

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[规则](#)

[背景信息](#)

[TCAP解决方法](#)

[嗅探器以太网线路](#)

[Platform.log TCAP Trace](#)

[MDL Trace工具](#)

[附录 A : MDL标记](#)

[附录 B : 注销SS7点代码](#)

[附录 C : SCCP消息类型](#)

[单元数据\(UDT\)](#)

[单元数据服务\(UDTS\)](#)

[UDTS返回原因](#)

[附录 D : TCAP消息的MDL接口](#)

[附录 E : 内部MDL接口](#)

[相关信息](#)

简介

处理功能应用程序零件(TCAP)交互式应用程序的提供支持在分布式环境。TCAP定义了在其用户之间的一份端到端协议。这在SS7的网络或该另一的网络可能查找支持TCAP (IP)。

先决条件

要求

本文读者应该有以下知识：

- [Cisco媒介网关控制器版本9](#)

使用的组件

本文档中的信息根据Cisco PGW 2200软交换。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

规则

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

背景信息

TCAP协议包括两下层：

- 组分下层
- 处理下层

组件下层接口用转换引擎。转换引擎是服务用户或子系统编号(SSN)的等同。组件下层支持这些服务：

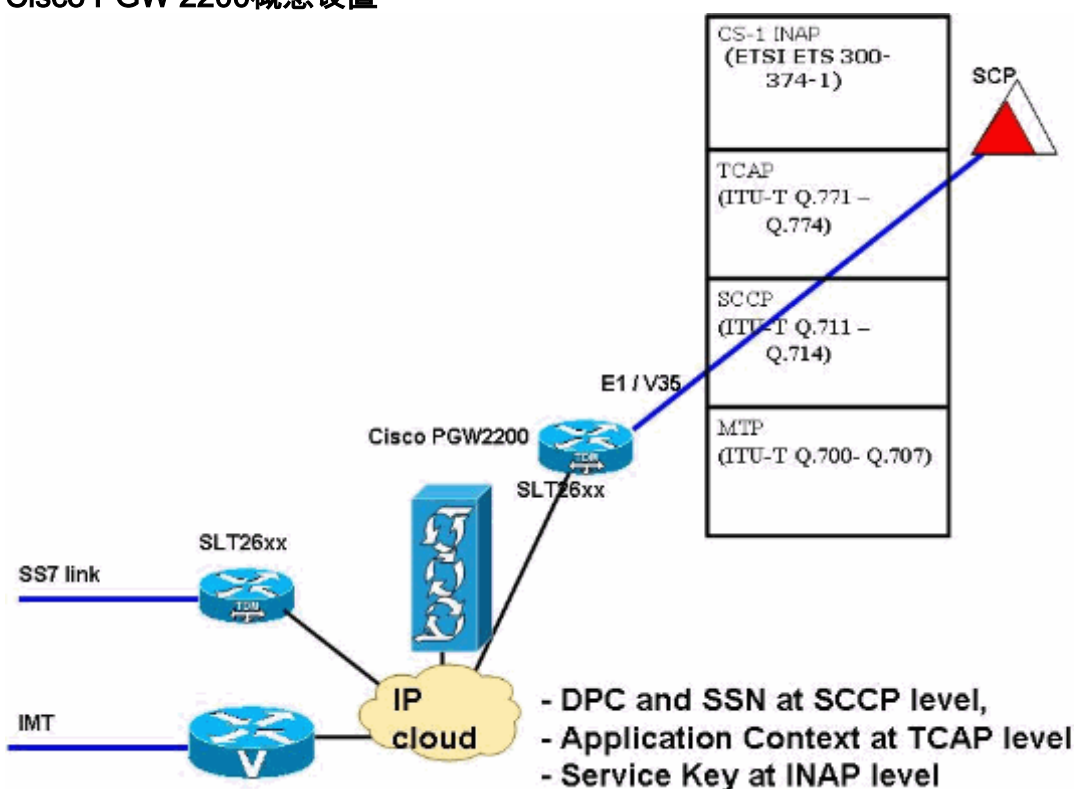
- 操作和回复的关联。
- 异常情况处理。

与信令连接控制零件(SCCP)的处理下层接口。TCAP只支持无连接网络服务。处理下层与SCCP网络通过无连接接口。

TCAP软件使用SCCP软件服务路由消息对目的节点的TCAP用户。TCAP和SCCP软件之间的接口紧密结合。从引擎的每TCAP请求包含全球称号和目的地子系统编号。TCAP提供子系统编号给SCCP为信号转发点(STP)代码查找。如果SS7地址和路由正确地配置和完全能操作，请排除故障SCCP和TCAP信息通过和接收在Cisco PGW 2200和一远程SCCP或者TCAP对等体之间。

