

Cisco PGW 2200 は信頼できる UDP ( RUDP ) を使用し、ローカル MTP1 と MTP2 デバイス ( Signaling Link Terminal ( SLT ) など ) 間で MTP3 や上位レイヤの SS7 メッセージを送信します。この通信は通常、Cisco PGW 2200 ローカルイーサネットインターフェイスのポート 7000 経由で行われます。これは、設定可能です。XECfgParm.dat で PGW 「stPort」ポートを設定する方法の詳細については、「[コンフィギュレーションガイド](#)」を参照してください。

Cisco PGW 2200 とローカル MTP2 制御デバイス間で送信されたパケットを表示するには、任意のイーサネットスニファを使用できます。ただし、すべての MTP および SCCP プロトコルがデコードされたメッセージの表示をサポートしているとは限りません。イーサネットスニファを使用できない場合は、トラブルシューティングに UNIX の snoop コマンドを使用します。snoop コ

マンドの出力はユーザにとってわかりやすいものではありませんが、最悪のシナリオでは役に立ちます。

SS7 プロトコル スタックをサポートするイーサネット スニファが推奨されます。これは、Cisco PGW 2200 イーサネット インターフェイスで表示されるパケットをデコードできるようにします。また、オンラインで入手できる [Ethereal](#) などのオープン ソースのスニファも使用できます。

市販のスニファ ユーティリティを使用できない場合は、ターゲットの Cisco PGW 2200 で **snoop** コマンドを発行し、Cisco PGW 2200 で送受信されるメッセージの 16 進数のデータ出力を表示します。Cisco PGW 2200 のルート権限を使用して次のコマンドを発行し、設定された「stPort」から送信される 16 進数のデータを表示します。**snoop** コマンドの詳細については、『SUN Administrative』ガイドの「snoop man pages」を参照してください。

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

次のコマンドを発行し、イーサネット デバイス、hmeX、ポート 7000 で送信されたパケットをスヌープします。

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

次は、**snoop** コマンドを使用してキャプチャされた SS7 パケットの出力例です。

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

Cisco の Snooper ( 使用可能な場合 ) を使用して SCCP メッセージの 16 進数ダンプを表示することもできます。SCCP メッセージヘッダーはデコードされますが、出力の表示は選択した Snooper のバージョンによって異なります。重要な点は、メッセージタイプが表示されて、コールフローのトラブルシューティングをどこで開始するべきかが示されることです。16 進数ダンプに表示されるメッセージタイプ 09 の UDT メッセージ、メッセージタイプ 0a は UDTs サービスメッセージのエラーを示します。また、SS7 の PC が表示されるため、メッセージフローの方向も役に立ちます。16 進数ダンプの残りの部分が表示されている場合 ( Snooper のバージョンにより異なる )、メッセージの SCCP と TCAP の一部をさらにデコードするために使用できません。これは、SCCP と TCAP の業界標準に基づいています。

次は、TCAP データを含む UDT SCCP メッセージ ( PSTN に送信 ) の Snooper の出力です。



```

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 '2T▼.....#*....
20 00

```

Cisco PGW 2200 から送信された配信不可能な SCCP UDT メッセージがある場合や SCCP (リモートノード上) でメッセージに関する問題がある場合、Cisco PGW 2200 は UDTS 応答メッセージを受信します。このメッセージは、トラブルシューティングに非常に役立つ「リターンの原因」を示します。UDTS はメッセージタイプ 10 (または 16 進数で 0a) です。

次は、TCAP データを含む UDT SCCP メッセージ (PSTN から受信) の例です。

注: このメッセージは単なる例であり、実際のクエリー応答の組み合わせとシーケンスが反映されているとは限りません。表示される形式および情報量は、Snooper のバージョンによって異なります。

```

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...I.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 *.▼.....

```

この Snooper の出力は IAM、UDT、UDTS、および REL のシーケンスを表示します。

注: このメッセージは単なる例であり、実際のクエリー応答の組み合わせとシーケンスが反映されているとは限りません。表示される形式および情報量は、Snooper のバージョンによって異なります。

```

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

```

これは、SS7 の SCCP と TCAP 情報を含む SS7 スニファトレースです。

```

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

```

## トラブルシューティングのヒント：UDTS リターンの原因

UDTS メッセージの場合、「リターンの原因」はメッセージ タイプ 0a の後の最初のバイトです。この値は、STP や SCP が UDTS エラー応答を送信した理由を特定するのに役立ちます。この情報はスニファには表示されないため、「[Platform.log TCAP トレース](#)」セクションに進み、Cisco PGW 2200 ログの TCAP トレースを有効にしてください。

