

ASR 5000-reeks Problemen oplossen met GTPC en GTPU en bijbehorende padfouten

Inhoud

[Inleiding](#)

[Achtergrond](#)

[Relevante opdrachten](#)

[detectie van padfouten](#)

[Diverse voorbeelden van traject fail traps en uitleg](#)

[Gerelateerde Cisco Support Community-discussies](#)

Inleiding

Dit artikel behandelt het begrijpen en oplossen van problemen Evolved General Packet Radio Service Tunneling Protocol for Control Plane (EGTPC) / GTPC (3rd Generation Partnership Project Technical Specification [29.274](#)), GPRS Tunneling Protocol User Plane (GTPv1-U) (3GPP TS [29.281](#)), en het meest relevant voor dit artikel, Restoration Procedures ([3GPP TS 23.007](#)) kwesties, die draaien rond concepten zoals herstarttellers, padstoringen detectie en echo-verzoeken / antwoorden. Het bevat een lijst met relevante opdrachten en wat te zoeken in de uitvoer, samen met uitleg van gerelateerde configureerbare items. Dit is zeker een gebied waarvoor veel vragen worden gesteld. Merk op dat het oplossen van specifieke problemen met oproepbesturing op zichzelf een zeer uitgebreid aandachtsgebied is, maar dit artikel vermeldt alleen hoe statistieken voor oproepbesturing kunnen worden vastgelegd - alles daarbuiten (dat wil zeggen oproepsetup-problemen) wordt niet behandeld. Echt dit artikel richt zich vooral op het oplossen van problemen en het verifiëren van de peer-verbindingen, zonder welke er geen oproepbesturing zou zijn. Hoewel geschreven voor operators, kunnen ontwerpers / uitvoerders hier ook profiteren van de inhoud.

Voorbeelden gebruiken allemaal valse (gewijzigde) IP-adressen van het netwerk van een echte operator.

Achtergrond

Elke zijde van een GTP-verbinding (een van S5 (Serving Gateway (SGW) - Packet Data Network Gateway (PGW)), S11 (Multimedia Management Entity (MME) - SGW), S4 (Serving GPRS Service Node (SGSN) - Gateway GPRS Service NodeŽ (GGSN))) onderhoudt een herstartteller die is ontworpen om de peer te waarschuwen als de oorspronkelijke node opnieuw is opgestart, zodat de peer de juiste actie kan ondernemen. Hoewel een node een unieke teller kan gebruiken voor elke peer waarmee hij praat, is het ontwerp meestal om DEZELFDE teller te gebruiken voor alle peers, omdat wanneer de node opnieuw wordt opgestart, de teller voor alle peers moet worden verhoogd, en dus het bijhouden van een afzonderlijke waarde op peer-basis geen extra voordeel oplevert, omdat ze allemaal toch met één moeten toenemen. In feite is het gebruik van dezelfde herstartteller voor alle peers nuttig bij het oplossen van een groot netwerk waar alle

peers kunnen worden gecontroleerd om ervoor te zorgen dat de herstarttellers die zijn ontvangen dezelfde (enkele) waarde hebben voor de node waarmee die peers praten.

De herstartteller KAN worden opgenomen in alle soorten controleberichten, terwijl de herstartteller MOET worden opgenomen in alle echo-verzoeken en antwoorden. Zodra er een herstartteller verandert op een node, worden de verschillende peers hierover geïnformeerd via de volgende controleberichten die naar de respectievelijke peers worden verzonden. Wanneer een herstart plaatsvindt, per specificatie, heeft de ontvangende kant de mogelijkheid om de oproepen die het heeft tijdelijk tijdelijk te behouden of te laten vallen, de laatste is meestal wat het geconfigureerde gedrag is.

In het geval van het gebruik van echo-verzoeken, is de methode om de berichten te verzenden met een geconfigureerd interval, en als er geen antwoord wordt ontvangen na een geconfigureerd aantal herhalingen (elk na een geconfigureerde time-out), wordt het pad als down beschouwd en wordt de juiste actie ondernomen. Configuratie maakt het mogelijk dat echo's optioneel worden verzonden, en dit zou ook van toepassing zijn op de peer, wiens aanpak niet noodzakelijkerwijs hetzelfde hoeft te zijn, vooral wanneer de peer van een andere leverancier en / of andere provider zou kunnen zijn. Wat de reacties van ECHO betreft, is een knooppunt alleen verplicht om te reageren op een verzoek van ECHO - of het nu echos ontvangt of niet en met welke frequentie niet significant is - het hoeft alleen maar een echorespons te sturen voor elk verzoek en actie te ondernemen als de herstartteller wijzigingen heeft ontvangen. Natuurlijk, als het ook echo-verzoeken verstuurt en niet tijdig antwoorden ontvangt, moet het ook actie ondernemen voor dat scenario.

Merk op dat de detectie van padfouten ook kan worden gedaan via de GTP-U-verbinding (gebruikersvliegtuig) via Echo, maar ALLEEN via time-out, omdat er geen herstartteller is opgenomen in GTP-U-berichten zoals er is voor GTP-C zoals eerder besproken. GTP-U kan wel of niet hetzelfde pad gebruiken als door controleberichten. Als het hetzelfde pad is, is de GTP-U-detectie mogelijk niet zo waardevol, omdat GTP-C-detectie voldoende moet zijn.

Samenvattend, hier zijn de manieren waarop padfoutdetectie kan worden gedaan:

- Opnieuw opstarten Counter Change (via MSG Req/Rsp of Echo)
- Time-out voor GTPC Echo Req
- Time-out voor GTPU Echo Req
- Control Req Message Time-out (meestal niet geconfigureerd)

Relevante opdrachten

Om problemen met het pad op te lossen, maakt u kennis met de volgende opdrachten. Hieronder volgt een uitleg van de relevante details van elk.

