Dépannage des GTPC et GTPU de la gamme ASR 5000 et défaillances de chemins associées

Table des matières

Introduction

Fond

Commandes pertinentes

Détection des pannes de chemin

Divers exemples de pièges d'échec de chemin et explications

Discussions connexes de la communauté d'assistance Cisco

Introduction

Cet article traite de la compréhension et du dépannage du protocole général évolué de tunnellisation de service radio par paquets pour le plan de contrôle (EGTPC) / GTPC (3rd Generation Partnership Project Technical Specification 29.274), du protocole de tunnellisation GPRS User Plane (GTPv1-U) (3GPP TS 29.281), et plus particulièrement de cet article, des procédures de restauration (3GPP TS 23.007) , qui tournent autour de concepts tels que les compteurs de redémarrage, la détection de chemin et l'écho demandes/réponses. Il inclut une liste des commandes pertinentes et des éléments à rechercher dans le résultat, ainsi qu'une explication des configurations associées. Il s'agit certainement d'un domaine pour lequel de nombreuses questions sont soulevées. Notez que le dépannage de problèmes de contrôle d'appel spécifiques est en soi un domaine d'intérêt très étendu, mais cet article ne mentionne que la façon de capturer des statistiques pour le contrôle d'appel - tout ce qui dépasse cela (par exemple les problèmes d'établissement d'appel) n'est pas couvert. En fait, cet article se concentre principalement sur le dépannage et la vérification des connexions homologues, sans lesquelles il n'y aurait pas de contrôle d'appel. Bien qu'écrit pour les opérateurs, les concepteurs / implémenteurs pourraient également bénéficier du contenu ici.
Tous les exemples utilisent de fausses adresses IP (modifiées) à partir du réseau d'un opérateur réel.

Fond

Chaque côté d'une connexion GTP (l'un de S5 (Serving Gateway (SGW) - Packet Data Network Gateway (PGW)), S11 (Multimedia Management Entity (MME) - SGW), S4 (Serving GPRS Service Node (SGSN) - Gateway GPRS Service Node (GGSN)) maintient un compteur de redémarrage conçu pour alerter l'homologue si le noeud d'origine a redémarré afin que cet homologue puisse prendre l'action appropriée. Alors qu'un noeud peut utiliser un compteur unique pour chaque homologue avec lequel il communique, la conception tend à utiliser le MÊME compteur pour tous les homologues, puisque lorsque le noeud redémarre, le compteur doit être incrémenté pour tous les homologues, et donc le suivi d'une valeur séparée sur une base homologue ne sert aucun avantage supplémentaire puisqu'ils doivent tous augmenter d'un de

toute façon. En fait, l'utilisation du même compteur de redémarrage pour tous les homologues est utile pour dépanner un grand réseau où tous les homologues peuvent être vérifiés pour s'assurer que les compteurs de redémarrage reçus ont la même valeur (unique) pour le noeud avec lequel ces homologues parlent.
Le compteur de redémarrage PEUT être inclus dans tous les types de messages de contrôle, tandis que le compteur de redémarrage DOIT être inclus dans toutes les requêtes et réponses d'écho. Dès qu'un changement de compteur de redémarrage se produit sur un noeud, les différents homologues en sont informés par l'intermédiaire des messages de contrôle suivants envoyés aux homologues respectifs. Lorsqu'un redémarrage se produit, par spécification, le côté récepteur a la possibilité de maintenir temporairement les appels pendant un court moment ou de les abandonner, ce dernier qui est typiquement ce qui est le comportement configuré.
Dans le cas de l'utilisation de requêtes d'écho, la méthode consiste à envoyer les messages sur un intervalle configuré, et si aucune réponse n'est reçue après un nombre configuré de nouvelles tentatives (chacune après un délai d'attente configuré), le chemin est considéré comme inactif et l'action appropriée est prise. La configuration permet l'envoi facultatif d'échos, ce qui s'applique également à l'homologue, dont l'approche n'est pas nécessairement la même, en particulier lorsque l'homologue peut provenir d'un fournisseur différent et/ou d'un opérateur différent. En effet, en ce qui concerne les réponses Echo, un noeud n'est obligé de répondre qu'à une requête Echo - qu'il reçoive ou non des échos et avec quelle fréquence n'est pas significative - il a juste besoin d'envoyer une réponse d'écho pour chaque requête, et d'agir si le compteur de redémarrage a reçu des modifications. Bien sûr, comme déjà mentionné, SI elle envoie également des requêtes d'écho et ne reçoit pas de réponses en temps utile, elle doit également prendre des mesures pour ce scénario.
Notez que la détection de défaillance de chemin peut également être effectuée sur la connexion GTP-U (plan utilisateur) via Echo, mais UNIQUEMENT via le délai d'attente, puisqu'il n'y a pas de compteur de redémarrage inclus dans les messages GTP-U comme il y en a pour GTP-C comme discuté précédemment. Le protocole GTP-U peut utiliser ou non le même chemin que les messages de contrôle. S'il s'agit du même chemin, la détection GTP-U n'est peut-être pas aussi intéressante, car la détection GTP-C doit être suffisante.
En résumé, voici comment la détection des pannes de chemin peut être effectuée :