## Platform.log TCAP トレース

MML はユーザが TCAP トレースを開始し、TCAP チャンネル コントローラの <Trace> メッセージを /opt/CiscoMGC/var/log/platform.log にダンプできるようにします。TCAP トレースはユーザが SS7 チャンネル コントローラに送信される TCAP や SCCP メッセージを確認し、MTP3 経由で SS7 スイッチにルーティングできるようにします。PGW 2200 ソフトウェアを通過する TCAP クエリーのメッセージ フローについては、「[付録 E](#)」を参照してください。

TCAP トレースは `sta-tcap-trc` コマンドを使用して mml を介して開始されます。関連情報をキャプチャするには、TCAP と SS7 チャンネル コントローラのデバッグ ログを有効にします。

次の例は、TCAP トレースを有効にする方法を示します。

```
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" ;
```

注: デバッグ ログはシステムのパフォーマンスに影響する可能性があるため、コールの量が多い実稼働環境では使用しないでください。必要に応じて、メンテナンス ウィンドウを計画してください。

### Cisco PGW 2200 によって送信された TCAP メッセージ

IN\_TRIGGER がエンジンに送信されると、エンジンは PGW 2200 からメッセージの送信を開始します。プロトコル レベルから渡される情報は、TCAP チャンネル コントローラにリレーされます。TCAP の部分は、SCCP チャンネル コントローラに送信されます。また、TCAP メッセージが「送信された」ことを示すログが platform.log で作成されます。前の UDT メッセージ (このドキュメントのスニファ部分に表示される) から、PGW 2200 が platform.log にこのメッセージの関連情報をどのように記録するかを確認できます。このプラットフォーム ログは、「[サンプル SCCP メッセージの詳細: ユニットデータ/ユニットデータ サービス](#)」テーブル内 («[付録 C](#)») のデータ コンテンツと一致します。このテーブルの最初の値は、データ長さの値 (16 進数で 52 = 10 進数で 82) です。実際の TCAP のデータ部分がメッセージの長さの後に続きます。スニファまたは Snooper を使用できない場合は、この platform.log を使用して、TCAP と SCCP トランザクションの表示とデバッグを実行できます。

トラブルシューティングのヒント: TCAP メッセージが SCCP に送信されない場合、MDL またはエンジン レベルに問題があります。MDL トレースのトラブルシューティングを行い、Ltrigger と LTriggerRelease 信号を調べます。

この出力は、TCAP ダウン スタックを SCCP または MTP に送信する PGW 2200 ログを示します。

```
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
```

TCAP が SCCP にメッセージを送信した後、SS7 チャンネル コントローラは、RECEIVED MSG FROM SCCP を再生し、メッセージの受信を示すためにメッセージの 16 進数の表現をログに記録します。この 16 進数ダンプは、次の出力に示すように SCCP および TCAP の一部を含みます

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 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

## トラブルシューティングのヒント :

- 「[付録 C](#)」に示す SCCP メッセージ形式を使用し、メッセージ タイプ、SCCP ヘッダー情報 ( 「[出力](#)」に黄色で表示 ) および TCAP データの開始部 ( 「[出力](#)」に青色で表示 ) をデコードします。「[出力](#)」の 1e0002 は、dpc.dat からの宛先ポイント コードを表し、SCCP メッセージ ダンプはタイプ「1」の直後に開始されます ( 先頭は SCCP メッセージ タイプです )。
- PGW 2200 は SCCP、TCAP および SS7 イベントのカウンタとアラームをログに記録します。測定が有効になっている場合は、TCAP メッセージのカウンタを確認します。また、送信される SCCP、UDT、および UDTS を確認します。MGC の操作手順については、次のドキュメントを参照してください。[システム測定の管理Cisco MGC の測定TCAP のトランザクションの取得](#)
- SS7 チャネル コントローラが PGW 2200 から送信されるメッセージを受信しない場合は、TCAP が SCCP にメッセージを送信したかどうかを確認します。TCAP レイヤからメッセージが送信される場合、正常な SCCP メッセージを作成するために十分な情報が SCCP がないことが原因である可能性があります。これは、SS7 サブシステムが適切にプロビジョニングされていないか、使用できないことを示している可能性があります。確認するには、次のリストをチェックしてください。SS7 ポイント コードの設定と状態SS7 サブシステムの設定 SS7 サブシステムのルーティングの設定ローカルおよびリモート SSN の状態IN Service 設定 ( trigger.dat ) **システムの確認**

```
mml>rtrv-spc:all MGC-01 - Media Gateway Controller 2004-03-26 13:22:05.492 EST M RTRV "ss7svc1:DPC=001.022.001,DNW=2:OPC=001.001.001:IS"
"ss7svc2:DPC=001.022.002,DNW=2:OPC=001.001.001:IS"
"itussn1:DPC=001.004.001,DNW=2:OPC=001.001.001:IS"
"itussn2:DPC=001.003.001,DNW=2:OPC=001.001.001:IS"
"itussn3:DPC=001.004.001,DNW=2:OPC=001.001.001:IS" ; mml> prov-rtrv:ss7subsys:NAME="itussn1"
MGC-01 - Media Gateway Controller 2004-03-26 11:48:26.321 EST M RTRV
"session=fix551tgp:ss7subsys" ; mml> rtrv-ssn: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" ; mml> prov-rtrv:SS7ROUTE:NAME="route4" MGC-01 - Media Gateway Controller 2004-03-30 11:53:08.493 EST M RTRV "session=fix551tgp:SS7ROUTE" ;
```
- これらの情報がすべてが正しいと思われる場合は ( 上の出力に示すように )、SSN、SCCPCalledParty アドレスや SCCPCallingParty アドレスなどの TCAP プロトコル レベルから送信されたタグ付き値を確認します。