- [EGTPC-service | GTPU-service] Alles weergeven
- EGTPC-peers weergeven [EGTP-service <service>]
- EGTPC-test ECHO GTP-versie {1 | 2} SRC-adres <Serviceadres> Peer-adres <Peer-adres>
- EGTPC-statistieken weergeven [[[SGW-adres | PGW-adres | MME-adres) <adres>] [alleen DEMUX]] [Alleen debug] [Verbose] | Redenen voor padfalen]
- GTPU-statistieken weergeven [peer-address <peer>]
- GTPC-statistieken weergeven [GGSN-service <service>] [SMGR-instance <instance>]
- demux-mgr statistieken weergeven <egtpinmgr | egtpegmgr | gtpumgr> all (SGW-egress | GGSN/ PGW,/SGW ingress | GTPU)

- EGTPCPathFail[Wissen], EGTPUPathFail[Wissen] SNMP-traps
- Redenen voor het verbreken van de sessie weergeven - GTPC-path-failure, GTPU-path-failure, path-failure

Andere opdrachten voor het tellen van abonnees met betrekking tot GTP:

Suboverzicht weergeven GGSN-service <service>

Subservice [Samenvatting] GTPU-service <Service> weergeven

EGTPC-sessies weergeven [EGTP-service <service>]

De basisopdracht om de status en configuratie van alle EGTP-services te controleren, is "show egtpc-service". Het grootste deel van de uitvoer moet zijn wat al bekend is via de configuratie, evenals de status moet worden GESTART. Het unieke stukje informatie dat is verzameld, is de "Restart Counter" die nergens anders te vinden is. De bijbehorende opdracht "show gtpu-service all" retourneert configureerbare items voor de GTPU-service(s) die zijn gekoppeld aan de EGTPC-services.

"Laat EGTPC-peers zien" is erg handig bij het identificeren van alle peers, inclusief statussen, herstarttellers en het aantal huidige/maximale abonnees.

- Merk op dat dit commando zowel SGW- als SGSN-peers omvat, terwijl er in het verleden een afzonderlijk commando was voor SGSN / GGSN.
- Aangezien deze opdracht alle peer-typen omvat, kan het peer-type worden gefilterd via de servicenaam (hoewel het trefwoord egtpc-service bedrieglijk ook een GGSN-servicenaam neemt).
- Als er geen abonnees zijn met de peer, is de status Inactief en is GTPC Echo uitgeschakeld. In die staat KUNNEN er voor EGTPC enkele peers zijn waarvoor de Restart-teller wordt gerapporteerd, terwijl de meeste 0 kunnen zijn. Dit gebeurt in situaties waarin de peer actief echoverzoeken verzendt waarop ASR5K zou moeten reageren. De herstartteller kan worden bepaald door de EGTPC-testopdracht uit te voeren (hieronder besproken), hoewel de teller in de inactieve staat niet wordt bijgewerkt in "show egtpc peers", ook al is deze ontvangen. Ondertussen wordt voor GGSN de herstartteller nooit vergeten.

```
[local]PGW-GGSN> show egtpc peers egtp-service GGSN1
```

	Service		Restart--+	No. of		
	ID		Counter		restarts	
					Current	Max
vvvvv	v	Peer Address	v	v	sessions	sessions
-----	---	-----	---	---	-----	-----
IDNKG	6	203.0.113.69	165	0	0	1
IDNKG	6	203.0.113.70	25	0	0	1
IDNKG	6	203.0.113.71	165	0	0	1
IDNKG	6	203.0.113.72	25	0	0	1
IDNKG	6	203.0.113.73	53	0	0	1
IDNKG	6	203.0.113.74	53	0	0	1
IDNKG	6	192.0.2.9	46	0	0	2
IDNKG	6	192.0.2.10	44	0	0	1
IDNKG	6	192.0.2.11	47	0	0	2
IDNKG	6	192.0.2.12	44	0	0	2
IDNKG	6	192.0.2.13	54	0	0	1
IDNKG	6	192.0.2.14	47	0	0	1

IDNKG	6	209.165.200.225	35	1	0	3
IDNKG	6	209.165.200.226	35	3	0	3
IDNK	6	198.51.100.204	156	0	0	1
IDNK	6	198.51.100.205	158	0	0	1
IDNK	6	198.51.100.211	54	0	0	1
IDNK	6	198.51.100.212	54	0	0	1
IDNK	6	198.51.100.213	56	0	0	1
IDNK	6	198.51.100.214	56	0	0	1
IDNK	6	198.51.100.215	54	0	0	1

Let op een paar leeftijdsgenoten hieronder die veel hoger tellen dan de andere leeftijdsgenoten. Deze zijn voor SGW's die fysiek op dezelfde locatie of de zustersite van de PGW zijn gevestigd waarnaar de MME het meest belt.

```
[local]PGW> show egtpc peers
Monday March 30 20:10:26 UTC 2015
```

+-----Status:		(I) - Inactive	(A) - Active			
+---GTPC Echo:		(D) - Disabled	(E) - Enabled			
+---Restart Counter Sent:		(S) - Sent	(N) - Not Sent			
+--Peer Restart Counter:		(K) - Known	(U) - Unknown			
+--Type of Node:		(S) - SGW	(P) - PGW			
		(M) - MME	(G) - SGSN			
		(L) - LGW	(E) - ePDG			
		(C) - CGW				
		(U) - Unknown				
	Service	Restart-----+	No. of			
	ID	Counter		restarts		
				Current	Max	
vvvvv	v	Peer Address	v	v	sessions	sessions

IDNUS	5	198.51.100.4	0	0	0	1
IDNKS	5	198.51.100.33	75	0	0	1
IDNUS	5	198.51.100.35	0	0	0	2
IDNUS	5	198.51.100.65	0	0	0	1
IDNUS	5	198.51.100.67	0	0	0	2
IDNUS	5	198.51.100.68	0	0	0	2
IDNUS	5	198.51.100.98	0	0	0	2
IDNUS	5	198.51.100.100	0	0	0	1
IDNUS	5	198.51.100.129	0	0	0	2
IDNUS	5	198.51.100.130	0	0	0	2
IDNUS	5	198.51.100.161	0	0	0	2
IDNUS	5	198.51.100.194	0	0	0	2
IDNUS	5	198.51.100.196	0	0	0	1
IDNUS	5	198.51.100.225	0	0	0	5
IDNUS	5	198.51.100.226	0	0	0	2
IDNUS	5	198.51.100.228	0	0	0	1
IDNUS	5	198.51.100.3	0	0	0	2
IDNUS	5	198.51.100.36	0	0	0	1
IDNUS	5	198.51.100.64	0	0	0	4
IDNUS	5	198.51.100.66	0	0	0	2
IDNUS	5	198.51.100.130	0	0	0	2
IDNKS	5	198.51.100.24	2	0	0	2
AESKS	5	198.51.100.56	3	0	1	5
IDNKS	5	198.51.100.99	18	0	0	2
AESKS	5	198.51.100.100	7	0	1	3
IDNKS	5	198.51.100.131	46	0	0	2
IDNKS	5	198.51.100.132	18	0	0	2