- Changement de compteur de redémarrage (via le contrôle msg Req/Rsp ou Echo)
- Expiration de la demande d'écho GTPC
- Expiration du délai de demande d'écho GTPU
- Délai d'expiration du message de demande de contrôle (généralement non configuré)

Commandes pertinentes

Afin de dépanner tout type de problème de chemin, familiarisez-vous avec les commandes suivantes. Vous trouverez ci-dessous une explication des détails pertinents de chacun d'eux.

- show [egtpc-service | gtpu-service] all
- show egtpc peers [egtp-service <service>]
- egtpc test echo gtp-version {1} | 2} src-address <adresse de service> peer-address <adresse d'homologue>
- show egtpc statistics [[(adresse-sgw) | adresse-pgw | adresse-mem) <adresse>] [demux-only]] [debug-only] [verbose] | raisons-d'échec-chemin]

- show gtpu statistics [adresse-homologue <homologue>]
- show gtpc statistics [ggsn-service <service>] [smgr-instance <instance>]
- show demux-mgr statistics <egtpinmgr | egtpegmgr | gtpumgr> tous (SGW-egress | GGSN/PGW,/SGW en entrée | GTPU)
- EGTPCPathFail[Clear], EGTPUPathFail[Clear] Interruptions SNMP
- show session disconnect-reason - gtpc-path-failure, gtpu-path-failure, path-failure

Autres commandes de comptage des abonnés associées au protocole GTP :
show sub [summary] ggsn-service <service>
show sub [summary] gtpu-service <service>
show egtpc sessions [egtp-service <service>]
La commande de base pour vérifier l'état et la configuration de tous les services EGTP est « show egtpc-service ». La plupart des informations affichées doivent correspondre à ce qui est déjà connu via la configuration, et l'état doit être STARTED. L'information unique glanée est le « Compteur de redémarrage » qui ne peut être trouvé nulle part ailleurs. La commande associée "show gtpu-service all" retourne des configurations pour le ou les services GTPU associés aux services EGTPC.
« show egtpc peers » est très utile pour identifier tous les homologues, y compris les états, les compteurs de redémarrage et le nombre d'abonnés actuel/max.

