

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Informações de Apoio](#)

[Resolução de TCAP](#)

[Sniffer a linha de Ethernet](#)

[Traço de Platform.log TCAP](#)

[Ferramenta do traço MDL](#)

[Apêndice A: Etiquetas MDL](#)

[Apêndice B: Point code do fazer logoff SS7](#)

[Apêndice C: Tipos de mensagem de SCCP](#)

[Unitdata \(UDT\)](#)

[Serviço de Unitdata \(UDTS\)](#)

[Causas do retorno UDTS](#)

[Anexo D: Interface MDL para Mensagem TCAP](#)

[Anexo E: Relação interna MDL](#)

[Informações Relacionadas](#)

[Introdução](#)

A peça dos aplicativos das potencialidades de transação (TCAP) fornece o apoio para aplicativos interativos em um ambiente distribuído. O TCAP define um protocolo fim-a-fim entre seus usuários. Isto pode ser ficado situado em uma rede SS7 ou em uma outra rede que apoie TCAP (IP).

[Pré-requisitos](#)

[Requisitos](#)

Os leitores deste documento devem ter conhecimento de:

- [Liberação 9 do controlador do gateway do Cisco media](#)

[Componentes Utilizados](#)

A informação neste documento é baseada em Softswitch Cisco PGW 2200.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto

potencial de qualquer comando.

Convenções

Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

Informações de Apoio

O protocolo de TCAP consiste em duas secundário-camadas:

- Secundário-camada componente
- Secundário-camada da transação

As relações da secundário-camada do componente com o Engine de conversão. O Engine de conversão é o equivalente de um usuário de serviço ou de um número do subsistema (SSN). A secundário-camada componente apoia estes serviços:

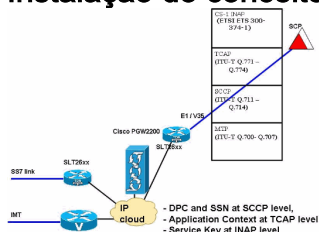
- Associação de operações e respostas.
- Manipulação anormal da situação.

As relações da secundário-camada da transação com sinalização da peça de controle da conexão (SCCP). O TCAP apoia somente um serviço de rede sem conexão. A secundário-camada da transação comunica-se com o SCCP através da relação sem conexão.

O software TCAP usa os serviços do software de SCCP para distribuir as mensagens ao usuário TCAP no nó de destino. A relação entre o TCAP e o software de SCCP é acoplada firmemente. Cada pedido TCAP do motor contém um número do subsistema do título global e do destino. O TCAP fornece o número do subsistema ao SCCP para a consulta do código dos pontos de transferência do sinal (STP). Se o SS7 endereça e as rotas estão configuradas corretamente e plenamente operacional, pesquisa defeitos o SCCP e a informação de TCAP passados e recebidos entre Cisco PGW2200 e um par remoto SCCP ou TCAP.

Cisco PGW2200 usa o SCCP para encapsular para fora perguntas TCAP para o message transfer part do transporte (MTP). Esta comunicação SCCP entre pares é enviada sem uma conexão sobre o MTP. Cisco PGW2200 usa o SCCP Unidata (UDT) para enviar dados ao nó SCCP remoto para comunicação sem conexão. O PGW2200 recebe uma resposta válida quando o mensagem de SCCP UDT é entregue com sucesso. Isto é tipicamente sob a forma de um mensagem de UDT. A troca destes mensagens de UDT facilita a comunicação sem conexão entre o PGW2200 e o par remoto SCCP (tal como o [SCP] do ponto de controle de serviço para consultas de banco de dados TCAP). O PGW2200 define um campo opcional no UDT que indica o par SCCP se “retorne no erro” os índices de toda a mensagem que enviar ao nó remoto se o mensagem de UDT é undeliverable. A mensagem do serviço de Unidata (UDTS) é usada para facilitar esta resposta de erro. O mensagem de UDTS indica ao PGW2200 que um mensagem de UDT recebido no nó remoto (tal como o STP ou o SCP) não pode ser entregue ao destino.