## Cisco PGW 2200 が受信する TCAP メッセージ

逆のロジックを使用し、SS7 スタックの TCAP または SCCP ユーザレイヤを宛先とする Cisco PGW 2200 が受信する SS7 メッセージをトレースすることができます。PGW 2200 ログは、SS7 チャネルコントローラが受信して (SS7 回線から)、処理のために TCAP に送信する SS7 メッセージを表示します。メッセージは、SS7 スタックの各レイヤで分割されます。また、OPC と DPC、サービスインジケータ (SIO) と信号リンク選択 (SLS) にも注目します。OPC と DPC は、ITU 形式で表されています (この例のみ)。

**トラブルシューティングのヒント:** SS7 回線から受信したメッセージタイプを確認します。UDTS メッセージを受信した場合は、「リターンの原因」を確認します。

この出力は、SS7 回線から SCCP メッセージを受信したときの PGW 2200 ログを示します。

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

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

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

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

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

<lines omitted>

```
Thu Dec 4 15:23:04:954 2003 EST | TCAP (PID 9513) <Trace>  
PROT_TRACE_TCAP_PDU_RX: Hex dump of TCAP message received, SSN=103, LEN=118,  
65 74 48 45 0 0 0 49 11 f6 b2 a28 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 fa0 d  
30 b 80 1 a81 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 a81 1 1 a2 3 80 1 1
```

**トラブルシューティングのヒント:** 「[付録 C](#)」に示す SCCP メッセージ形式を使用してメッセージタイプ、SCCP ヘッダー情報 (「[出力](#)」に黄色で表示) および TCAP のデータの開始部をデコードします。上の出力の 1e0002 は、dpc.dat に示されているように、PGW での受信メッセージの発信側アドレス (OPC) を表します。SCCP メッセージダンプは「0」の直後に開始されず (先頭は SCCP メッセージタイプです)。

この出力は、SCCP または MTP 経由で UDTS TCAP を受信したときの PGW 2200 ログから取得したものです。

```
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
ffffff83 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.

```

この出力は、SCCP または MTP 経由で無効な TCAP を受信したときの PGW 2200 ログから取得したものです。

```

Tue Mar 23 16:24:51:565 2004 EST | ss7-i-1 (PID 22997) <Trace>
PROT_TRACE_MTP3_PDU: Hex dump of MTP3 and UP messages
1d0005 0 CP DATA IND len: 12 data: 83 09 48 08 02 0a ←msgtype 10= UDTS

```

```

Tue Mar 23 16:24:51:565 2004 EST | ss7-i-1 (PID 22997) <Debug>
>>>> from: 821 to opc 809 (bytes 7) sio 83 sls 0:

```

```

Tue Mar 23 16:24:51:565 2004 EST | ss7-i-1 (PID 22997) <Trace>
PROT_TRACE_MTP3_PDU: Hex dump of MTP3 and UP messages
1e0002 0 0a 03 00 00 00 00 00 ←Msg Type 10 (UDTS), Return cause = 03 =
<lines omitted>

```

```

Tue Mar 23 16:24:51:565 2004 EST | ss7-i-1 (PID 22997) <Debug>
RECEIVED SCCP STACK MSG
<lines omitted>