Cisco PGW 2200使用SCCP封装传输消息传输部分的TCAP查询。对等体之间的此SCCP通信发送，不用在MTP的一连接。Cisco PGW 2200使用SCCP Unidata (UDT)发送数据到无连接通信的远程SCCP节点。当SCCP UDT消息顺利地，传送PGW2200收到有效答复。以UDT消息的形式，这典型地是。这些UDT消息交换实现PGW2200和远程SCCP对等体之间的无连接通信(例如TCAP数据库查找的服务控制点[SCP])。PGW2200定义了陈述SCCP对等体UDT的一个可选字段如果“请返回在传送对远程节点的错误”所有信息内容，如果UDT消息是无法投递的。Unidata服务(UDTS)消息用于实现此错误反应。UDTS消息表明对PGW2200 UDT消息接收在远程节点(例如STP或SCP)不可能传送到目的地。

Cisco PGW 2200概念设置



TCAP解决方法

在[Background Information部分](#)(UDT/UDTS)讨论的SCCP消息传送是关键，当您排除故障TCAP服务和功能时。在您排除故障发送或接收前的TCAP数据请解决所有问题在SCCP层。UDT和UDTS消息的格式在[附录C](#)显示。

请使用这些Cisco PGW 2200工具调试要求TCAP的呼叫(TCAP/SCCP)服务：

- [嗅探器以太网线路](#)用工具例如Ethereal、UNIX监听和刺探者。
- 在PGW2200的[Platform.log TCAP trace](#)。
- [MDL](#)呼叫处理的[Trace工具](#)在协议级。

嗅探器以太网线路

传送MTP3和上层的SS7信息的Cisco PGW 2200用途可靠UDP (RUDP)在本地MTP1和MTP2设备之间(例如信令链接终端[SLT])。此通信在端口7000典型地完成在Cisco PGW 2200本地以太网接口。这可配置。参考[配置指南](#)关于在配置PGW “stPort”端口的详细信息XECfgParm.dat的。

您能使用所有以太网嗅探器查看在Cisco PGW 2200和其本地MTP2控制设备之间的发送的数据包。然而，不是所有支持用于的MTP和SCCP协议显示一个解码的消息。如果以太网嗅探器不供给客户，请使用snoop命令的UNIX排除故障。输出snoop命令不是用户友好的，然而是有用在最坏的情况。

支持SS7协议栈的以太网嗅探器更喜欢。它允许您解码在Cisco PGW 2200以太网接口看到的数据包。可能也使用 [开放源嗅探器](#)例如[Ethereal](#)并且是线上可以得到的。

如果商业嗅探器工具不是可用的，请发出snoop命令在目标Cisco PGW 2200发现发送的消息的六角形的数据输出到/从Cisco PGW 2200。使用在Cisco PGW 2200的根权限，请发出此命令发现六角形的数据发送在已配置的‘stPort外面’。关于snoop命令的更多信息，参考‘监听联机资料的或SUN管理指南。

```
#snoop -d <ethernet device name> -x 42 port <stPort>  
发出此命令监听数据包派出了以太网设备， hmeX， 在端口7000。
```

```
#snoop -d hmeX -x 42 port 7000  
这是获取SS7数据包示例输出用snoop命令。
```

```
#snoop -d hme0 -x 42 port 7000  
PGW2200 -> C2600.cisco.com UDP D=7000 S=7000 LEN=96  
0: 4004 d4b5 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 6e22 e120 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 d4b5 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.....
```

思科的刺探者可能也使用(若有)显示SCCP消息的HEX转储。SCCP信息标题解码，但是输出的显示

依靠选择的刺探者版本。重点在哪里是消息类型可视并且给予征兆至于启动排除故障呼叫流。HEX转储显示消息类型09是UDT消息，并且消息类型0a是指示一个错误的UDTS服务消息。因为SS7 PCs显示，消息流的方向也是有用的。如果HEX转储的其余显示(取决于刺探者版本)可以用于进一步解码消息的SCCP和TCAP部分。这根据SCCP和TCAP的业界标准。

这是UDT SCCP消息的刺探者输出与TCAP数据的(对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
    
```

如果有从Cisco PGW 2200的一无法投递的SCCP UDT发送的消息并且/或者SCCP (在远程节点)有问题消息，Cisco PGW 2200收到UDTS响应消息。此消息指示是非常有用的在故障排除的一个‘回归原因’。UDTS是消息类型10 (或0a十六进制)。

这是一个UDTS SCCP消息的示例与TCAP数据的(从PSTN)。

注意： 此消息是仅示例并且可能不反射一个实际查询答复组合/顺序。显示的格式和信息量根据刺探者版本变化。

```

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 *.v.....
    