Instalação do conceito de Cisco PGW2200



memória de HEX do mensagem de SCCP. O encabeçamento de mensagem de SCCP é decodificado mas o indicador da saída é dependente da versão de Snooper escolhida. O ponto importante é que o tipo de mensagem é visível e dá uma indicação a respeito de onde começar pesquisar defeitos o fluxo de chamadas. A cópia parcial da memória de HEX mostra que o tipo de mensagem 09 é um mensagem de UDT e o tipo de mensagem 0a é o mensagem de serviço UDTS que indica um erro. O sentido do fluxo de mensagem é igualmente útil desde que o SS7 PC é mostrado. Se o resto da cópia parcial da memória de HEX é mostrado que (depende da versão de espião) pode ser usada para decodificar mais o SCCP e as partes TCAP de mensagem. Isto é baseado nos padrões para indústria para o SCCP e o TCAP.

Esta é a saída de Snooper do mensagem de UDT SCCP com dados TCAP (ao 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 0d06 0b 2a 81 ....^.....*.
76 82 15 01 01 01 01 00 01 6c 27 a1 25 02 01 01 v.....f.%...
02 01 00 30 1d 80 04 00 01 5f91 82 08 83 10 65 ...0....._....e
27 32 54 76 0f83 07 03 11 03 23 22 11 11 9a 02 '2Tv.....#*....
20 00
```

Se há um mensagem de SCCP UDT undeliverable enviado de Cisco PGW2200 e/ou um SCCP (no nó remoto) tem problemas com a mensagem, Cisco PGW2200 recebe um mensagem de resposta UDTS. Esta mensagem indica “uma causa do retorno” que seja muito útil no Troubleshooting. O UDTS é o tipo de mensagem 10 (ou 0a encantar).

Este é um exemplo de um mensagem UDTS SCCP com dados TCAP (do PSTN).

Nota: Esta mensagem é um exemplo somente e não pode refletir uma combinação de resposta de consulta real/sequência. O formato e a quantidade de informação indicados variam segundo a versão de espião.

```
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.....
```

Esta saída de Snooper indica mim está, sequência UDT, UDTS, e REL.

Nota: Esta mensagem é um exemplo somente e não pode refletir uma combinação de resposta de consulta real/sequência. O formato e a quantidade de informação indicados variam segundo a versão de espião.

```
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
```

Este é um farejador de rastreamento SS7 que inclua SS7 SCCP e informação de TCAP.

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

Pesquise defeitos a ponta: Causa do retorno UDTS

Para um mensagem de UDTS, “a causa do retorno” é o primeiro byte após o tipo de mensagem 0a. Este valor ajuda a determinar porque o STP/SCP envia uma resposta de erro UDTS. Se esta informação não é visível no sniffer, continue à seção do [traço de Platform.log TCAP](#) a fim permitir traços TCAP no log de Cisco PGW2200.

[Traço de Platform.log TCAP](#)

MML permite que um usuário comece um traço TCAP mensagens desse <Trace> das descargas para o controlador de canal TCAP em /opt/CiscoMGC/var/log/platform.log. Um traço TCAP permite que o usuário considere o TCAP/mensagens de SCCP enviados ao controlador de canal SS7 para distribuir para fora ao interruptor SS7 sobre o MTP3. Veja o [apêndice E](#) para o fluxo de mensagem de uma pergunta TCAP através do software PGW2200.

O traçado TCAP é começado através do mml com o comando **sta-tcap-trc**. A fim capturar a informação relevante, permita o debug logging para o controlador de canal TCAP e SS7.

Este é um exemplo de como permitir um traço TCAP:

```
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 */ ;
```

Nota: O debug logging pode ter um efeito no desempenho de sistema e não deve ser usado em um ambiente de produção sob o volume de chamada alto. Planeie por favor sua janela de manutenção em conformidade.

Mensagens TCAP enviados por Cisco PGW2200

Uma vez que um `IN_TRIGGER` é enviado ao motor, os seres do motor para enviar a mensagem fora do PGW2200. A informação passada para baixo do nível de protocolo é retransmitida ao controlador de canal TCAP. A parte TCAP é enviada para baixo ao controlador de canal SCCP. Também, um log é criado em `platform.log` para indicar que um mensagem TCAP “esteve transmitido”. Do mensagem UDT anterior (mostrado na parcela do sniffer deste documento) você pode ver como o PGW2200 registra relativo à informação a esta mesma mensagem em `platform.log`. Este log da plataforma combina o conteúdo de dados mostrado na [divisão do mensagem de SCCP da amostra](#): Tabela de [Unitdata/serviço de Unitdata no C do apêndice](#). Desta tabela, o primeiro valor é o valor do comprimento de dados (52 encantam = o decimal 82). A porção de dados real TCAP segue o tamanho da mensagem. Caso o sniffer ou Snooper não estiverem disponível, este `platform.log` pode ser usado para ver/debuga o TCAP e as transações SCCP.

Pesquise defeitos a ponta: Se o mensagem TCAP não é enviado para baixo ao SCCP, há um problema a MDL ou nível do motor. Pesquise defeitos o traço MDL e olhe o sinal de `Ltrigger` e de `LTriggerRelease`.

Esta saída mostra que o log PGW2200 que envia o TCAP empilha para baixo ao SCCP/MTP.

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

Depois que o TCAP envia a mensagem ao SCCP, o controlador de canal SS7 joga o `MSG RECEBIDO DO SCCP` e registra a representação encantar da mensagem para indicar o recibo da mensagem. Esta cópia parcial da memória de HEX inclui o SCCP e as partes TCAP segundo as indicações desta saída.

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

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

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

Pesquise defeitos pontas:

- Use o formato do mensagem de SCCP mostrado no [C do apêndice](#) para decodificar o tipo de mensagem, a informação de cabeçalho SCCP (mostrada na [saída no amarelo](#)) e o começo dos dados TCAP (mostrados na [saída no azul](#)). O `1e0002` na [saída](#) representa o código de ponto de destino de `dpc.dat` e a descarga do mensagem de SCCP começa imediatamente depois do tipo "1" (começo com tipo de mensagem de SCCP).
- O PGW2200 registra contra e alarmes para os eventos SCCP, TCAP e SS7. Se as medidas

Pesquisa defeitos a ponta: Use o formato do mensagem de SCCP mostrado no [C do apêndice](#) para decodificar o tipo de mensagem, a informação de cabeçalho SCCP (mostrada na [saída no amarelo](#)) e o começo dos dados TCAP. O 1e0002 na saída acima representa o endereço chamador (OPC) para a mensagem recebida no PGW como representado em dpc.dat. A descarga do mensagem de SCCP começa imediatamente depois do "0" (começo com tipo de mensagem de SCCP).

Esta saída é do log PGW2200 quando recebe UDTs TCAP sobre o SCCP/MTP:

```
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 ffffffff c3 09 08 65 0a ffffffff 8b 21 08 30
00 18 38 33 44 44 29 62 27 48 01 02 6c 22 ffffffff a1 20 02 01 01 02 01 00 30 18 ffffffff 80 04 00 00
00 01 ffffffff 82 07 01 10 18 38 33 44 44 ffffffff 83 07 01 11 07 13 11 00 10 Thu Mar 25 18:35:35:385
2004 EST | TCAP (PID 27283) <Debug>Got 91 bytes from fifo /tmp/sccp_input (fd=16) Thu Mar 25
18:35:35:385 2004 EST | ss7-i-1 (PID 27288) <Debug>RECEIVED SCCP STACK MSG !--- Indicates
message is from MTP(SS7 stack).!--- Lines omitted.Thu Mar 25 18:35:35:385 2004 EST | TCAP (PID
27283) <Debug>00 01 00 01 1E 00 15 00 00 00 1A 00 00 02 00 00 00 00 00 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.
```

Esta saída é do log PGW2200 quando recebe um mensagem inválida de TCAP sobre o 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 ← 0A= 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
```