```

```

Tue Mar 23 16:24:51:566 2004 EST | TCAP (PID 22992) <Debug>
00 01 00 01 1E 00 15 00 00 00 1A 00 00 02 00 00 00 00 00 00 08 21 00 00 08
09 FFF00 0A 03 00 00 00 00 00 ← OA= dec (10) = UDTS message is
incorrect format missing parameters

```

```

Tue Mar 23 16:24:51:566 2004 EST | TCAP (PID 22992) <Error>
TIOS_ERR_SCCP_SYNTAX_ERR: Syntax error in SCCP switch 1 suId = 0

```

## MDL トレース ツール

Cisco PGW 2200 は、TCAP トランザクションを開始するためにトリガーを使用します。TCAP プロトコル トランザクションは、TCAP コントロール レイヤとの間のメッセージの送受信に IN\_TRIGGER 方式を使用します。コールの分析で結果タイプ 22 がヒットすると、IN\_TRIGGER TCAP プロトコルが初期化されます。TCAP の情報とメッセージは、TCAP プロトコル レイヤ (たとえば、MDL 言語で記述されたトリガー) と Cisco PGW 2200 エンジン プロセスの間でタグ、長さ、値または TLV 構文を使用して交換されます。その後、エンジンは次の処理のために TCAP チャネル コントローラに情報を転送します。



Cisco PGW 2200 MDL トレースを使用し、TCAP プロトコル レイヤと TCAP コントローラの間 (エンジン経由) で送受信されるデータを表示します。TCAP チャネル コントローラは、受信した MDL メッセージに必要な処理を行い、適切な IOCC (TALI-IOCC、IP-IOCC または SS7-IOCC) に転送します。また、エンジンは TCAP チャネル コントローラから受信した TCAP メッセージ情報 (SCCP および MTP3 経由で) を、IN\_TRIGGER と呼ばれる TCAP プロトコル レイヤに渡すことができる TLV 形式に変換します。プロトコル レベルで TCAP コールをトレースするには、次の手順を実行します。

1. MDL トレースを開始します。mml> `sta-sc-trc:ss7svc1:log="udts",confirm`
2. TCAP サービスをトリガーするコールを発信します (分析の結果タイプ IN\_TRIGGER をヒットします)。
3. MDL トレースを停止します。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. `get_trc` を実行し、キャプチャされた MDL トレースを表示します。  
`get_trc.sh udts_ss7svc2_20040324174103.btr`
5. オプション S を実行し、PGW 2200 の内部プロセス間のメッセージ フローを示すコールの「sim print」を表示します。
6. オプション D を実行し、PGW 2200 コードによってコールの実際のトレースを表示します。  
注: オプション D および S によって `get_trc.sh` に表示されるコンテンツは、データが内部データ タイプと変数名で表示されるため、わかりにくい場合があります。ただし、TCAP トランザクションをデバッグするには何を確認するべきかについての説明は、「TCAP 用 MDL トレースの分析」セクションにあります。

## TCAP 用 MDL トレースの分析

「sim print」 ( `get_trc.sh` のオプション S ) を使用し、Cisco PGW 2200 プロトコル レベルの全体的なコール フローを表示します。sim print は、「[付録 D](#)」に示したようなものになります。これと異なる場合は、取得したコール フローがどこで異なるのかをメモしておき、そのイベントでトラブルシューティングを開始します。TCAP のトラブルシューティングについては、次のイベントのいずれかに注目してください。

- LTrigger
- LTriggerInformation
- LTriggerNext
- LtriggerRelease

これらは IN\_TRIGGER ステート マシンを駆動する内部イベントです。

Cisco PGW 2200 MDL トレースを使用し、これらの各イベントの実際のコード フローを確認します。OUTPUT IN\_TRIGGER 内の LTrigger の結果、および他の 3 つのイベントはエンジンからの INPUT IN\_TRIGGER メッセージによって IN\_TRIGGER で送受信されます。

## 発信 TCAP メッセージ

TCAP 用 MDL で送受信されるメッセージを特定するには、MDL トレースで IN\_TRIGGER を探します。「[MDL トレースからの IN\\_TRIGGER 構文のサンプル](#)」の図は、MDL とエンジン間で送受信されるメッセージを示します。OUTPUT は、IN\_TRIGGER が TCAP メッセージの転送要求をエンジンに送信したことを示しています。