```

此刺探者输出显示IAM、UDT、UDTS和REL顺序。

注意： 此消息是仅示例并且可能不反射一个实际查询答复组合/顺序。显示的格式和信息量根据刺探者版本变化。

10-01-19-0001 1-001-1[02057] ITU SCCP.->IAM (01) CQ=0000 CQPA=0103444 CQPA=00110001
02-08 Pr:0 Ni:NTL
10-01-19-0001 1-001-1[02057] ITU SCCP.->UDT (09) CGPA=0103TCAPMsgType=
Pr:0 Ni:NTL
10-01-19-0001 1-001-1[02057] ITU SCCP.->UDTS (0a) CGPA=0012TCAPMsgType=
Pr:0 Ni:NTL
10-01-19-0001 1-001-1[02057] ITU SCCP.->REL (0a) CQ=0000 CQPA=0103444 CQPA=00110001
02-08 Pr:0 Ni:NTL

这是包括SS7 SCCP和TCAP信息的SS7嗅探器跟踪。

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

排除故障提示：UDTS返回原因

对于UDTS消息，‘回归原因’是第一个字节在消息类型_{0a}。此值帮助确定STP/SCP为什么发送UDTS错误反应。如果此信息不是可视在嗅探器，请继续对[Platform.log TCAP Trace](#)部分为了启用在Cisco PGW 2200日志的TCAP跟踪。

[Platform.log TCAP Trace](#)

MML允许用户开始TCAP trace该转存TCAP频路控制点的<Trace>消息到/opt/CiscoMGC/var/log/platform.log。TCAP trace允许用户发现对SS7频路控制点的TCAP/SCCP发送的消息路由对SS7交换MTP3。请参阅[附录E](#)关于TCAP查询的消息流通过PGW2200软件。

TCAP跟踪通过mml开始用sta-tcap-trc命令。为了捕获相关信息，enable (event) debug日志TCAP和SS7频路控制点的。

这是示例如何启用TCAP trace：

```
mml> set-log:TCAP-01:debug,confirm MGC-01 - Media Gateway Controller 2004-03-26 11:17:31.503
ESTM COMPLD "TCAP-01" ;mml> set-log:ss7-i-1:debug,confirm MGC-01 - Media Gateway
Controller 2004-03-26 11:17:40.715 ESTM COMPLD "ss7-i-1" ;mml> sta-tcap-trc MGC-01 -
Media Gateway Controller 2004-03-26 11:05:27.040 ESTM RTRV SROF "TCAP-01" /* Component
already started */ ;
```

注意： Debug日志在系统性能有效果，并且不应该用于在高呼叫容量下的一个生产环境。请相应地请计划您的维护窗口。

由Cisco PGW 2200的TCAP发送的消息

一旦IN_TRIGGER发送到引擎，传送信息的引擎生存在PGW2200外面。从协议级通过的下来信息被传递到TCAP频路控制点。TCAP部分发送下来到SCCP频路控制点。并且，日志在platform.log创建指示TCAP消息‘已发送’。从上一个UDT消息(表示在本文的嗅探器部分)您能看到PGW2200如何记录相关的信息对在platform.log的此同样消息。此平台日志匹配在[示例SCCP消息细分](#)显示的数据内容：在[附录C的单元数据/单元数据服务表](#)。从此表，第一个值是数据长度值(52十六进制= 82十进制)。实际TCAP数据部分跟随消息长度。在嗅探器或刺探者不是可用的情况下，此platform.log可以是使用的查看/调试TCAP和SCCP处理。

排除故障提示： 如果TCAP信息没有传送下来对SCCP，有问题在MDL或引擎级别。排除故障MDL trace并且注视着Ltrigger和LTriggerRelease信号。

此输出显示发送TCAP的PGW2200日志下来堆积对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 11 f 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 f 83 7 3
11 3 23 22 11 11 9a 2 20 0
```

在TCAP传送信息对SCCP后，SS7频路控制点播放SCCPMSG并且记录消息的六角形的表示指示消息的收据。如此输出所显示，此HEX转储包括SCCP和TCAP部分。

排除故障提示：

- 请使用显示的SCCP消息格式在[附录C](#)解码消息类型、SCCP报头信息(显示在[输出中](#)以黄色)和TCAP数据的开始处(显示在用蓝色[输出中](#))。1e0002在[输出中](#)代表从dpc.dat的终点代码，并且SCCP消息输出在类型"1"之后开始(从SCCP消息类型开始)。
- PGW2200记录计数器和报警SCCP、TCAP和SS7事件的。如果评定启用，请检查计数器TCAP消息。并且请检查SCCP、接收和传送的UDT和UDTS。参考MGC运行程序的这些文档。[管理系统测量Cisco MGC评定获取TCAP处理](#)
- 如果SS7频路控制点不接收发送的消息在PGW2200外面，请验证该TCAP传送消息下来对SCCP。如果TCAP层传送消息下来，可以这是因为SCCP没有建立足够的信息适当的SCCP消息。这可能也是暗示SS7子系统没有正确提供也不是可用的。检查此列表验证：SS7点代码配置和状态SS7子系统配置SS7子系统路由配置本地和远程SSN状态IN服务配置(trigger.dat)**系统确认**

```
mml>rtrv-spc:all MGC-01 - Media Gateway Controller 2004-03-26 13:22:05.492 ESTM RTRV
"ss7svcl:DPC=001.022.001,DNW=2:OPC=001.001.001:IS"
"ss7svcl2: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
ESTM RTRV "session=fix55ltgp:ss7subsys" /* NAME = itussn1DESC = pc_ssn rte-ssn 48SVC =
scplPRI = 1MATEDAPC = LOCALSSN = 101PROTO = SS7-ITUSTPSCPIND = 1TRANSPROTO = SCCPOPC =
opclSUAKEY = REMOTESSN = 48 */ ;mml> rtrv-lsnn:all MGC-01 - Media
Gateway Controller 2004-03-26 11:49:01.985 ESTM RTRV "TCAP-01:SSN=12,PST=IS" "TCAP-
01:SSN=101,PST=IS" "TCAP-01:SSN=102,PST=IS" ;mml> rtrv-rsnn:all MGC-01 - Media Gateway
Controller 2004-03-26 11:49:04.695 ESTM 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 ESTM RTRV
"session=fix55ltgp:inservice" /* NAME = finap-initdpSKORTCV = 90001GTORSSN =
ROUTEBYSSNGTFORMAT = NOGTMSNAME = finap-initdp */;mml> prov-rtrv:SS7ROUTE:NAME="route4"
MGC-01 - Media Gateway Controller 2004-03-30 11:53:08.493 ESTM RTRV
"session=fix55ltgp:SS7ROUTE" /* NAME = route4DESC = rte to 1.4.1 scplOPC = opclDPC =
scplLNKSET = ls3PRI = 1 */ ;
```
- 如果所有此信息看来正确(如显示的输出所显示以上)请验证从TCAP协议级发送的下来标记为的值例如SSN、SCCPCalledParty地址和SCCPCallingParty地址。