- Notez que cette commande couvre à la fois les homologues SGW et SGSN, alors que dans le passé il y avait une commande séparée pour SGSN/GGSN.
- Comme cette commande inclut tous les types d'homologue, le type d'homologue peut être filtré sur via le nom de service (même si le mot clé egtpc-service prend également un nom de service GGSN de façon trompeuse).
- S'il n'y a aucun abonné avec l'homologue, l'état est Inactif et l'écho GTPC est désactivé. Dans cet état, pour EGTPC, il PEUT y avoir certains homologues pour lesquels le compteur Restart est signalé, alors que la plupart peuvent être 0. Cela se produit dans les situations où l'homologue envoie activement des requêtes d'écho auxquelles ASR5K devrait répondre. Le compteur de redémarrage peut être déterminé en exécutant la commande de test EGTPC (discutée ci-après), bien que dans l'état inactif, le compteur ne sera pas mis à jour dans « show egtpc peers » même s'il a été reçu. Pendant ce temps, pour GGSN, le compteur de redémarrage n'est jamais oublié.

```
[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
```

Notez quelques homologues ci-dessous qui ont des comptes beaucoup plus élevés que les autres homologues. Il s'agit des SGW physiquement situés sur le même site ou sur le site frère du PGW vers lequel le MME achemine la plupart des appels.

```
[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

Notez que l'exécution de cette commande sur un noeud de secours ICSR (Inter-Chassis Service Redundancy) ne concerne que les homologues actifs, auquel cas les numéros doivent être proches, par exemple dans la paire suivante :

PGW> show egtpc peers ... 20:51:52 (ACTIF)						PGW-ICSR> show egtpc peers ... 20:51:51 (VEILLE)			
	Pair	RC	Nombre de sessions	Max Sess	Pair	RC	Nombre de sessions	Max 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	ADNKS	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" est utilisé pour vérifier un homologue spécifique pour voir s'il est accessible ou non et doit être exécuté dans le contexte où le service est défini.

- Utilisez gtp-version 1 pour les connexions GGSN-SGSN
- L'utilisation de la commande ping n'est pas un test valide, bien que si elle réussit, on sache qu'il existe un certain niveau d'accessibilité.
- Le compteur de redémarrage de l'homologue (Récupération) s'affiche.
- L'exécution de cette commande incrémentera les compteurs de requête d'écho Tx / réponse d'écho Rx dans « show egtpc statistics »
- Le compteur de redémarrage de l'homologue est signalé après le mot clé Recovery (11 dans l'exemple ci-dessous)
- Avertissement : bien que cela ne devrait pas se produire dans des circonstances normales, si un écho de test est effectué sur un homologue et qu'il expire pour une raison quelconque (si tout fonctionne alors il ne devrait pas expirer), tous les appels seraient abandonnés (même si les choses fonctionnaient très bien jusqu'au point de faire le 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" est destiné à EGTPC (v2) et signale les succès/échecs de nombreux messages de contrôle EGTPC (Request/Response, Tx/Rx), y compris Create Session, Modify Bearer, Delete Session, Downlink Data Notify, Release Access Bearer, Create Bearer, Update

Bearer, Modify Bearer, Suspend, etc., et les réponses associées. Il inclut tous les compteurs de gestion de chemin pour la requête/réponse d'écho Tx/Rx, et sur la base des compteurs, la croissance de ces compteurs peut être prévue et utilisée pour le dépannage si elle n'augmente pas comme prévu.
L'exemple suivant montre les échos et les réponses initiés et auxquels les deux parties ont répondu. Notez que le SGW a été configuré pour envoyer une fois par minute tandis que le PGW a été configuré pour une fois toutes les trois minutes (les deux côtés n'ont pas besoin d'avoir le même paramètre) et donc toutes les trois minutes il y a deux fois le nombre de paquets (6:16:33, 6:19:33). Gardez à l'esprit que bien que les deux côtés soient sur le même "cycle" dans cet exemple, ils ne doivent pas être et peuvent fonctionner indépendamment l'un de l'autre. Les compteurs de redémarrage ont également été capturés dans une colonne. Notez que le compteur de redémarrage de 11 de la réponse d'écho du SGW correspond à la valeur de sa réponse à l'écho de test du PGW ci-dessus (le test a été effectué à un moment différent de la capture de paquets, mais pour la même paire d'homologues).

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

L'option demux-only capture tous les compteurs de gestion de chemin (requêtes/réponses d'écho) ainsi que les messages de contrôle qui n'atteignent jamais un sessmgr, qui est généralement Create Session Request/Response. Demux process etpinmgr résidant sur la carte Demux est responsable du traitement de toutes les requêtes/réponses d'écho et du traitement initial de tous les messages de contrôle d'appel avant de les distribuer aux sessmgr appropriés (aléatoirement pour les nouveaux appels, et spécifiquement pour les appels existants ancrés sur un sessmgr spécifique). L'option sessmgr-only ne compte inversement que les messages qui sont traités par sessmgr, ce qui n'inclut PAS les compteurs de gestion de chemin. Ne spécifier aucun qualificatif est la valeur par défaut pour compter toutes les valeurs.
"show gtpc statistics" est la commande analogue pour les services SGSN/GGSN (GTPC v1) (CPC, Update PDP Context, etc.)
"show gtpu statistics" rapporte les statistiques sur le trafic du plan utilisateur traité par le ou les services GTP-U qui sont associés au ou aux services EGTPC. Un compteur intéressant est l'indication d'erreur Tx/Rx qui serait envoyée lorsque le noeud récepteur n'a pas d'enregistrement de l'abonné qui devrait être associé à l'identificateur de terminal (TEID) du paquet en question qui pourrait se produire pour un certain nombre de raisons (Le schéma/variable bulkstats est GTPU1/ err-ind-tx). Si le MME/SGW n'a pas été informé que le PGW n'a plus de liaison avec l'abonné, les appels peuvent devoir être effacés manuellement sur le MME/SGW (ou attendre qu'ils expirent) pour résoudre complètement un tel problème d'appels orphelins.

Les versions spécifiques aux homologues des commandes stat ci-dessus sont nécessaires lors du dépannage d'homologues spécifiques, ce qui est souvent le cas.

Détection des pannes de chemin

Le piège EGTPCPathFail est la clé pour savoir quand une défaillance de chemin s'est produite. Parmi les valeurs de base que l'on suppose déjà, il indique le nombre d'appels abandonnés, les anciens et nouveaux compteurs de redémarrage, et la raison de l'échec. En outre, la raison de déconnexion de session gtpc-path-failure sera incrémentée.
L'interruption EGTPCPathFailClear indique le compteur de redémarrage ainsi que la raison de l'effacement du chemin. La version la plus récente de StarOS v17 est mieux formulée pour dire « raison claire » au lieu de « raison d'échec » (il a effacé, pas échoué !) pour éviter toute confusion.
Les nombres de défaillances de chemins suivants sont rapportés par "show egtpc stat path-failure-reason"