IDNKS	5	198.51.100.163	41	0	0	3
IDNKS	5	198.51.100.164	6	0	0	2
AESKS	5	198.51.100.184	11	0	1	2
IDNKS	5	198.51.100.216	7	0	0	2
AESKS	5	203.0.113.24	5	0	3	8
IDNKS	5	203.0.113.56	3	1	0	13
IDNKS	5	203.0.113.88	2	0	0	18
IDNKS	5	203.0.113.120	6	0	0	3
IDNKS	5	203.0.113.152	3	0	0	4
AESKS	5	203.0.113.184	2	0	1	9
IDNKS	5	203.0.113.216	4	0	0	2
IDNKS	5	203.0.113.248	10	0	0	1
IDNUS	5	203.0.113.4	0	0	0	2
IDNUS	5	203.0.113.33	0	0	0	4
IDNUS	5	203.0.113.65	0	0	0	2
IDNKS	5	203.0.113.121	5	0	0	3
IDNKS	5	203.0.113.153	6	0	0	5
IDNUS	5	203.0.113.162	0	0	0	3
IDNUS	5	203.0.113.32	0	0	0	5
AESKS	5	203.0.113.35	5	1	6	37
IDNUS	5	203.0.113.64	0	0	0	1
IDNUS	5	203.0.113.66	0	0	0	4
IDNUS	5	203.0.113.97	0	0	0	1
AESKS	5	203.0.113.98	4	0	1	4
IDNUS	5	203.0.113.161	0	0	0	4
IDNUS	5	203.0.113.193	0	0	0	10
IDNUS	5	203.0.113.228	0	0	0	12
IDNUS	5	203.0.113.233	0	0	0	4
AESKS	5	192.0.2.4	9	0	3	10
AESKS	5	192.0.2.24	6	0	10	19
AESKS	5	192.0.2.32	8	0	83	116
AESKS	5	192.0.2.33	55	0	14	21
AESKS	5	192.0.2.65	18	0	1	2747
AESKS	5	192.0.2.66	4	0	3	1639
AESKS	5	192.0.2.97	44	39	33	1826
AESKS	5	192.0.2.98	5	1	6	2075
IDNKS	5	192.0.2.195	8	0	0	7
AESKS	5	192.0.2.3	46	0	3230	5346
AESKS	5	192.0.2.5	12	0	84	202
AESKS	5	192.0.2.7	2	0	31	443
AESKS	5	192.0.2.24	3	0	116	117
AESKS	5	192.0.2.36	15	0	4194	6889
AESKS	5	192.0.2.44	7	0	112	331
AESKS	5	192.0.2.68	34	0	6104	9798
AESKS	5	192.0.2.69	7	0	2259	5027
AESKS	5	192.0.2.88	17	0	3899	8880
AESKS	5	192.0.2.164	16	0	833131	1298070
AESKS	5	192.0.2.184	3	0	1437053	2153317
AESKS	5	192.0.2.193	21	0	11364	26577
AESKS	5	192.0.2.195	5	0	50	162
AESKS	5	192.0.2.226	21	0	1950	3931
AESKS	5	192.0.2.227	6	0	4348	9332

Merk op dat het uitvoeren van deze opdracht op een Inter-Chassis Service Redundancy (ICSR) standby node alleen relevant is voor actieve peers, in welk geval de nummers dicht moeten zijn, bijvoorbeeld in het volgende paar:

PGW> egtpc-peers weergeven ... 20:51:52 (ACTIEF)					PGW-ICSR> toon egtpc-peers ... 20:51:51 (STANDBY)				
	peer	RC	Aantal	Max		peer	RC	Aantal	Max

			sessies	Sess			sessies	Sess	
AESKS	192.0.2.5	2	31	443	ADSKS	192.0.2.5	2	31	443
AESKS	192.0.2.24	3	117	123	DRANKJE	192.0.2.24	3	117	123
AESKS	192.0.2.36	15	4223	6889	ADSKS	192.0.2.36	15	4223	6889
AESKS	192.0.2.44	7	119	331	ADSKS	192.0.2.44	7	119	331
AESKS	192.0.2.68	34	6174	9798	ADSKS	192.0.2.68	34	6171	9797
AESKS	192.0.2.69	7	2311	5027	ADSKS	192.0.2.69	7	2311	5027
AESKS	192.0.2.88	17	3914	8880	ADSKS	192.0.2.88	17	3913	8874
AESKS	192.0.2.164	16	833705	1298070	ADSKS	192.0.2.164	16	832037	1294412
AESKS	192.0.2.184	3	1438418	2153317	ADSKS	192.0.2.184	3	1435461	2147280
AESKS	192.0.2.193	21	11447	26577	ADSKS	192.0.2.193	21	11443	26568
AESKS	192.0.2.195	5	48	162	ADSKS	192.0.2.195	5	48	162
AESKS	192.0.2.226	21	1962	3931	ADSKS	192.0.2.226	21	1962	3931
AESKS	192.0.2.227	6	4465	9332	ADSKS	192.0.2.227	6	4466	9332

"EGTPC Test ECHO" wordt gebruikt om een specifieke peer te controleren om te zien of deze bereikbaar is of niet en moet worden uitgevoerd in de context waarin de service is gedefinieerd.