## トラブルシューティングのヒント



- MDL トレースを使用し、IN\_TRIGGER または OUTPUT が送信されなかった場合に、TRIGGER メッセージがエンジンに送信されたことを確認します。
- IN\_TRIGGER の結果設定のダイヤルプランをチェックします。
- inservice または trigger.dat の設定をチェックします。
- メッセージが SS7 チャネル コントローラから送信されたことを確認します。メッセージが SS7 チャネル コントローラから送信されなかった場合、これは SCCP チャネル コントローラにコールをルーティングする、または有効なメッセージを作成するために十分な情報がなかったことが原因です。
- SCCP の設定と SS7\_SUBSYSTEM の設定をチェックします。
- SSN の状態をチェックします。
- PC の状態をチェックします。

IN\_TRIGGER の出力に成功すると、Cisco PGW 2200 MDL トレースは IN\_TRIGGER への INPUT としてそのメッセージへの応答を表示します。

### MDL トレースからの IN\_TRIGGER 構文のサンプル

```
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 34 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
```

INPUT メッセージは、TCAP プロトコルから送信された要求 (または OUTPUT メッセージ) に関するエンジンからの応答です。エンジンはエンジンそのものの代わりに、または TCAP のレイヤの代わりに応答できます。

IN\_TRIGGER メッセージは、SCP に LINE で送信される UDT メッセージを作成するために使用する、TCAP および SCCP 情報をエンジンおよびチャネル コントローラにエンジンに送信するよう MDL に指示します。エンジンに送信された情報は trigger.dat ファイルから取得され、このメッセージの出力の真上に表示されます。MDL で作成されたとおりにメッセージの内容を表示するには、テキスト IN\_TRIGGER から上にスクロールします。次のように、メッセージを作成するプロシージャの開始部は、SendMessage()... で示されています。



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

```

## トラブルシューティングのヒント

- TCAP クエリーが誤ったデータで Cisco PGW 2200 から送信された場合は、MDL トレースを使用して Cisco PGW 2200 がその情報を正確にどこから取得したかを確認できます。ほとんどの情報は trigger.dat ファイルから取得されます。Cisco PGW 2200 が発信メッセージの情報をどこから取得したかを確認するには、問題の TCAP 要素を検索します ( IN\_TRIGGER から )。たとえば、TCAP タイプが誤ってエンコードされている場合は、MDL トレースで文字列 tcapType を探します ( これは、書き込みフィールド tcapType の周辺にあります )。
- TCAP のコンテンツをエンコードするために Cisco PGW 2200 が trigger.dat をどこから読み込むかを確認するには、次の表に示す文字列を検索します。これらの文字列は trigger.dat 情報を取得するために使用されるプロシージャコールを表します。これらのプロシージャコールは、INPUTLTrigger イベントと問題の OUTPUTIN\_TRIGGER メッセージの間で発生します。

名前	説明	MDL 検索文字列
TT	トリガー テーブル レコード	GetTT

MA	メッセージアクションレコード	GetMA
MS	メッセージ送信レコード	GetMS
OS	送信動作	GetOS
PS	パラメータ送信レコード	GetPS
RR	受信した応答レコード	GetRR
MR	メッセージ受信レコード	GetMR
または	受信動作	GetOR
PR	パラメータ受信レコード	GetPR
RA	応答アクションレコード	GetRA
AD	アクションデータ	GetAD

## 着信 TCAP メッセージ

INPUT メッセージは、要求に関するエンジンからの応答です。エンジンはエンジンそのものの代わり、または TCAP のレイヤの代わりに応答できます。着信メッセージは次の出力例に示すように、Cisco PGW 2200 MDL トレースの INPUT\_IN\_TRIGGER メッセージ文字列によって識別されます。この例では、デコードされるメッセージも示します。これは、TCAP 応答の潜在的な問題を特定する必要がある場合に便利です。

Cisco PGW 2200 MDL が受信したエンジンのメッセージをデコードするには、このドキュメントで前述したのと同じ TLV 形式を使用します。これらのメッセージはテキスト、INPUT\_IN\_TRIGGER の直後にデコードされます。

```
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
```

これは、UDTS メッセージに対する着信応答の出力例です。