Ferramenta do traço MDL

Cisco PGW2200 usa disparadores para iniciar uma transação de TCAP. As transações de protocolo TCAP usam o método `IN_TRIGGER` para enviar e receber mensagens a e da camada de controle TCAP. Quando a análise do atendimento bate o tipo `22` do resultado, o protocolo de TCAP `IN_TRIGGER` está inicializado. A informação de TCAP/mensagens é trocada entre a camada de protocolo de TCAP (por exemplo, os disparadores escritos no linguagem MDL) e o processo do motor de Cisco PGW2200 usando uma etiqueta, um comprimento, e um valor ou uma sintaxe de TLV. O motor encaminha então a informação ao controlador de canal TCAP para o processamento adicional.

Use o traço MDL de Cisco PGW2200 para ver os dados que são enviados a e da camada de protocolo de TCAP ao controlador TCAP (através do motor). O controlador de canal TCAP faz o processamento necessário nos mensagens MDL recebidos e encaminhar-los ao IOCC apropriado (TALI-IOCC, IP-IOCC ou SS7-IOCC). O motor igualmente converte a informação de mensagem TCAP recebida do controlador de canal TCAP (através do SCCP/MTP3) em um formato TLV que possa ser passado à camada de protocolo de TCAP, igualmente conhecido como o `IN_TRIGGER`. Para seguir um TCAP chame a nível de protocolo, terminam estas etapas:

1. Comece um traço MDL.
`.mm1> sta-sc-trc:ss7svc1:log="ufts",confirm`
2. Faça um atendimento que provoque um serviço TCAP (tipo `IN_TRIGGER` do resultado da análise das batidas).
3. Pare o traço MDL.
`.mm1> 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/ufts_ss7svc2_20040324174103.btr
4. Execute o `get_trc` para ver o traço capturado MDL.
`get_trc.sh`
`ufts_ss7svc2_20040324174103.btr`
5. Execute a opção **S** para ver do “uma cópia sim” do atendimento que mostra o fluxo de mensagem entre os processos PGW2200 internos.
6. Execute a opção **D** para ver o traço real do atendimento com o código PGW2200. **Nota:** O índice mostrado pelas opções **D** e **S** em `get_trc.sh` não pode ser óbvio compreender enquanto os dados são mostrados com tipos e nomes variável de dados internos. Contudo uma descrição do que a procurar para debugar transações de TCAP é mostrada na **análise de rastreamento de MDL para a seção TCAP**.

Análise de rastreamento de MDL para o TCAP

Use do “a cópia sim” (opção S de `get_trc.sh`) para ver o fluxo de chamadas total a nível de protocolo de Cisco PGW2200. A cópia do sim assemelha-se a esse mostrado no [apêndice D](#). Se não faz, para tentar fazer uma anotação de onde o fluxo de chamadas derivado diverja e comece a pesquisar defeitos com esse evento. Para o Troubleshooting TCAP, focalize sua atenção em um destes eventos.

- LTrigger
- LTriggerInformation
- LTriggerNext
- LtriggerRelease

Estes são os eventos internos que conduzem a máquina de estado `IN_TRIGGER`.

Use o traço MDL de Cisco PGW2200 para ver o fluxo real do código para cada um destes eventos. O LTrigger conduz a uma SAÍDA `IN_TRIGGER`, e os outros três são enviados recebidos pelo `IN_TRIGGER` por uma mensagem da ENTRADA `IN_TRIGGER` do motor.

informação. A maioria da informação vem do arquivo trigger.dat. Para ver onde Cisco PGW2200 derivou sua informação para o mensagem enviada, procure acima (do IN_TRIGGER) pelo elemento TCAP na pergunta. Por exemplo, se o tipo TCAP é codificado incorretamente, busca para o tcapTypein da corda o traço MDL (em torno do tcapType do writingfield).