- Gebruik gtp-versie 1 voor GGSN-SGSN-verbindingen
- Het gebruik van ping-commando is geen geldige test, maar als het succesvol is, is het bekend dat er een bepaald niveau van bereikbaarheid is.
- De peer restart counter (Recovery) wordt weergegeven.
- Als u deze opdracht uitvoert, worden de Tx echo request / Rx Echo response counters verhoogd in "egtpc-statistieken weergeven"
- De herstartteller van de peer wordt gerapporteerd na het gereserveerde woord Recovery (11 in het voorbeeld hieronder)
- Waarschuwing: hoewel dit niet zou moeten gebeuren in normale omstandigheden, als een test echo wordt uitgevoerd op een peer en het time-out om welke reden dan ook (als alles werkt dan moet het niet time-out), zouden alle oproepen vallen (ook al dingen waren prima werken tot het punt van het doen van de test).

```
[Ingress]PGW> egtpc test echo gtp-version 2 src-address 209.165.201.1 peer-address 198.51.100.35
Monday November 03 15:08:12 UTC 2014
EGTPC test echo
```

```
-----
Peer: 198.51.100.35 Tx/Rx: 1/1 RTT(ms): 1 (COMPLETE) Recovery: 11 (0x0B)
```

"show egtpc statistics" is voor EGTPC (v2) en rapporteert over het succes/falen van veel EGTPC-controleberichten (Request/Response, Tx/Rx), waaronder Create Session, Modify Bearer, Delete Session, Downlink Data Notify, Release Access Bearer, Create Bearer, Update Bearer, Modify Bearer, Suspend, etc., en de bijbehorende reacties. Het bevat alle tellers voor padbeheer voor Echo Request/Response Tx/Rx, en op basis van de timers kan de groei van deze timers worden voorspeld en worden gebruikt om problemen op te lossen als deze niet toenemen zoals verwacht. In het volgende voorbeeld worden echo's en reacties van beide kanten geïnitieerd en beantwoord.

Merk op dat de SGW is geconfigureerd om één keer per minuut te verzenden, terwijl de PGW is geconfigureerd voor één keer elke drie minuten (beide zijden hoeven niet dezelfde instelling te hebben) en dus elke drie minuten zijn er twee keer het aantal pakketten (6: 16: 33, 6: 19: 33). Houd er rekening mee dat hoewel beide zijden zich in dit voorbeeld op dezelfde "cyclus" bevinden, ze niet onafhankelijk van elkaar hoeven te zijn en kunnen draaien. De herstarttellers zijn ook vastgelegd in een kolom. Merk op dat de herstartteller van 11 uit de echorespons van de SGW overeenkomt met de waarde van de reactie op de PGW-testecho van bovenaf (de test werd op een ander tijdstip uitgevoerd dan de pakketopname, maar voor hetzelfde peer-paar).

No.	Time	Source	Destination	Restart Counter	Info
1	06:15:33	SGW	PGW	11	Echo Request
2	06:15:33	PGW	SGW	18	Echo Response
3	06:16:33	SGW	PGW	11	Echo Request
4	06:16:33	PGW	SGW	18	Echo Response
5	06:16:33	PGW	SGW	18	Echo Request
6	06:16:33	SGW	PGW	11	Echo Response
7	06:17:33	SGW	PGW	11	Echo Request
8	06:17:33	PGW	SGW	18	Echo Response
9	06:18:33	SGW	PGW	11	Echo Request
10	06:18:33	PGW	SGW	18	Echo Response
11	06:19:33	SGW	PGW	11	Echo Request
12	06:19:33	PGW	SGW	18	Echo Response
13	06:19:33	PGW	SGW	18	Echo Request
14	06:19:33	SGW	PGW	11	Echo Response
15	06:20:33	SGW	PGW	11	Echo Request
16	06:20:33	PGW	SGW	18	Echo Response
17	06:21:33	SGW	PGW	11	Echo Request
18	06:21:33	PGW	SGW	18	Echo Response

De optie alleen demux bevat alle tellers voor padbeheer (echoverzoeken/antwoorden) en controleberichten die nooit een sessie bereiken, wat meestal Sessieverzoek/antwoord aanmaken is. Demux-procesetpinmgr die op de Demux-kaart aanwezig is, is verantwoordelijk voor de afhandeling van alle echo-verzoeken/antwoorden en de eerste afhandeling van alle oproepcontroleberichten voordat deze worden verspreid naar de juiste sessiemakers (willekeurig voor nieuwe oproepen, en specifiek voor bestaande oproepen die op een specifieke sessiemgr zijn verankerd). De optie alleen voor sessmgr telt daarentegen alleen berichten die door sessmgr's worden afgehandeld, maar bevat GEEN tellers voor padbeheer. Geen van beide kwalificaties opgeven is de standaard voor het tellen van alle waarden.

"show gtpc statistics" is de overeenkomstige opdracht voor SGSN/GGSN-services (GTPC v1) (CPC, Update PDP Context, enz.)

"show gtpu statistics" rapporteert statistieken over het gebruikersvliegtuigverkeer dat wordt afgehandeld door de GTP-U-service(s) die zijn gekoppeld aan de EGTPC-service(s). Een interessante teller is Error Indication Tx / Rx die zou worden verzonden wanneer de ontvangende node heeft geen record van de abonnee die moet worden gekoppeld aan de Terminal Endpoint Identifier (TEID) van het pakket in kwestie, die kan gebeuren om een aantal redenen (Het bulkstats schema / variabele is GTPU1 / err-ind-tx). Als de MME/SGW niet op de hoogte is gesteld dat de PGW niet langer bindend is voor de abonnee, moeten oproepen mogelijk handmatig worden gewist op de MME/SGW (of wachten tot ze een time-out hebben) om een dergelijk probleem van verweesde oproepen volledig op te lossen.

De peer-specifieke versies van de bovenstaande stat-opdrachten zijn nodig bij het oplossen van specifieke peers, wat vaak het geval is.

detectie van padfouten

De EGTPCPathFail-trap is de sleutel om te weten wanneer een padfout is opgetreden. Onder een aantal basiswaarden die men al zou aannemen, meldt het het aantal geannuleerde oproepen, de oude en nieuwe herstarttellers en de reden voor het falen. Daarnaast wordt de reden voor het verbreken van de sessie gtpc-path-failure verhoogd.

De trap EGTPCPathFailClear geeft de herstartteller aan, samen met de reden voor het wissen van het pad. De meest recente StarOS v17 is beter geformuleerd om te zeggen "duidelijke reden" in plaats van "reden van falen" (het gewist, niet gefaald!) om verwarring te voorkomen.