输入Cisco PGW 2200的TCAP消息

反向逻辑可以用于跟踪进入Cisco PGW 2200被注定给SS7堆叠的TCAP/SCCP用户层的SS7消息。PGW2200日志表示进入SS7频路控制点的SS7消息(从SS7线路)和发送对处理的TCAP。消息被划分在SS7堆叠的每块层。并且，请注释OPC/DPC、服务指示器(SIO)和信令链路选择(SLS)。OPC和DPC在ITU格式代表(在本例中仅)。

排除故障提示：验证从SS7线路接收的消息类型。如果UDTS消息是接收检查‘回归请导致’。

当收到从SS7线路时的SCCP消息此输出显示PGW2200日志：

```
Thu Dec 4 15:23:04.953 2003 EST | ss7-i-1 (PID 9518) <Debug>
CP Received PDU from ssetl d 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 opcl: 809 (bytes 134) sio: 83 sio 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 mmm30 03 07 0b 04 mmmc3 09 08 67 04 mmmc3 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 43 00 0 0 49 11f66 2a 28 28 6 7 0 11 86 5 1 1 1 a0 1d61 1b a1 d 6 b
2a 81 76 82 15 1 1 1 0 1 a2 3 2 1 0 a3 5 a1 3 2 1 1 6c 3d a1 17 2 1 4 2 1 17 30 f a d
80 b 80 1 a 81 1 0 a2 3 80 1 1 a1 22 2 1 5 2 1 23 30 1 a 80 10 30 e a0 c a0 a a1 5 a0 3 81
1 6 82 1 a 81 1 1 a2 3 80 1 1
```

排除故障提示：请使用显示的SCCP消息格式在[附录C](#)解码消息类型、SCCP报头信息(显示在输出中以黄色)和TCAP数据的开始。在以上输出的1e0002在dpc.dat表示主叫地址(OPC)消息的接收在PGW如代表。SCCP消息输出在"0"之后开始(从SCCP消息类型开始)。

此输出是从PGW2200日志，当接收UDTS TCAP经过SCCP/MTP时：

```
Thu Mar 25 18:35:35:385 2004 EST | ss7-i-1 (PID 27288) <Debug>CP Received PDU from ssetId 3,
chan 0Thu 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 ffffffff1 20 02 01 01 02 01 00 30 18 ffffffff80 04 00 00
00 01 ffffffff82 07 01 10 18 38 33 44 44 ffffffff83 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/sscp_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 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.
```

此输出是从PGW2200日志，当收到一个无效TCAP消息经过SCCP/MTP时：

```
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 suld = 0
```

MDL跟踪工具

Cisco PGW 2200使用触发启动TCAP处理。TCAP协议处理使用IN_TRIGGER方法到/从TCAP控制层发送和收到信息。当呼叫分析命中数结果类型22，IN_TRIGGER TCAP协议初始化。TCAP信息/消息被交

换在TCAP协议层(例如，用MDL语言写入的触发)使用标记、长度和值或者TLV语法，和Cisco PGW 2200引擎进程之间。引擎然后寄信息给进一步处理的TCAP频路控制点。

请使用Cisco PGW 2200 MDL trace发现到/从TCAP协议层发送到TCAP控制器的数据(通过引擎)。TCAP频路控制点执行必要处理在接收的MDL消息并且寄他们给适当的IOCC (TALI-IOCC、IP-IOCC或者SS7-IOCC)。引擎也转换从TCAP频路控制点接收的TCAP留言信息(通过SCCP/MTP3)到可以通过到TCAP协议层的TLV格式，亦称IN_TRIGGER。要跟踪TCAP呼叫在协议级，请完成这些步骤：