```
[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
```

Remarques :

- Le compteur de redémarrage des messages de contrôle à sessmgr peut être de n'importe quel type de message de contrôle (tous ne sont pas répertoriés ici)
- Changement du compteur de redémarrage à partir du SGSN via la prochaine demande de contexte Create PDP après ce changement (cpc-restart-counter-change)
  - Changement du compteur de redémarrage à partir du SGSN via la requête de contexte PDP de mise à jour suivante après ce changement (qui serait un transfert à partir d'un autre SGSN) (upc-restart-counter-change)
  - Path clear failure reason sessions-add est un terme trompeur : il ne s'agit pas d'un échec, mais d'un succès, et il indique qu'une nouvelle session est arrivée pour établir la connexion
  - Si les valeurs signalées pour les compteurs de redémarrage et le nombre de sessions pour les déroutements PathFail et PathClear ne semblent pas s'aligner, ouvrez un ticket avec Cisco pour plus d'explications.

Il est important de comprendre comment la détection de défaillance de chemin EGTPC/GTPC et, dans une moindre mesure, EGTPU/GTPU (car la détection de défaillance du plan utilisateur n'est pas toujours configurée) est mise en oeuvre pour surveiller la connexion entre un GGSN ou un PGW et les divers SGSN et SGW homologues. Normalement, les connexions au sein d'un réseau de fournisseurs de services sont relativement stables, auquel cas les pannes de chemin sont peu fréquentes. D'un autre côté, l'expérience a montré que les chemins GTPC parmi les SGSN itinérants se connectant aux GGSN peuvent ne pas être aussi stables, et donc avec de mauvaises connexions, le résultat est des pannes de chemin qui se produisent assez fréquemment.
La configuration suivante couvre les minuteurs et la fréquence de transmission des messages de

contrôle et d’écho qui peuvent être utilisés pour détecter les pannes de chemin. Le même concept s'applique aux services GGSN, SGW, MME et EGTP :

service GGSN1

retransmission-timeout 20	Pour les messages de contrôle GTP (par rapport aux messages d'écho)
max-retransmissions 5	pour le contrôle GTP et les messages d'écho - combien de fois retransmettre avant le déclenchement de la détection de défaillance de chemin
echo-interval 120	Un écho est envoyé toutes les 2 minutes (120 secondes).
echo-retransmission-timeout 5	Pour Echos, si aucune réponse, il retransmet toutes les 5 secondes
path-failure detection-policy gtp echo	La détection des pannes de chemin est possible via l'écho gtp (au lieu de ne pas l'utiliser comme méthode de détection des pannes de chemin (tout en envoyant des échos)).
associate gtpu-service GGSN1_gtpu	Associe le service GTPU nommé GGSN1_gtpu à ce service GGSN
bind ipv4-address 209.165.201.2	Les adresses IP de liaison de l' utilisateur et du plan de contrôle sont identiques dans les exemples de configuration suivants, mais ne sont pas nécessairement configurées en tant que telles avec d'autres implémentations de fournisseur de services.