Het volgende zijn het aantal mislukte paden dat wordt gerapporteerd vanuit "show egtpc stat path-failure-reasons"

```
[local]PGW> show egtpc statistics path-failure-reasons
Reasons for path failure at EGTPC:
Echo Request restart counter change:          3
Echo Response restart counter change:          0    echo-rsp-restart-counter-change
No Echo Response received:                    0    no-response-from-peer
Control message restart counter change at demux: 23    Create Session Req Restart Counter changed or
Control message restart counter change at sessmgr: 0    cpc-restart-counter-change or upc-restart-cou
Total path failures detected:                  26
```

Opmerkingen:

- Controlebericht herstart teller bij sessmgr kan elk type controlebericht zijn (niet alle hier vermeld)
- Gewijzigde herstartteller van de SGSN via de volgende Create PDP Context Request na die wijziging (cpc-restart-counter-change)
 - Gewijzigde herstartteller van de SGSN via de volgende Update PDP Context Request na die wijziging (die een overdracht van een andere SGSN zou zijn) (upc-herstart-teenverandering)
 - Pad duidelijk falen reden sessies-add is een bedrieglijke term - het is geen mislukking, maar een succes, en het geeft aan dat een nieuwe sessie is gekomen om de verbinding tot stand te brengen
 - Als de waarden die zijn gerapporteerd voor herstarttellers en sessietellingen voor de traps PathFail en PathClear niet op één lijn lijken te staan, opent u een ticket met Cisco voor verdere uitleg.

Het is belangrijk om te begrijpen hoe EGTPC/GTPC en in mindere mate EGTPU/GTPU (aangezien de detectie van storingen in het gebruikersvlak niet altijd is geconfigureerd) wordt geïmplementeerd om de verbinding tussen een GGSN of PGW en de verschillende peer-SGSN en SGW's te bewaken. Normaal gesproken zijn de verbindingen binnen een serviceproviders netwerk redelijk stabiel, in welk geval pad storingen zijn zeldzaam. Aan de andere kant heeft de ervaring geleerd dat GTPC-paden tussen roamende SGSN's die verbinding maken met GGSN's mogelijk niet zo stabiel zijn, en dus met slechte verbindingen is het resultaat dat padstoringen op een vrij frequente basis plaatsvinden.

De volgende configuratie heeft betrekking op timers en transmissiefrequentie voor besturings- en echoberichten die kunnen worden gebruikt om padstoringen te detecteren. Hetzelfde concept geldt voor GGSN-, SGW-, MME- en EGTP-diensten:

ggsn-service GGSN1

tijdpad voor doorgifte 20

Voor GTP-controleberichten (versus de echoberichten)

Max. hertransmissie 5	voor ZOWEL GTP-besturings- als echoberichten – hoe vaak opnieuw moet worden verzonden voordat de detectie van padfouten wordt geactiveerd
echo-interval 120	Een echo wordt elke 2 minuten (120 seconden) verzonden.
echo-retransmission-timeout 5	Voor Echos, als er geen reactie is, wordt het elke 5 seconden opnieuw verzonden
detectiebeleid voor padstoringen-gtp echo	Padfout detectie is mogelijk via gtp echo (versus het niet gebruiken van deze als een methode voor het detecteren van padfout (maar nog steeds het verzenden van echo's).
associate gtpu-service GGSN1_gtpu	Koppelt de genoemde GTPU-service GGSN1_gtpu aan deze GGSN-service
Binding IPv4-adres 209.165.201.2	De IP-adressen van de gebruiker en het controlevlak zijn <u>hetzelfde</u> in de volgende voorbeeldconfiguraties, maar hoeven niet noodzakelijkerwijs als zodanig te worden geconfigureerd met andere implementaties van serviceproviders.

```
[local]GGSN> show ggsn-service name GGSN1
```

```
Service name:      GGSN1
Context:           XGWin
Bind:              Done
Local IPv4 Address: 209.165.201.2      Local IP Port: 2123
Retransmission Timeout: 20 (secs)
Max Retransmissions: 5
Restart Counter:   41
Echo Interval:     120 (secs)
GTPC Echo Mode      : Default
GTPC Echo Retransmission Timeout : 5
Guard Interval:     100 (secs)
Path Failure Detection on gtp msgs: Echo
```

Opmerkingen:

- Aangezien de tijd 0 is wanneer een Echo-aanvraagcyclus start (elke 2 minuten), mislukt het pad als er na een TOTAAL van 30 seconden (6 pogingen) geen reacties zijn (0 + 5 sec + (5 * 5 sec))
- Process egtpinmgr (gtpcmgr afgeschreven sinds v14) draait op de demux kaart is verantwoordelijk voor de verwerking van de echo pakketten
- "toon gtp-statistieken", sectie "Berichten voor padbeheer" bevat statistieken voor de echo-verzoeken/antwoorden voor het controlepad in BEIDE richtingen:

```
[Ingress]GGSN> show gtpc statistics
Monday October 27 23:02:07 UTC 2014
```

Path Management Messages:

```
Echo Request RX:      169951248   Echo Response TX:      169951248
Echo Request TX:      40838432    Echo Response RX:      40784820
```

Een andere methode voor paddetectie bevindt zich in het gebruikersvlak in plaats van in het controlevlak, en dit wordt geregeld door de volgende configuratie.

```
gtpu-service GGSN1_gtpu
no echo-interval
max-retransmissions 4
echo-retransmission-timeout 5
path-failure detection-policy gtp echo
no path-failure clear-trap gtp echo
bind ipv4-address 209.165.201.2
exit
```

Opmerkingen:

- "no echo-interval" schakelt het verzenden van echo-verzoeken voor gebruikersvlak uit en daarom heeft CLI "path-failure detection-policy gtp echo" geen relevantie – er wordt GEEN detectie van padfouten uitgevoerd op GTPU
- GGSN reageert echter nog steeds op echo-verzoeken van de SGSN
- "Show GTPU STAT" en "Show DEMUX-MGR STAT GTPUMGR" leggen echostatistieken vast

```
[local]PGW-GGSN> show gtpu statistics
Wednesday October 29 23:43:22 UTC 2014
Path Management Messages:
Echo Request Rx:          84690303  Echo Response Rx:          0
Echo Request Tx:          0  Echo Response Tx:          84690043
```

```
[local]PGW-GGSN> show demux-mgr statistics gtpumgr
Wednesday October 29 23:42:04 UTC 2014
Echo stats:
Echo Req Rcvd:            84690303
Echo Rsp Sent:            84690043
```

Diverse voorbeelden van traject fail traps en uitleg

Meestal zal de peer restart teller met één toenemen wanneer er een herstart is. In dit geval zorgde een bug in PGW ervoor dat het de herstartteller overschreefde ten opzichte van de waarde van de RC van een andere peer (eerste val), en dus toen het volgende pakket over de verbinding arriveerde, behandelde het het als een herstart (tweede val) denkend dat de RC was veranderd. Het pad wordt vervolgens op elk opeenvolgend pakket (derde val) gewist. Het punt hier is niet om te weten over een specifieke bug, maar eerder om te proberen op de hoogte te zijn van eventuele tegenveranderingen die niet kunnen worden verklaard.