1. 开始MDL trace。mml> `sta-sc-trc:ss7svc1:log="udts",confirm`
2. 做触发TCAP服务的一呼叫(命中数分析结果类型IN_TRIGGER)。
3. 终止MDL trace。mml> `stp-sc-trc:all MGC-01 - Media Gateway Controller 2004-03-24 17:41:04.702 ESTM COMPLD "ALL:Trace stopped for the following files: ../var/trace/udts_ss7svc2_20040324174103.btr`
4. 运行get_trc查看获取MDL trace。get_trc.sh udts_ss7svc2_20040324174103.btr
5. 运行选项S发现‘sim打印显示在内部PGW2200进程之间的消息流的’呼叫。
6. 运行选项D发现呼叫的实际trace通过PGW2200代码。**注意：**当数据以内部数据数据类型和变量名称，表示在get_trc.sh的选项显示的内容D和S可能不是显然的了解。然而寻找什么的说明调试TCAP处理在TCAP部分的MDL痕量分析显示。

MDL TCAP的痕量分析

请使用‘sim打印’(get_trc.sh的选项S)查看在Cisco PGW 2200协议级的整体呼叫流。sim打印类似于在附录显示的那个D。如果它不，设法记录下来派生的呼叫流分流的和请开始排除故障与该事件。对于TCAP故障排除，请集中您的注意于这些事件之一。

- Ltrigger
- LTriggerInformation
- LTriggerNext
- LtriggerRelease

这些是驱动IN_TRIGGER状态机的内部事件。

请使用Cisco PGW 2200 MDL trace为这些事件中的每一个发现实际代码流。Ltrigger导致输出IN_TRIGGER，并且人三发送的已接收由IN_TRIGGER由从引擎的一个输入IN_TRIGGER消息。

流出的TCAP消息

识别出入TCAP的MDL的消息，IN_TRIGGER搜索在MDL trace。从MDL Trace图形的示例IN_TRIGGER语法显示消息被派出的和一个接收到MDL到/从引擎。表明IN_TRIGGER发送一个要求引擎转发TCAP消息。

故障排除提示

- 请使用MDL trace验证信息传送了到引擎，如果IN_TRIGGER或发送。
- 检查Dialplan IN_TRIGGER结果配置。
- 检查在职并且/或者trigger.dat配置。
- 验证信息传送了在SS7频路控制点外面。如果消息未曾做它在SS7频路控制点外面，它是有SCCP的频路控制点的结果建立足够的信息路由呼叫或一个有效消息。
- 检查SCCP配置和SS7_SUBSYSTEM配置。
- 检查SSN状态。

- 检查PC状态。

如果IN_TRIGGER输出是成功的，Cisco PGW 2200 MDL trace显示对该消息的答复作为到IN_TRIGGER。

示例从MDL Trace的IN_TRIGGER语法

```
OUTPUT IN_TRIGGER: 00 00 00 0a 00 00 00 69 00 01 0b 00 01 00 01 01 00 02 00 01 01 00 03 00 07 01 00 00 00 00 00 00 00 0c 00 01 03 00 0f
00 01 01 00 13 00 0d 02 00 2a b1 76 b2 15 01 01 01 01 00 01 00 05 00 01 01 00 06 00 03 01 02 00 00 07 00 01 01 00 09 00 1d 80 04 00 01 5f 91
82 08 83 10 65 27 32 54 76 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 b1 76 b2 15 01 01 01 01 00 01 00 05 00 01 01 00 06 00 03 01 00 17 00 07 00 01 04 00 09 00
0fa0 0d 30 0b 80 01 0a 81 01 00 a2 03 80 01 01 00 05 00 01 01 00 06 00 03 01 00 23 00 07 00 01 05 00 09 00 1a 80 10 30 0e a0 0c a0 0a a1 05 a0
03 81 01 06 b2 01 0a 81 01 01 a2 03 80 01 01 00 0a 00 00
```

是从引擎的答复关于从TCAP协议(或信息)传送的请求。引擎能代表TCAP层独自地响应代表或。

IN_TRIGGER消息表明MDL发送TCAP/SCCP信息下来到将使用的引擎和频路控制点修建在对SCP的LINE被派出的UDT消息。信息发送下来对引擎从trigger.dat文件派生，并且直接地上面此消息输出显示。要看到此消息内容，因为MDL建立了它，请从文本IN_TRIGGER移动消息建立步骤的开始是由SendMessage()表示的显示此处。

```
FUNCTION SendMessage() BEGIN

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

    <messageData>.processId = bit(self), 32) -> '000000000000000000000000000000001101001'B

    <messageData>.callRef := bit(CC.db.essentialData.releaseData.DATA.globalCallRefElem.DATA, 32)
-> '000000000000000000000000

0000000000000000101'B

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

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

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

    OUTPUT Begin TO LINE AS <messageData> -> ELEMENT

    SET TcapTimer := <defaultTimer> -> 5000

...<omitted lines>

    NEXTSTATE <state> -> STATE_WaitResponse

END INPUT

END STATE

ok
```