```
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
```

Cisco PGW 2200 MDL トレース ( TCAP コールに対する ) から取得できる別の重要な情報は、LTriggerRelease 原因値です。LTriggerRelease にエンコードされた INErrorElem も、コールまたは TCAP トランザクションが想定どおりに機能しない原因に関する情報を提供します。IN\_TRIGGER が受信した最初の LTrigger イベントに応じて送信された LTriggerRelease を示す次の Cisco PGW 2200 MDL 図を参照してください。IN\_TRIGGER イベントと INErrorElem 値の詳細については、「[付録 E](#)」を参照してください。



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

## 付録 A : MDL タグ

Cisco PGW 2200 MDL タグは、Cisco PGW 2200 MDL とエンジンの間で交換されます。この付録では、TCAP トランザクションで使用するすべてのタグの順序、内容、および形式について説明します。これらのタグ値を入力するために使用される情報は、コール コンテキストと trigger.dat ファイルの入力値から取得されます。トリガー ファイルは、TCAP メッセージを作成するためにエンジンで送受信する必要がある情報、および応答の受信時に TCAP メッセージを処理するためにエンジンから受信する必要がある情報を指定するために使用されます。

次のタグは、TCAP のコール処理に使用されます。

- **TAG ID 1 : TCAP Type**説明 TCAP MDL のタイプを示しますデータ長 : 固定 ( 1 ) データ形式 : 1 = ETSI 300 374-1  
2 = Bell Core GR-1298-CORE  
TR-NWT-001284  
TR-NWT-001285  
3 = Bell Core Pre AIN  
GR-1428-CORE
- **TAG ID 2 : System Destination**説明 イベントの内部宛先データ長 : 固定 ( 1 ) データ形式 : オクテット内容 : 0 = 内部 SCP、1 = Trillium TCAP
- **TAG ID 3 : SCCP Called Address**説明 Trillium に必要な SCCP データデータ長 : 可変データ形式 : Octet 1 Routing Indicators

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

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

Octet 2 DPC Network

Octet 3 DPC Cluster

Octet 4 DPC Member

Octet 5 Called SSN

Octet 6 GTFormat

0 - No global Title Included

1 - Global Title includes nature of address indicator only (ITU)

- Global title includes translation type,  
numbering plan and encoding scheme.(ANSI)

2 - Global Title Includes translation type only.(ITU/ANSI)

3 - Global title includes translation type,  
numbering plan and encoding scheme.

(ITU). - not used in ANSI.

4 - Global Title includes translation type, numbering plan,  
encoding scheme and nature of address digits.

(ITU). - Not used in ANSI.

Octet 7 Translation Type Value

Octet 8 Numbering Plan

0 - Unknown

1 - ISDN Telephony

2 - Telephony

3 - Data

4 - Telex

5 - Maritime Mobile

6 - Land Mobile

7 - ISDN Mobile

Octet 9 Nature Of Number

1 - Subscriber Number

2 - National Number

3 - International Number

Octet 10 Number Of Digits in octets 11 to 43

Octet 11 to 43

Digits in IA5 format

- **TAG ID 4 : SCCP Calling Address** 説明 Trillium に必要な SCCP データデータ長 : 可変データ形式 : Octet 1 Routing Indicators

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

Bit B DPC is present (Octets 2 to 4 have valid data)

Bit C SSN is present (Octet 5 has valid data)

Octet 2 DPC Network

Octet 3 DPC Cluster

Octet 4 DPC Member

Octet 5 Calling SSN

- **TAG ID 5 : TCAP Component Type** 説明 TCAP コンポーネントのタイプデータ長 : 固定 ( 1 ) データ形式 : Octet

0 = Unknown

1 = Invoke

2 = Return Result Last

3 = Return Error

4 = Reject

5 = Return Result Not Last

6 = Invoke Last  
7 = Invoke Not Last

- **TAG ID 6 : TCAP Operation Code説明** TCAP メッセージのオペレーション コードデータ長 : 可変 ( ANSI の場合は常に 4 ) データ形式 : Octet 1 Flag