- Para ver onde Cisco PGW2200 lê trigger.dat para codificar o índice TCAP, procure pelas cordas mostradas nesta tabela. Estas cordas representam os atendimentos de procedimento usados para recuperar a informação trigger.dat. Estes atendimentos de procedimento devem ocorrer entre o evento INPUTLTRIGGER e a mensagem OUTPUTIN_TRIGGER na pergunta.

Nome	Descrição	Caractere de texto de pesquisa MDL
TT	Registro da tabela do disparador	GetTT
MA	Registro da ação da mensagem	GetMA
MS	Mensagem que envia o registro	GetMS
OS	Emissão da operação	GetOS
PS	Parâmetro que envia o registro	GetPS
RR	Registro da resposta recebida	GetRR
MR	Mensagem que recebe o registro	GetMR
OU	Recepção da operação	GetOR
PR	Parâmetro que recebe o registro	GetPR
RA	Registro da ação da resposta	GetRA
AD	Dados da ação	GetAD

Mensagens recebida de TCAP

O mensagem de entrada é a resposta do motor na referência ao pedido. O motor pode responder em seu próprio nome ou em nome da camada TCAP. O mensagem recebida é identificado pela série de mensagem da ENTRADA IN_TRIGGER no traço MDL de Cisco PGW2200 segundo as indicações destas saídas de exemplo. Este exemplo igualmente mostra a mensagem que é descodificada. Isto é útil se você precisa de identificar quaisquer problemas que puderem existir com a resposta TCAP.

Para descodificar o mensagem de Engine recebido pelo MDL de Cisco PGW2200, use o mesmo formato TLV descrito mais cedo neste documento. Este a mensagem é descodificada imediatamente depois do texto, IN_TRIGGER ENTRADO.

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

Este é exemplo de saída de uma resposta recebida a um mensagem de 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 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

```

Uma outra parte de informação que valiosa você pode obter do traço MDL de Cisco PGW2200 (para atendimentos TCAP) é o valor de causa da LTriggerRelease. O INErrorElem codificado na LTriggerRelease igualmente fornece a introspecção em porque um atendimento ou uma transação de TCAP não trabalham como esperado. Veja este gráfico MDL de Cisco PGW2200 que mostra uma LTriggerRelease que seja mandada em resposta ao evento inicial do LTrigger recebido pelo IN_TRIGGER. Veja o [apêndice E](#) para detalhes sobre eventos IN_TRIGGER e valores do INErrorElem.

OD

END FUNCTION

VAR iNErrorElem := NULL

iNErrorElem.DATA.error = 42 → TRIG_ERROR_UNKNOWN

INSERT iNErrorElem INTO <signalData>

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

FI

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

NEXTSTATE <state> -> STATE_WaitResponse

END INPUT

END STATE

Apêndice A: Etiquetas MDL

As etiquetas MDL de Cisco PGW2200 são trocadas entre o MDL de Cisco PGW2200 e o motor. Este apêndice descreve a ordem, o índice, e o formato de todas as etiquetas usadas nas transações de TCAP. A informação usada para povoar estes valores da etiqueta é obtida do contexto do atendimento e os valores povoados no trigger.dat arquivam. O arquivo do disparador é usado igualmente para indicar o que devem ser enviadas para/desde o motor para a construção do mensagem TCAP e o que deve ser recebido do motor para o processamento de mensagem TCAP quando uma resposta é recebida.

Estas etiquetas são usadas para o Processamento de chamadas TCAP:

- **ETIQUETA ID 1? Tipo TCAP****Descrição:** Indicação do tipo de MDL TCAP**Comprimento de dados:** fixed(1)**Formato de dados:**