Wed Jan 14 05:22:07 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service EGTP1, i

Wed Jan 14 05:22:07 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service EGTP1, i

Wed Jan 14 05:22:07 2015 Internal trap notification 1113 (EGTPCPathFailClear) context XGWin, service EGTP1, i

Dit voorbeeld is van een GGSN waar de peer SGSN zijn RC heeft gewijzigd van 66 naar 67 via een Create PDP Context Request:

Mon Oct 27 11:48:54 2014 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service GGSN1, i

Mon Oct 27 11:48:54 2014 Internal trap notification 1113 (EGTPCPathFailClear) context XGWin, service GGSN1, i

Dit voorbeeld is van een PGW waarbij de peer-ePDG zijn RC heeft gewijzigd van 69 naar 70 via een Sessieverzoek aanmaken:

Fri Jun 12 05:21:54 2015 Internal trap notification 1112 (EGTPCPathFail) context PGWin, service EGTP1, i

Fri Jun 12 05:21:55 2015 Internal trap notification 1113 (EGTPCPathFailClear) context PGWin, service EGTP1, i

In dit voorbeeld laten de uitvoer van verschillende PGW-knooppunten die allemaal met dezelfde SGW zijn verbonden, zien dat de SGW niet meer bereikbaar is vanaf de PGW's per EGTPCPathFail-val bij de verschillende PGW's binnen een kort tijdsbestek van elkaar. Merk op dat alle tonen dezelfde oude herstart teller als eerder werd besproken dat normaal gesproken een knooppunt gebruikt dezelfde teller voor al zijn collega's. In het geval dat er geen reactie is, wordt de nieuwe herstartteller weergegeven op 9, maar de nieuwe teller kan beter als ongedefinieerd worden afgedrukt, omdat de verbinding is uitgeschakeld vanaf het moment dat de trap wordt geactiveerd en dus heeft een herstartteller geen zin.

Mon Dec 22 09:07:29 2014 Internal trap notification 1112 (EGTPCPathFail) context PGWin, service EGTP1, i

Mon Dec 22 09:08:54 2014 Internal trap notification 1112 (EGTPCPathFail) context PGWin, service EGTP1, i

Mon Dec 22 09:09:00 2014 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service EGTP1, i

Mon Dec 22 09:07:42 2014 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service EGTP1, i

Mon Dec 22 09:08:06 2014 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service EGTP1, i

Mon Dec 22 09:07:34 2014 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service EGTP1, i

Mon Dec 22 09:08:20 2014 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service EGTP1, i

Mon Dec 22 09:06:37 2014 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service EGTP1, i

Mon Dec 22 09:07:29 2014 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service EGTP1, i

Mon Dec 22 09:07:01 2014 Internal trap notification 1112 (EGTPCPathFail) context PGWin, service EGTP1, i

Het volgende voorbeeld toont een padfout en dezelfde nieuwe herstartteller als in het vorige

voorbeeld, maar een half uur later wordt weergegeven wanneer de peer weer omhoog komt en de teller met één is gestegen van 15 naar 16. Na verloop van tijd neemt het aantal toe voor die peer:

<#root>

Wed Jul 22 05:07:08 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service EGTP1, i

Wed Jul 22 05:38:00 2015 Internal trap notification 1113 (EGTPCPathFailClear) context XGWin, service EGT

16

, peer session count 1, clear reason sessions-update

[local]PGW> show egtpc peers

Wednesday July 22 10:04:30 UTC 2015

	Service	Restart-----+	No. of
	ID	Counter	restarts
vvvvv	v	Peer Address	v
			v
			Current
			sessions
			Max
			sessions

AESKS	5	198.51.100.162	

16

1 1157 2186

[XGWin]PGW> egtpc test echo gtp-version 2 src-address 209.165.201.13 peer-address 198.51.100.162

Wednesday July 22 10:52:41 UTC 2015

EGTPC test echo

Peer: 198.51.100.162

Tx/Rx: 1/1 RTT(ms): 51 (COMPLETE) Recovery:

16

(0x10)

In dit voorbeeld wordt een fout in het herhalingspad weergegeven als gevolg van het **opnieuw opstarten van de CPC** voor drie specifieke SGSN-peers met de GGSN. De output hieronder toont slechts een van de peers 209.165.200.225, maar ze hebben allemaal hetzelfde probleem. Herhaalde run van "show egtpc peers" en "show gtpc stat sgsn-address" toont ook de wisselende waarden. "show gtpc stat" bevestigt geen probleem met connectiviteit met betrekking tot echo-verzoeken en antwoorden, die in dit geval alleen worden geïnitieerd door de GGSN (aanvaardbaar zoals eerder besproken). Een zijbalk punt om hier te worden gemaakt is dat er vaak meerdere stukken van gegevens die kunnen worden verzameld die allemaal bevestigen aan dezelfde conclusie - men kan ervoor kiezen om de gegevens die het meest overtuigend met de minimale hoeveelheid inspanning te verzamelen.