[故障排除提示](#) 如果TCAP查询被发送在有不正确的数据的Cisco PGW 2200外面，MDL trace可以用于正确地发现Cisco PGW 2200哪里派生了其信息。大多数信息来自trigger.dat文件。要看到Cisco PGW

2200哪里派生了其出局信息的信息，请搜索(从IN_TRIGGER)有问题的TCAP的元素。例如，如果TCAP类型不正确地编码，字符串tcapTypein的搜索MDL trace (在writingfield tcapType)

- 要看到Cisco PGW 2200哪里读trigger.dat编码TCAP内容，请搜索在此表里显示的字符串。这些字符串代表用于的过程调用获取trigger.dat信息。这些过程调用应该发生在INPUTLTRIGGER的和有问题的OUTPUTIN_TRIGGER的。

名称	说明	MDL搜索字符串
TT	触发表记录	GetTT
MA	消息操作记录	GetMA
MS	发送记录的消息	GetMS
OS	操作发送	GetOS
PS	发送记录的参数	GetPS
RR	收到的响应记录	GetRR
MR	接收记录的消息	GetMR
或者	操作接收	GetOR
PRS	接收记录的参数	GetPR
RA	答复操作记录	GetRA
AD	操作数据	GetAD

流入TCAP消息

是从引擎的答复关于请求。引擎能代表TCAP层独自地响应代表或。如此示例输出所显示，传入消息由在Cisco PGW 2200 MDL trace的IN_TRIGGER消息串识别。此示例也表示解码的消息。这是有用，如果需要识别可能存在与TCAP答复的任何问题。

要解码Cisco PGW 2200接收的引擎消息MDL，请使用描述的同个TLV格式前在本文。这些消息在文本之后解码，IN_TRIGGER。

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

这是一个流入的响应的输出示例:对UDTS消息的：

```
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 okokreading 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
okokContinuing State Machine: IN_TRIGGER (105) STATE * INPUT Information AS <messageData>
CC.db.nonEssentialData.TCAPTransactionUnixEndTimeElem.DATA :=
MGetTime(CC.db.nonEssentialData.TCAPTransactionMsecEndTimeElem.DATA) -> 1080257735
```

您能从Cisco PGW 2200 MDL trace得到的另一个重要的信息(TCAP呼叫)是LTriggerRelease原因值。在LTriggerRelease编码的INErrorElem也提供见解到呼叫或TCAP处理为什么不运作正如所料。请参阅显示LTriggerRelease被派出以回应IN_TRIGGER接收的最初的Ltrigger事件的此Cisco PGW 2200 MDL图形。关于IN_TRIGGER事件和INErrorElem值的详情，请参阅[附录E](#)。

附录 A : MDL标记

Cisco PGW 2200 MDL标记交换在Cisco PGW 2200 MDL和引擎之间。此附录描述用于TCAP处理的所有标记定货、内容和格式。用于的信息填充这些标记值从在trigger.dat文件和值得到填充的呼叫上下文。触发文件也用于指示应该从TCAP消息处理的引擎接收什么应该发送到/从TCAP消息建立的引擎，并且什么，当答复接收时。

这些标记使用TCAP呼叫处理：

- **TAG ID 1 ? TCAP类型说明：TCAP MDL种类的征兆数据长度：fixed(1)数据格式：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 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 okokreading 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 okokContinuing State Machine: IN_TRIGGER
(105) STATE * **INPUT Information** AS <messageData>
CC.db.nonEssentialData.TCAPTransactionUnixEndTimeElem.DATA :=
MGetTime(CC.db.nonEssentialData.TCAPTransactionMsecEndTimeElem.DATA) -> 1080257735
- **TAG ID 2 ? 系统目的地说明：事件的内部目的地数据长度：fixed(1)数据格式：八位字节内容
：0 =内部SCP，1 =延龄草TCAP**
- **TAG ID 3 ? SCCP被叫地址说明：延龄草要求的SCCP数据数据长度：变量数据格式：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 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 okokreading 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 okokContinuing State Machine: IN_TRIGGER
(105) STATE * **INPUT Information** AS <messageData>
CC.db.nonEssentialData.TCAPTransactionUnixEndTimeElem.DATA :=
MGetTime(CC.db.nonEssentialData.TCAPTransactionMsecEndTimeElem.DATA) -> 1080257735
- **TAG ID 4 ? SCCP主叫地址说明：延龄草要求的SCCP数据数据长度：变量数据格式：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 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 okokreading 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 okokContinuing State Machine: IN_TRIGGER
(105) STATE * **INPUT Information** AS <messageData>
CC.db.nonEssentialData.TCAPTransactionUnixEndTimeElem.DATA :=
MGetTime(CC.db.nonEssentialData.TCAPTransactionMsecEndTimeElem.DATA) -> 1080257735
- **TAG ID 5 ? TCAP组件类型说明：TCAP组件的类型数据长度：fixed(1)数据格式：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 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 okokreading 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 okokContinuing State Machine: IN_TRIGGER
(105) STATE * INPUT Information AS <messageData>
CC.db.nonEssentialData.TCAPTransactionUnixEndTimeElem.DATA :=
MGetTime(CC.db.nonEssentialData.TCAPTransactionMsecEndTimeElem.DATA) -> 1080257735

```

• TAG ID 6 ? TCAP操作代码说明 : TCAP消息操作代码数据长度 : 变量(总是4 ANSI的)数据格式