```
[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
```

Remarques :

- Le temps 0 étant donné lorsqu'un cycle de requête d'écho démarre (toutes les 2 minutes), s'il n'y a aucune réponse après un TOTAL de 30 secondes (6 tentatives), le chemin échoue (0 + 5 sec + (5 * 5 sec))
- Le processus egtpinmgr (gtpcmgr déconseillé depuis v14) exécuté sur la carte demux est responsable du traitement des paquets d'écho
- La section « show gtp statistics » de la section « Path Management Messages » contient des statistiques sur les requêtes/réponses d’écho pour le chemin de contrôle dans LES DEUX directions :

```
[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
```

Une autre méthode de détection de chemin est sur le plan utilisateur au lieu du plan de contrôle, et ceci est régi par la configuration suivante.

```
    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
```

Remarques :

- « no echo-interval » désactive l'envoi de requêtes d'écho pour le plan utilisateur et par conséquent, « path-failure detection-policy gtp echo » de l'interface de ligne de commande n'a aucune pertinence - NO Path failure detection is done on GTPU
- Cependant, GGSN répond toujours aux requêtes d'écho du SGSN
- « show gtpu stat » et « show demux-mgr stat gtpumgr » capturent les statistiques d'écho

```
[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
```

Divers exemples de pièges d'échec de chemin et explications

Généralement, le compteur de redémarrage de l'homologue augmente de un en cas de redémarrage. Dans ce cas, un bogue dans PGW l'a fait passer le compteur de redémarrage à la valeur du RC d'un autre homologue (premier trap), et donc quand le paquet suivant est arrivé sur la connexion, il l'a traité comme un redémarrage (second trap) en pensant que le RC avait changé. Le chemin s'efface alors sur tout paquet successif (troisième déroutement). Le but ici

n'est pas de connaître un bogue spécifique, mais plutôt d'essayer de connaître les changements de compteur qui ne peuvent pas être expliqués.