  - 0 = None
  - 1 = Local
  - 2 = Global
  - 3 = National
  - 4 = Private

Octet 2 Operation Class

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

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

Octet n Op Code Least byte (ITU)
- **TAG ID 7 : TCAP Invoke ID説明** コンポーネントの IDデータ長 : 固定 ( 1 ) データ形式 : オクテット
- **TAG ID 8 : TCAP Correlation ID説明** このコンポーネントに関連するコンポーネントの IDデータ長 : 固定 ( 1 ) データ形式 : オクテット
- **TAG ID 9 : TCAP Dialogue Component ANSI説明** 最初パラメータ以降の TCAP メッセージの本文データ長 : 可変データ形式 : オクテット
- **TAG ID 10 : TCAP Dialogue End Marker説明** 最初パラメータ以降の TCAP メッセージの本文 ( SEQUENCE ) データ長 : 固定 ( 0 ) データ形式 : なし
- **TAG ID 11 : Error説明** エラー データデータ長 : 固定 ( 1 ) データ形式 : オクテット内容 :
  - 1 = TCAP\_ERROR\_SSN\_OOS
  - 2 = TCAP\_ERROR\_PC\_UNAVAILABLE
  - 3 = TCAP\_ERROR\_SERVICE\_NOT\_RESPONDING
  - 4 = TCAP\_TRIGGER\_TIMEOUT
- **TAG ID 12 : STP-SCP group index説明** STP-SCP グループ インデックス、データは分析から渡されます。データ長 : 固定 ( 1 ) データ形式 : オクテット内容 : STP-SCP グループ インデックスの値。
- **TAG ID 13 : TCAP Transport Protocol説明** トランスポート プロトコルのタイプデータ長 : 固定 ( 1 ) データ形式 : オクテット内容 :
  - 1 = TCAP\_TRANSPORT\_SCCP
  - 2 = TCAP\_TRANSPORT\_TCP\_IP
- **TAG ID 14 : TCAP External Error / Problem説明** エラー コンポーネントと結果コンポーネントで送受信されるエラーや問題の値データ長 : 可変データ形式 : オクテット
- **TAG ID 15 : TCAP Body Type説明** コンポーネントの本文のタイプデータ長 : 固定 ( 1 ) データ形式 : オクテット内容 :
  - 1 = TCAP\_BODY\_SEQUENCE
  - 2 = TCAP\_BODY\_SET
- **TAG ID 16 : TCAP Dialog info説明** Trillium TCAP は、このタグを MDL に送信されるすべてのメッセージに含めます。MDL はこの情報を保存し、コールに関連付けられたダイアログの以降のすべてのメッセージまたは単方向のメッセージで Trillium TCAP に送信する必要があります。データ長 : 可変データ形式 : オクテット
- **TAG ID 17 : TCAP Transaction Id説明** Trillium TCAP は、このタグを MDL に送信されるすべてのメッセージに含めます。MDL は CDB に送信するためにこの情報を保存する必要があります。データ長 : 可変データ形式 : オクテット
- **TAG ID 18 : TCAP Database Id説明** Trillium TCAP は、このタグを MDL に送信されるすべてのメッセージに含めます。MDL は CDB に送信するためにこの情報を保存する必要があります。データ長 : 可変データ形式 : オクテット

## 付録 B : SS7 ポイント コードのログオフ

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.)

	クラス タ	ネットワ ーク	メン バ	ポイン ト コード
ESTI ( 14 ビット )	3 ビット	8 ビット	3 ビット	14 ビット
ANSI ( 24 ビット )	8 ビット	8 ビット	8 ビット	24 ビット
PC 1-1-1 ( パディングなし、14 ビットのみ )	001	000 00001	001	001000 = 8 000000 01 = 01
PC 1-4-1 ( パディングなし、14 ビットのみ )	001	00000100	001	001000 = 8 001000 01 = 21
PC 3-3-3	011	00000011	011	011000 = 18 000110 11 = 1B

## 付録 C : SCCP メッセージ タイプ

メッセージ タイプ	メッセージ タイプ コード
CR 接続要求	0000 0001
CC 接続確認	0000 0010
CREF 接続拒否	0000 0011
RLSD リリース済み	0000 0100
RLC リリース完了	0000 0101
DT1 データ フォーム 1	0000 0110
DT2 データ フォーム 2	0000 0111
AK データ確認応答	0000 1000
UDT ユニットデータ	0000 1001
UDTS ユニットデータ サービス	0000 1010
ED 優先データ	0000 1011
EA 優先データの確認応答	0000 1100
RSR リセット要求	0000 1101
RSC リセット確認	0000 1110
ERR プロトコル データ ユニット	0000 1111

のエラー	
IT 非アクティブ テスト	0001 0000
XUDT 拡張ユニットデータ	0001 0001
XUDTS 拡張ユニットデータ サービス	0001 0010
LUDT 長いユニットデータ	0001 0011
LUDTS 長いユニットデータ サービス	0001 0100

## ユニットデータ ( UDT )

UDT メッセージには、次が含まれています。

- 3つのポインタ。