```

: 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 okokreading 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 okokContinuing State Machine: IN_TRIGGER
(105) STATE * INPUT Information AS <messageData>
CC.db.nonEssentialData.TCAPTransactionUnixEndTimeElem.DATA :=
MGetTime(CC.db.nonEssentialData.TCAPTransactionMsecEndTimeElem.DATA) -> 1080257735

```

• TAG ID 7 ? TCAP调用ID说明 : 组件的ID数据长度 : fixed(1)数据格式 : 八位字节

• TAG ID 8 ? TCAP相关性ID说明 : 此组件关联组件的ID数据长度 : fixed(1)数据格式 : 八位字节

• TAG ID 9 ? TCAP对话组件ANSI说明 : 一个TCAP消息的正文从向前第一个参数的数据长度 : 变量数据格式 : 八位字节

• TAG ID 10 ? TCAP对话末端标记说明 : 一个TCAP消息的正文从向前第一个参数的(顺序)数据长度 : fixed(0)数据格式 : 无

• TAG ID 11 ? 错误说明 : 错误数据数据长度 : fixed(1)数据格式 : 八位字节内容 : 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 okokreading 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 okokContinuing State Machine: IN_TRIGGER
(105) STATE * INPUT Information AS <messageData>
CC.db.nonEssentialData.TCAPTransactionUnixEndTimeElem.DATA :=
MGetTime(CC.db.nonEssentialData.TCAPTransactionMsecEndTimeElem.DATA) -> 1080257735

```

• TAG ID 12 ? STP-SCP分组指数说明 : STP-SCP分组指数 , 从分析通过的数据。数据长度 : fixed(1)数据格式 : 八位字节内容 : STP-SCP分组指数给定值。

• TAG ID 13 ? TCAP传输协议说明 : 传输协议的类型数据长度 : fixed(1)数据格式 : 八位字节内容 :

```

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 okokreading 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 okokContinuing State Machine: IN_TRIGGER
(105) STATE * INPUT Information AS <messageData>
CC.db.nonEssentialData.TCAPTransactionUnixEndTimeElem.DATA :=
MGetTime(CC.db.nonEssentialData.TCAPTransactionMsecEndTimeElem.DATA) -> 1080257735

```

• TAG ID 14 ? TCAP外部错误/问题说明：在错误&结果组件接收或发送的错误或问题值数据长度：变量数据格式：八位字节

• TAG ID 15 ? TCAP体型说明：组件正文的类型数据长度：fixed(1)数据格式：八位字节内容

```

: 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 okokreading 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 okokContinuing State Machine: IN_TRIGGER
(105) STATE * INPUT Information AS <messageData>
CC.db.nonEssentialData.TCAPTransactionUnixEndTimeElem.DATA :=
MGetTime(CC.db.nonEssentialData.TCAPTransactionMsecEndTimeElem.DATA) -> 1080257735

```

• TAG ID 16 ? TCAP对话信息说明：延龄草TCAP在所有发送的消息包括此TAG对MDL。MDL应该存储此信息和发送它到在所有随后的消息的延龄草TCAP与呼叫或单向的消息的涉及的对话。数据长度：变量数据格式：八位字节

• TAG ID 17 ? TCAP交易ID说明：延龄草TCAP在所有发送的消息包括此TAG对MDL。MDL应该存储发送的此信息对CDB。数据长度：变量数据格式：八位字节

• TAG ID 18 ? TCAP数据库Id说明：延龄草TCAP在所有发送的消息将包括此TAG对MDL。MDL应该存储发送的此信息对CDB。数据长度：变量数据格式：八位字节

附录 B：注销SS7点代码

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.)

	团星	网络	成员	点代码
ESTI (14个位)	3位	8位	3位	14个位
ANSI (24个位)	8位	8位	8位	24个位
PC 1-1-1 (没有填充符, 14位)	001	000 00001	001	001000 = 8 00000001 = 01
PC 1-4-1 (没有填充符, 14位)	001	0000010 0	001	001000 = 8 00100001 = 21
PC 3-3-3	011	0000001 1	011	011000 = 18 00011011 = 1B

附录 C : SCCP消息类型

消息类型	消息类型代码
CR连接请求	0000 0001
CC连接确认	0000 0010
拒绝的CREF连接	0000 0011
发布的RLSD	0000 0100
完整RLC的版本	0000 0101
DT1数据表1	0000 0110
DT2数据表2	0000 0111
AK数据确认	0000 1000
UDT Unitdata	0000 1001
UDTS Unitdata服务	0000 1010
ED加快的数据	0000 1011
EA加快的数据确认	0000 1100
reset请求的RSR	0000 1101
RSC重置确认	0000 1110
ERR协议数据单元错误	0000 1111
IT非活动测验	0001 0000
XUDT延长的单元数据	0001 0001
XUDTS扩展的单元数据服务	0001 0010
LUDT龙牌单元数据	0001 0011
LUOTS长单元数据服务	0001 0100