```
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
```
- **ETIQUETA ID 2? Destino de Sistema****Descrição:** Destino interno do evento**Comprimento de dados:** fixed(1)**Formato de dados:** Octeto**Índice:** 0 = SCP interno, 1 = Trillium TCAP
- **ETIQUETA ID 3? Endereço do chamador SCCP****Descrição:** Dados SCCP exigidos pelo trillium**Comprimento de dados:** Variável**Formato de dados:**

```
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
```
- **ETIQUETA ID 4? Endereço chamador SCCP****Descrição:** Dados SCCP exigidos pelo trillium**Comprimento de dados:** Variável**Formato de dados:**

```
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
```
- **ETIQUETA ID 5? Tipo do componente TCAP****Descrição:** Tipo de componente TCAP**Comprimento de dados:** fixed(1)**Formato de dados:**

```
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
```

- **ETIQUETA ID 6? Código de operação TCAP** Descrição: Código de operação do mensagem TCAP Comprimento de dados: Variável (sempre 4 para o ANSI) Formato de dados: 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
```

- **ETIQUETA ID 7? O TCAP invoca o ID** Descrição: ID do componente Comprimento de dados: fixed(1) Formato de dados: Octeto
- **ETIQUETA ID 8? ID de correlação de TCAP** Descrição: ID do componente a que este componente correlaciona Comprimento de dados: fixed(1) Formato de dados: Octeto
- **ETIQUETA ID 9? Componente de Diálogo TCAP ANSI** Descrição: Corpo de um mensagem TCAP do primeiro parâmetro avante Comprimento de dados: Variável Formato de dados: Octeto
- **ETIQUETA ID 10? Marcador da extremidade do diálogo TCAP** Descrição: Corpo de um mensagem TCAP do primeiro parâmetro avante (SEQUÊNCIA) Comprimento de dados: fixed(0) Formato de dados: Nenhum
- **ETIQUETA ID 11? Erro** Descrição: Dados de erro Comprimento de dados: fixed(1) Formato de dados: Octeto Índice: 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 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
```

- **ETIQUETA ID 12? Deslocamento predeterminado de grupo STP-SCP** Descrição: Deslocamento predeterminado de grupo STP-SCP, dados passados da análise. Comprimento de dados: fixed(1) Formato de dados: Octeto Índice: Valor de índice do grupo STP-SCP.

- **ETIQUETA ID 13? Protocolo de transporte TCAP** Descrição: Tipo de protocolo de transporte Comprimento de dados: fixed(1) Formato de dados: Octeto Índice: 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

```

- **ETIQUETA ID 14? Erro de TCAP Externo/problema** Descrição: Valor do erro ou do problema recebido ou enviado em componentes do erro & do resultado Comprimento de dados:

Variável Formato de dados: Octeto

- **ETIQUETA ID 15? Tipo de corpo TCAP** Descrição: Tipo de corpo do componente Comprimento de dados: fixed(1) Formato de dados: Octeto Índice: 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

- **ETIQUETA ID 16? Informação de diálogo TCAP** Descrição: Trillium TCAP inclui esta ETIQUETA em todas as mensagens enviadas ao MDL. O MDL deve armazenar esta informação e enviá-la a Trillium TCAP em todos os mensagens subsequente para o diálogo ou as mensagens unidirecionais relativo ao atendimento. Comprimento de dados:

Variável Formato de dados: Octeto

- **ETIQUETA ID 17? Identificação de transação de TCAP** Descrição: Trillium TCAP inclui esta ETIQUETA em todas as mensagens enviadas ao MDL. O MDL deve armazenar esta informação para enviar ao CDB. Comprimento de dados: Variável Formato de dados: Octeto
- **ETIQUETA ID 18? Identificação do banco de dados TCAP** Descrição: Trillium TCAP incluirá esta ETIQUETA em todas as mensagens enviadas ao MDL. O MDL deve armazenar esta informação para enviar ao CDB. Comprimento de dados: Variável Formato de dados: Octeto

Apêndice B: Point code do fazer logoff 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.)