- この表に示すパラメータ。

パラメータ	Q.713 参照	タイプ ( F V O )	長さ ( オクテット )
メッセージ タイプ	2.1	F	1
プロトコル クラス	3.6	F	1
着信側アドレス	3.4	V	最小 3
発信側アドレス	3.5	V	最小 3
データ	3.16	V	2-X ( 注 1 )

注: SCCP の着信側アドレスおよび発信側アドレスは調査中であるため、このパラメータの最大長さはさらに調査する必要があります。また、SCCP の着信側アドレスおよび発信側アドレスにグローバル タイトルが含まれていない場合、最大 255 オクテットのユーザ データの転送を許可されます。

## ユニットデータ サービス ( UDTS )

UDTS メッセージには、次が含まれています。

- 3つのポインタ。
- この表に示すパラメータ。

パラメータ	Q.713 参照	タイプ ( F V O )	長さ ( オクテット )
メッセージ タイプ	2.1	F	1
リターンの原因	3.12	F	1
着信側アドレス	3.4	V	最小 3
発信側アドレス	3.5	V	最小 3
データ	3.16	V	2-X ( 注 )

注: SCCP の着信側アドレスおよび発信側アドレスは調査中であるため、このパラメータの最大長さはさらに調査する必要があります。また、SCCP の着信側アドレスおよび発信側アドレスに

グローバル タイトルが含まれていない場合、最大 255 オクテットのユーザ データの転送を許可されます。

次の表はユニットデータとユニットデータ サービスのサンプル SCCP メッセージの詳細を示しています。

パラメータ	タイプ (F V O)	長さ (オクテット)	発信メッセージの相関関係	着信メッセージの相関関係
メッセージタイプ	F	1	09	0a
プロトコルクラス	F	1	80	01
着信側アドレスポインタ	F	1	03	03
発信側アドレスポインタ	F	1	07	0d
データポインタ	F	1	0b	11
着信側アドレス	V	最小 3	04 c3 21 08 0c	04 c3 ~ 30 00
発信側アドレス	V	最小 3	04 c3 09 08 67	18 38 33 44 44
データ (TCAP DATA)	V	04 c3 09 08 67 18 38 33 44 44 データ (TCAP DATA) V	52 62 ~ 20 00	29 62 ~ 00 10

注: これらのメッセージは単なる例であり、実際のクエリー応答の組み合わせとシーケンスが反映されているとは限りません。

## UDTS リターンの原因

ユニットデータ サービス、拡張ユニットデータ サービス、または長いユニットデータ サービスのメッセージでは、「リターンの原因」パラメータ フィールドはメッセージが返される理由を含む 1 オクテットのフィールドです。ビット 1 ~ 8 は次のようにコーディングされます。

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

## 付録 D : TCAP メッセージ用 MDL インターフェイス

すべてのメッセージは一般的な TLV 形式に準拠します。

- **コール インスタンスとプロセス ID** : 8 バイト長さでエンジンによって受信され、変更されずにエンジンからの応答メッセージで返されます。
- **メッセージ ID** : TCAP プロトコル レイヤで送受信されるメッセージを示します (この表に表示される値)。
- **タグ付け ID** : タグの番号とタグのデータ (タグ ID、データ長、およびデータ) は TCAP のメッセージでリモートの宛先に送信される内容を指定します。長さが可変でデータ長 (オクテット) によって定義されるタグ項目のデータ フィールドを除く、すべてのフィールドのサイズは固定されています。それぞれのフィールドの合計長、コール インスタンスとプロセス ID、メッセージ ID、タグ ID とデータ長は、上位バイトで最初に送信されます。

## 付録 E : 内部 MDL インターフェイス

本質的に、TCAP ステート マシン オブジェクト (SMO) との通信はデータを含む信号を使用します。すべての MDL データ タイプは、信号と共に送信できます。信号とデータの名前と意味を次に示します。

- **LTrigger説明** これは、ダイアログを開始するために LCM が TCAP に送信する最初の信号です。ELAN では、INTriggerElem には stpScpGroupIndex も含まれています。MSG\_ACTION\_COPY\_STP\_SCP\_INDEX\_FROM\_SIGNAL\_DATA を使用するには、これを MA テーブルで設定する必要があります。コンポーネント : INTriggerElem、BNumberElem、BNumberDataElem
- **LTriggerInformation説明** この信号はダイアログが続行されると、LTrigger に応じて TCAP から LCM に送信されます。コンポーネント : INTriggerElem、BNumberElem、

## BNumberDataElem

- **LTriggerNext説明** この信号は既存のダイアログの後続のトリガー要求として LCM から TCAP に送信されます。コンポーネント： INTriggerElem、BNumberElem、BNumberDataElem
- **LTriggerRelease説明** この信号は、LCM または TCAP のいずれかから送信される最後の信号で、SCP から応答を受信した後、LTrigger に応じて TCAP から送信できます。コンポーネント： INErrorElem、BNumberElem、BNumberDataElem、INErrorElem の値は次のとおりです。

```
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\_INVALIID\_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

## **関連情報**

- [Cisco PGW 2200 ソフトスイッチ テクニカル ノート](#)
- [音声に関する技術サポート](#)
- [音声とユニファイド コミュニケーションに関する製品サポート](#)
- [Cisco IP Telephony のトラブルシューティング](#)
- [テクニカルサポートとドキュメント - Cisco Systems](#)