单元数据(UDT)

UDT消息包含：

- 三分球
- 在此表里指示的参数。

参数	Q.713参考	类型(F v O)	长度(八位位组)
消息类型	2.1	F	1
协议簇	3.6	F	1
被叫地址	3.4	v	3最低
主叫方地址	3.5	v	3最低
数据	3.16	v	2-X (注意1)

注意： 由于在SCCP呼叫和主叫用户名详细资料地址的持续的研究，最大长度此参数需要进一步研究。也注意255个八位位组转移用户数据允许，当SCCP呼叫和主叫用户名详细资料地址不包括全球称号时。

单元数据服务(UDTS)

UDTS消息包含：

- 三分球。
- 在此表里指示的参数。

参数	Q.713参考	类型(F v O)	长度(八位位组)
消息类型	2.1	F	1
回归原因	3.12	F	1
被叫地址	3.4	v	3最低
主叫方地址	3.5	v	3最低
数据	3.16	v	2-X (注意)

注意： 由于在SCCP呼叫和主叫用户名详细资料地址的持续的研究，最大长度此参数需要进一步研究。也注意255个八位位组转移用户数据允许，当SCCP呼叫和主叫用户名详细资料地址不包括全球称号时。

此表显示单元数据/单元数据服务的示例SCCP消息细分：

参数	类型(F v O)	长度(八位位组)	相关性出局信息	相关性传入消息
消息类型	F	1	09	0a
协议簇	F	1	80	01
被叫地址指示器	F	1	03	03
主叫方地址指示器	F	1	07	0d
数据指示器	F	1	0b	11
被叫地址	v	3最低	04个c3 21 08个0c	04个c3 ? 30 00
主叫方地址	v	3最低	04个c3 09 08 67	18 38 33 44 44
数据(TCAP DATA)	v	04个c3 09 08 67 18 38 33 44 44个数据 (TCAP DATA) V	52 62 ? 20 00	29 62 ? 00 10

注意： 这些消息是仅示例并且可能不反射一个实际查询答复组合/顺序。

UDTS返回原因

在单元数据服务、延长的单元数据服务或者长单元数据服务消息，“回归原因”参数字段是包含消息返回的原因的一个一个八位位组字段。比特1至8被编码如显示此处：

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.)

[附录 D : TCAP消息的MDL接口](#)

所有消息遵守一个普通的TLV格式 :


- 在从未改变的引擎的响应消息应该乘引擎接收和返回**呼叫实例和ProcessId** -长8个的字节 , 并且。
- **消息ID** -识别由TCAP协议层的信息(在此[表里](#)显示的值传送或接收)。
- **标记为的ID号码**标记和标记数据(标记ID、数据长度和数据)指明什么在对远程目的地的TCAP消息被派出。所有字段长度修复除了长度可变和定义标记项目的数据域(在八位位组)由数据长度。其中每一个字段总长度、呼叫实例和进程ID、消息ID、标记Id和数据长度由最高有效字节首先传送。

[附录 E : 内部MDL接口](#)

在内部 , 与TCAP状态机对象(SMO)的通信是通过与数据的信号。任何MDL数据类型可以用信号传送。列出得信号和数据的名称和含义此处。

- **Ltrigger说明** : 这是LCM发送对TCAP开始对话的第一个信号。在ELAN中 , `INTriggerElem`也包含 `stpScpGroupIndex`。在能将使用的此的MA表里必须设置 `MSG_ACTION_COPY_STP_SCP_INDEX_FROM_SIGNAL_DATA`。 **组件** : `INTriggerElem` , `BNumberElem` , `BNumberDataElem`
- **LTriggerInformation说明** : 当对话继续时 , 此信号从TCAP被发送到LCM以回应 `Ltrigger`。 **组件** : `INTriggerElem` , `BNumberElem` , `BNumberDataElem`
- **LTriggerNext说明** : 此信号从LCM被发送到TCAP , 在现有对话的一随后的触发请求。 **组件** : `INTriggerElem` , `BNumberElem` , `BNumberDataElem`
- **LTriggerRelease说明** : 在答复从SCP后 , 接收此信号是从LCM或TCAP将发送的为时 , 并且可以从TCAP发送以回应 `Ltrigger`。 **组件** : `INErrorElem` , `BNumberElem` , `BNumberDataElem`
`INErrorElem`有这些值 : 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.)

[相关信息](#)

- [Cisco PGW 2200 Softswitch技术说明](#)
- [语音技术支持](#)
- [语音和统一通信产品支持](#)
- [Cisco IP 电话故障排除](#) 
- [技术支持和文档 - Cisco Systems](#)