	Conjunt o	Rede	Membr o	Point code
ESTI (14 bit)	3 bit	8 bits	3 bit	14 bit
ANSI (24 bit)	8 bits	8 bits	8 bits	24 bit
PC 1-1-1 (nenhum estofamento, 14 mordido somente)	001	000 00001	001	001000 = 8 00000001 = 01

PC 1-4-1 (nenhum estofamento, 14 mordido somente)	001	0000010 0	001	001000 = 8 00100001 = 21
PC 3-3-3	011	0000001 1	011	011000 = 18 00011011 = 1B

Apêndice C: Tipos de mensagem de SCCP

Tipo de mensagem	Código do tipo de mensagem
Pedido de conexão CR	0000 0001
A conexão CC confirma	0000 0010
Conexão CREF recusada	0000 0011
RLSD liberado	0000 0100
O RLC libera completo	0000 0101
Formulário de dados DT1 1	0000 0110
Formulário de dados DT2 2	0000 0111
Reconhecimento dos dados AK	0000 1000
UDT Unitdata	0000 1001
Serviço UDTS Unitdata	0000 1010
Dados expedidos ED	0000 1011
Reconhecimento dos dados expedidos EA	0000 1100
Pedido da restauração RSR	0000 1101
Confirmação de reinicialização de RSC	0000 1110
Erro da unidade de dados de protocolo ERR	0000 1111
Teste da inatividade TI	0001 0000
Unitdata prolongado XUDT	0001 0001
Serviço prolongado do unitdata XUDTS	0001 0010
Dados de unidade longa de LUDT	0001 0011
Serviço longo do unitdata LUDTS	0001 0100

Unitdata (UDT)

O mensagem de UDT contém:

- Três ponteiros
- Os parâmetros indicados nesta tabela.

Parâmetro	Referência Q.713	Datilograf e (F V O)	Comprimento (octetos)
Tipo de mensagem	2.1	F	1
Classe de protocolo	3.6	F	1
Endereço da parte chamada	3.4	V	3 mínimos
Endereço da chamada originada	3.5	V	3 mínimos
Dados	3.16	V	2-X (nota 1)

Nota: Devido aos estudos em curso no endereço da parte chamada e chamadora SCCP, o comprimento máximo deste parâmetro precisa um estudo mais adicional. Igualmente nota-se que transferência de até 255 octetos dos dados do usuário está permitida quando o endereço da parte chamada e chamadora SCCP não inclui o título global.

Serviço de Unitdata (UDTS)

O mensagem de UDTS contém:

- Três ponteiros.
- Os parâmetros indicados nesta tabela.

Parâmetro	Referência Q.713	Datilograf e (F V O)	Comprimento (octetos)
Tipo de mensagem	2.1	F	1
Retorne a causa	3.12	F	1
Endereço da parte chamada	3.4	V	3 mínimos
Endereço da chamada originada	3.5	V	3 mínimos
Dados	3.16	V	2-X (nota)

Nota: Devido aos estudos em curso no endereço da parte chamada e chamadora SCCP, o comprimento máximo deste parâmetro precisa um estudo mais adicional. Igualmente nota-se que transferência de até 255 octetos dos dados do usuário está permitida quando o endereço da parte chamada e chamadora SCCP não inclui o título global.