<#root>

show snmp trap history verbose | grep -E "209.165.201.31|209.165.200.225|209.165.200.246"

Mon Jun 22 22:20:51 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service GGSN1, i

132

, peer session count 3, failure reason

cpc-restart-counter-change

Mon Jun 22 22:20:51 2015 Internal trap notification 1113 (EGTPCPathFailClear) context XGWin, service GGSN1, i
Mon Jun 22 22:20:55 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service GGSN1, i

129

, peer session count 3, failure reason cpc-restart-counter-change
Mon Jun 22 22:20:55 2015 Internal trap notification 1113 (EGTPCPathFailClear) context XGWin, service GGSN1, i
Mon Jun 22 22:20:58 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service GGSN1, i

132

, peer session count 3, failure reason cpc-restart-counter-change
Mon Jun 22 22:20:58 2015 Internal trap notification 1113 (EGTPCPathFailClear) context XGWin, service GGSN1, i
Mon Jun 22 22:21:02 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service GGSN1, i

129

, peer session count 2, failure reason cpc-restart-counter-change
Mon Jun 22 22:21:02 2015 Internal trap notification 1113 (EGTPCPathFailClear) context XGWin, service GGSN1, i
Mon Jun 22 22:21:03 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service GGSN1, i

132

, peer session count 4, failure reason echo-rsp-restart-counter-change
Mon Jun 22 22:21:03 2015 Internal trap notification 1113 (EGTPCPathFailClear) context XGWin, service GGSN1, i
Mon Jun 22 22:21:06 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service GGSN1, i

129

, peer session count 2, failure reason cpc-restart-counter-change
Mon Jun 22 22:21:06 2015 Internal trap notification 1113 (EGTPCPathFailClear) context XGWin, service GGSN1, i

[XGWin]GGSN> show egtpc peers | grep 209.165.200.225

	Service		Restart No. of			
	ID		Counter	restarts		
					Current	Max
vvvvv	v	Peer Address			sessions	sessions
			v	v	v	

Monday June 22 23:09:26 UTC 2015
AESKG 6 209.165.200.225

129

86339

2 57

[XGWin]GGSN> show egtpc peers | grep 209.165.200.225

Monday June 22 23:09:28 UTC 2015
AESKG 6 209.165.200.225

129

86341 3 57

[XGWin]GGSN> show egtpc peers | grep 209.165.200.225

Monday June 22 23:09:29 UTC 2015
AESKG 6 209.165.200.225

132

86342 7 57

```
[XGWin]GGSN> show egtpc peers | grep 209.165.200.225
Monday June 22 23:09:30 UTC 2015
AESKG 6 209.165.200.225
```

132

```
86344 9 57
```

```
[XGWin]GGSN> show egtpc peers | grep 209.165.200.225
Monday June 22 23:09:31 UTC 2015
AESKG 6 209.165.200.225
```

129

```
86345 11 57
```

```
[XGWin]GGSN> show egtpc peers | grep 209.165.200.225
Monday June 22 23:09:32 UTC 2015
AESKG 6 209.165.200.225
```

129

86345

```
7 57
```

```
[XGWin]GGSN> show gtpc statistics sgsn-address 209.165.200.225
Monday June 22 23:07:36 UTC 2015
SGSN Address: 209.165.200.225 Status: Active
Total Restarts:
```

86287

Restart Counter:

129

```
Session Stats:
Total Current: 2 S6b Assume Positive: 0
...
```

```
Path Management Messages:
Echo Request RX: 0 Echo Response TX: 0
Echo Request TX:
```

22919

Echo Response RX:

22917

Kijkend naar andere peers in het subnet, ze tonen allemaal een RC = 132 de hele tijd, terwijl de gebroken peers afwisselen tussen 129 en 132 zoals zojuist getoond, wat wijst op deze SGSN-peers als mogelijk het passeren van de verkeerde RC in de gebroken scenario's:

<#root>

```
Restart
Counter
AESKG 6 209.165.200.225 129
```

85115

```
8          57 <==
IDNKG  6      209.165.200.226 132 21      0      3
AESKG  6      209.165.200.227 132 60      47      92
AESKG  6      209.165.200.228 132 53      24      59
AESKG  6      209.165.200.229 132 69      92     126
AESKG  6      209.165.200.232 132 66      21      30
AESKG  6      209.165.200.236 132 43      8       21
AESKG  6      209.165.200.238 132 49      18      32
AESKG  6      209.165.200.239 132 39      1       6
AENKG  6      209.165.200.240 132 17      1       5
AESKG  6      209.165.200.241 132 67      6      43
AESKG  6      209.165.200.242 132 56      22      29
AENKG  6      209.165.200.243 132 40      2       9
AENKG  6      209.165.200.246 129
```

17713

```
3          24 <==
AESKG  6      209.165.200.247 132 64      29      45
AESKG  6      209.165.200.250 132 62      42      56
```

Herinner je van eerder, een andere benadering van kwesties als deze is om andere GGSN's (of PGW's voor LTE) te controleren om te zien of ze hetzelfde probleem hebben met die schijnbaar problematische SGSN-collega's, en in dit geval doen ze dat ook. Voor GGSN3 toonde de eerste run van het commando waarden 132 voor beide peers, maar een andere run toonde 129 voor een van de peers. Meerdere runs (hier niet weergegeven voor de beknoptheid) zouden alle GGSN's laten zien die in de loop van de tijd tussen de twee RC's voor die leeftijdsgenoten stuiteren:

```
[local]GGSN1> show egtpc peers | grep -E "209.165.200.246|209.165.200.225"
Tuesday June 23 11:37:23 UTC 2015
AENKG  7      209.165.200.246 129 20147      1      29
AESKG  7      209.165.200.225 129 92853      2      51
```

```
[local]GGSN2> show egtpc peers | grep -E "209.165.200.246|209.165.200.225"
Tuesday June 23 11:37:23 UTC 2015
AENKG  6      209.165.200.246 129 20450      4      17
AESKG  6      209.165.200.225 129 94021      3      54
```

```
[local]GGSN3> show egtpc peers | grep -E "209.165.200.246|209.165.200.225"
Tuesday June 23 11:37:26 UTC 2015
AENKG  6      209.165.200.246 132 20211      1      39
AESKG  6      209.165.200.225 132 91996      3      59
```

```
[local]GGSN4> show egtpc peers | grep -E "209.165.200.246|209.165.200.225"
Tuesday June 23 11:37:25 UTC 2015
AENKG  6      209.165.200.246 129 20309      1      24
AESKG  6      209.165.200.225 132 92694      9      57
```

```
[local]GGSN3> show egtpc peers | grep -E "209.165.200.246|209.165.200.225"
Tuesday June 23 11:44:35 UTC 2015
AENKG  6      209.165.200.246 129 20242      3      39
AESKG  6      209.165.200.225 132 92224      1      59
```