```
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 EGT
```

Cet exemple provient d'un GGSN où le SGSN homologue a changé son RC de 66 à 67 via une demande de contexte Create PDP :

```
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 GGS
```

Cet exemple provient d'un PGW où l'homologue ePDG a changé son RC de 69 à 70 via une demande de création de session :

```
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 EGT
```

Dans cet exemple, la sortie de différents noeuds PGW tous connectés au même SGW montre que le SGW a cessé d'être accessible à partir des PGW par un piège EGTPCPathFail au niveau des différents PGW dans un court laps de temps les uns des autres. Notez que tous affichent le même ancien compteur de redémarrage, comme nous l'avons vu précédemment, à savoir qu'un noeud utilise normalement le même compteur pour tous ses homologues. Dans le cas d'une absence de réponse, le nouveau compteur de redémarrage affiché est 9, mais le nouveau compteur pourrait mieux être imprimé comme non défini puisque la connexion est désactivée à partir du moment où le déroutement est déclenché et donc avoir un compteur de redémarrage n'a pas vraiment de sens.

```
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
```

L'exemple suivant montre une défaillance de chemin et le même nouveau compteur de redémarrage que dans l'exemple précédent, mais ensuite une demi-heure plus tard montre quand l'homologue se rétablit et que le compteur a augmenté de 15 à 16. Au fil du temps, le nombre augmente pour cet homologue :

<#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
|||||   |                               |    |   Current       Max
vvvvv   v              Peer Address     v    v   sessions    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)
```

Cet exemple montre une panne de chemin répétée due au redémarrage CPC pour 3 homologues SGSN spécifiques au GGSN. Le résultat ci-dessous montre un seul des homologues 209.165.200.225, mais ils ont tous le même problème. L'exécution répétée de « show egtpc peers » et « show gtpc stat sgsn-address » affiche également les valeurs alternatives. « show gtpc stat » ne confirme aucun problème de connectivité en ce qui concerne les requêtes et les réponses d'écho, qui dans ce cas sont uniquement initiées par le GGSN (acceptable comme discuté précédemment). Un point latéral à souligner ici est qu'il y a souvent plusieurs éléments de données qui peuvent être recueillis et qui corroborent tous la même conclusion - on peut choisir de recueillir les données qui sont les plus concluantes avec un minimum d'effort.

<#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 GGS
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 GGS
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 GGS
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 GGS
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 GGS
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 GGS

[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
```

En regardant les autres homologues dans le sous-réseau, ils montrent tous un RC = 132 tout le

temps, tandis que les homologues cassés alternent entre 129 et 132 comme indiqué ci-dessus, ce qui indique ces homologues SGSN comme passant potentiellement le mauvais RC dans les scénarios cassés :

<#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
```

Rappelez-vous qu'une autre approche des problèmes comme celui-ci consiste à vérifier d'autres GGSN (ou PGW pour LTE) pour voir s'ils ont le même problème avec ces homologues SGSN apparemment problématiques, et dans ce cas, ils le font également. Pour GGSN3, la première exécution de la commande a montré des valeurs de 132 pour les deux homologues, mais une autre exécution a montré des valeurs de 129 pour l'un des homologues. Plusieurs passages (non représentés ici pour des raisons de concision) montreraient tous les GGSN rebondissant entre les deux RC pour ces homologues au fil du temps :

```
[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
```

Enfin, pour confirmer le problème, le SGSN modifie le RC. L'option de menu de surveillance de l'abonné par l'homologue SGSN affiche les RC qui changent. En cas de modification, tous les appels sont déconnectés avec path-failure comme motif de déconnexion. Une capture d'écran d'une capture de paquets montre également ceci :

<#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

Cet exemple montre une défaillance de chemin où le problème se trouvait en fait dans le chemin et non dans le PGW ou le SGW. Par conséquent, le RC reste le même lorsque le chemin est rétabli. Dans le deuxième exemple homologue, la commande clear est le résultat de l'établissement de la requête d'écho :

```
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 EGT

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 EGT
```

Cet exemple montre une autre anomalie où, à plusieurs reprises après que l'ancien compteur de redémarrage a été réinitialisé à 2, il se retrouve en quelque sorte à 0 lorsque le chemin échoue. Cela s'est avéré être un bug. Encore une fois, la leçon ici est que si quelque chose semble étrange, il faut le signaler.

```
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
```

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.