Esta tabela mostra uma divisão do mensagem de SCCP da amostra para o serviço de Unitdata/Unitdata:

Parâmetr	Datilogra	Comprimen	Mensage	Mensage
----------	-----------	-----------	---------	---------

o	fe (F V O)	to (octetos)	m enviada da correlaçã o	m recebida da correlaçã o
Tipo de mensagem	F	1	09	0a
Classe de protocolo	F	1	80	01
Ponteiro do endereço da parte chamada	F	1	03	03
Ponteiro do endereço da chamada originada	F	1	07	0d
Ponteiro de dados	F	1	0b	11
Endereço da parte chamada	V	3 mínimos	04 c3 21 08 0c	04 c3? 30 00
Endereço da chamada originada	V	3 mínimos	04 c3 09 08 67	18 38 33 44 44
Dados (dados de TCAP)	V	04 c3 09 08 67 18 38 33 44 44 dados (dados de TCAP) V	52 62? 20 00	29 62? 00 10

Nota: Estas mensagens são exemplos somente e não podem refletir uma combinação de resposta de consulta real/sequência.

Causas do retorno UDTs

No serviço de Unitdata, em serviço prolongado de Unitdata, ou no mensagem de serviço longo de Unitdata, “o campo do parâmetro da causa do retorno” é um campo de um octeto que contenha a razão para um retorno da mensagem. Os bit 1 a 8 são codificados como mostrado aqui:

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.)

Anexo D: Interface MDL para Mensagem TCAP

Todas as mensagens aderem a um formato comum TLV:


- **A instância de chamada e ProcessId** - 8 bytes por muito tempo e devem ser recebidos pelo motor e ser retornados no mensagem de resposta do motor inalterado.
- **ID de mensagem** - Identifica a mensagem que é enviada ou recebida pela camada de protocolo de TCAP (valores mostrados nesta [tabela](#)).
- **O número de ID etiquetado de etiquetas e de dados da etiqueta** (etiqueta ID, comprimento de dados e dados) dita o que é mandado no mensagem TCAP ao destino remoto. Todos os tamanhos de campo são fixos à exceção do campo de dados de um artigo da etiqueta cujo o comprimento seja variável e seja definido (nos octetos) pelo comprimento de dados. Cada um dos campos comprimento total, identificação da instância de chamada e de processo, ID de mensagem, identificação da etiqueta e comprimento de dados é transmitido pelo byte mais significativo primeiramente.

Anexo E: Relação interna MDL

Internamente, uma comunicação com os objetos de máquina de estado TCAP (SMOs) é através dos sinais com dados. Qualquer tipo de dados MDL pode ser enviado com o sinal. Os nomes e os significados dos sinais e dos dados são alistados aqui.

- **LTriggerDescrição:** Este é o primeiro sinal que o LCM envia ao TCAP para começar o diálogo. Na disposição, `INTriggerElem` igualmente contém o `stpScpGroupIndex`.
`MSG_ACTION_COPY_STP_SCP_INDEX_FROM_SIGNAL_DATA` deve ser ajustado na tabela MA para que este seja usado.**Componentes:** `INTriggerElem`, `BNumberElem`, `BNumberDataElem`
- **LTriggerInformationDescrição:** Este sinal está enviado do TCAP ao LCM em resposta ao `LTrigger`, quando o diálogo continua.**Componentes:** `INTriggerElem`, `BNumberElem`, `BNumberDataElem`
- **LTriggerNextDescrição:** Este sinal é enviado do LCM ao TCAP como um pedido subsequente do disparador em um diálogo existente.**Componentes:** `INTriggerElem`, `BNumberElem`, `BNumberDataElem`
- **LTriggerReleaseDescrição:** Este sinal é o último a ser enviado do LCM ou do TCAP e pode ser enviado do TCAP em resposta ao `LTrigger` depois que uma resposta foi recebida do SCP.**Componentes:** `INErrorElem`, `BNumberElem`, `BNumberDataElem` O `INErrorElem` tem estes valores:
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.)

Informações Relacionadas

- [Notas Técnica de Softswitch Cisco PGW 2200](#)
- [Suporte à Tecnologia de Voz](#)
- [Suporte ao Produto de Voz e Comunicações Unificadas](#)
- [Troubleshooting da Telefonia IP Cisco](#) 
- [Suporte Técnico e Documentação - Cisco Systems](#)