Tot slot om het probleem te bevestigen is de SGSN het wijzigen van de RC, monitor abonnee menu optie door SGSN peer toont veranderende RC's. Wanneer het wel verandert, worden alle

oproepen losgekoppeld van **padstoring** als de reden voor het loskoppelen. Een screenshot van een pakketopname laat dit ook zien:

<#root>

[XGWin]GGSN> mon sub

Monday June 22 23:34:22 UTC 2015

y) By SGSN IP Address

IP Address: [209.165.200.225]

INBOUND>>>> 23:35:32:688 Eventid:47000(3)

GTPC Rx PDU, from 209.165.200.225:2123 to 209.165.202.129:2123 (176)

TEID: 0x00000000, Message type: GTP_CREATE_PDP_CONTEXT_REQ_MSG (0x10)

Sequence Number:: 0x1DB8 (7608)

...

IMSI: 300420078559902

Recovery: 0x81 (129)

...

CONTROL 23:36:00:363 Eventid:10285

Disconnect Reason: path-failure

INBOUND>>>> 23:36:48:414 Eventid:47000(3)

GTPC Rx PDU, from 209.165.200.225:2123 to 209.165.202.129:2123 (176)

TEID: 0x00000000, Message type: GTP_CREATE_PDP_CONTEXT_REQ_MSG (0x10)

Sequence Number:: 0x3E2D (15917)

...

IMSI: 300420125984926

Recovery: 0x84 (132)

)

...

INBOUND>>>> 23:37:28:337 Eventid:47000(3)

GTPC Rx PDU, from 209.165.200.225:2123 to 209.165.202.129:2123 (176)

TEID: 0x00000000, Message type: GTP_CREATE_PDP_CONTEXT_REQ_MSG (0x10)

Sequence Number:: 0x3517 (13591)

...

IMSI: 300420094205377

Recovery: 0x84 (132)

...

Monday June 22 2015

INBOUND>>>> 23:37:40:559 Eventid:47000(3)

GTPC Rx PDU, from 209.165.200.225:2123 to 209.165.202.129:2123 (176)

TEID: 0x00000000, Message type: GTP_CREATE_PDP_CONTEXT_REQ_MSG (0x10)

Sequence Number:: 0x4E47 (20039)

...

IMSI: 300420194755472

Recovery: 0x81 (129)

...

CONTROL 23:37:40:755 Eventid:10285

Disconnect Reason: path-failure

No.	Time	Source	Destination	Recovery	Info
42	17:50:03	SGSN	GGSN	129	Create PDP context request
63	17:50:03	SGSN	GGSN	129	Create PDP context request
68	17:50:04	SGSN	GGSN	129	Create PDP context request
69	17:50:04	SGSN	GGSN	129	Create PDP context request
70	17:50:04	SGSN	GGSN	132	Create PDP context request
94	17:50:06	SGSN	GGSN	129	Create PDP context request
157	17:50:11	SGSN	GGSN	132	Create PDP context request
164	17:50:12	SGSN	GGSN	132	Create PDP context request
166	17:50:12	SGSN	GGSN	129	Create PDP context request
167	17:50:12	SGSN	GGSN	132	Create PDP context request
169	17:50:13	SGSN	GGSN	132	Create PDP context request
171	17:50:13	SGSN	GGSN	132	Create PDP context request
195	17:50:14	SGSN	GGSN	129	Create PDP context request
209	17:50:16	SGSN	GGSN	132	Create PDP context request
249	17:50:17	SGSN	GGSN	129	Create PDP context request
397	17:50:18	SGSN	GGSN	132	Create PDP context request
398	17:50:18	SGSN	GGSN	129	Create PDP context request
436	17:50:20	SGSN	GGSN	129	Create PDP context request
490	17:50:22	SGSN	GGSN	129	Create PDP context request
495	17:50:23	SGSN	GGSN	129	Create PDP context request
688	17:50:26	SGSN	GGSN	132	Create PDP context request
756	17:50:27	SGSN	GGSN	132	Create PDP context request
772	17:50:30	SGSN	GGSN	132	Create PDP context request
774	17:50:31	SGSN	GGSN	132	Create PDP context request
779	17:50:31	SGSN	GGSN	132	Create PDP context request
855	17:50:32	SGSN	GGSN	129	Create PDP context request

Dit voorbeeld toont een padfout waarbij het probleem zich daadwerkelijk in het pad bevond en niet de PGW of SGW. Als gevolg daarvan blijft de RC hetzelfde wanneer het pad opnieuw wordt ingesteld. In het tweede peer-voorbeeld is het duidelijke een resultaat van de oprichting van een echo-verzoek:

Thu May 28 20:47:47 2015 Internal trap notification 1112 (EGTPCPathFail) context PGWin, service EGTP1, i

Thu May 28 20:49:36 2015 Internal trap notification 1113 (EGTPCPathFailClear) context PGWin, service EGTP1, i

Thu May 28 20:49:06 2015 Internal trap notification 1112 (EGTPCPathFail) context PGWin, service EGTP1, i

Thu May 28 20:50:49 2015 Internal trap notification 1113 (EGTPCPathFailClear) context PGWin, service EGTP1, i

Dit voorbeeld toont een andere anomalie waarbij herhaaldelijk nadat de oude herstartteller is gereset naar 2, deze op de een of andere manier weer op 0 eindigt wanneer het pad faalt. Dit bleek een bug te zijn. Nogmaals, de les hier is dat als iets vreemd lijkt, het moet worden gemeld.

Fri Mar 27 03:13:36 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service EGTP1, i

Fri Mar 27 03:13:37 2015 Internal trap notification 1113 (EGTPCPathFailClear) context XGWin, service EGT
Sat Mar 28 02:20:45 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service EGTP1, i
Sat Mar 28 02:20:47 2015 Internal trap notification 1113 (EGTPCPathFailClear) context XGWin, service EGT
Sat Mar 28 17:14:06 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service EGTP1, i
Sat Mar 28 17:14:06 2015 Internal trap notification 1113 (EGTPCPathFailClear) context XGWin, service EGT

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.