

Contenido

[Introducción](#)

[Antecedente](#)

[Comandos relevant](#)

[Detección de falla del trayecto](#)

[Diversos ejemplos de los desvíos y de las explicaciones del fall de la trayectoria](#)

[Discusiones relacionadas de la comunidad del soporte de Cisco](#)

Introducción

Este artículo cubre entendiendo y resolviendo problemas desarrollado General Packet Radio Service Tunneling Protocol para control avión (/plano del usuario del protocolo de túneles GTPC (especificación técnica [29.274](#) EGTPC del proyecto de la sociedad de la 3ra generación), GPRS (GTPv1-U) (3GPP TS [29.281](#)), y la más relevante a este artículo, problemas de los procedimientos de restauración ([3GPP TS 23.007](#)), que giran alrededor de los conceptos tales como los contadores del reinicio, la detección de fallas del trayecto, y los pedidos de eco/las respuestas. Incluye una lista de comandos relevant y qué a buscar en la salida, junto con la explicación de los configurables relacionados. Ésta es ciertamente un área para la cual se plantean muchas preguntas. Observe que resolver problemas los problemas específicos del Control de llamadas es en sí mismo una área de interés muy extensa, solamente las menciones de este artículo solamente cómo capturar las estadísticas para el Control de llamadas - cualquier cosa más allá de ese (es decir los problemas de la configuración de la llamada) no se cubre. Este artículo se centra realmente principalmente en el troubleshooting y verificar las conexiones de peer, sin las cuales no habría Control de llamadas. Mientras que estaban escritos para los operadores, los diseñadores/los ejecutores podrían también beneficiarse del contenido aquí. Ejemplos todos los IP Addresses de la falsificación del uso (cambiada) de la red de un operador real.

Antecedente

Cada lado de una conexión GTP (una de S5 (**gateway de la porción (SGW) - gateway de la red de los datos del paquete (PGW)**), S11 (**entidad de administración de las multimedias (MME) - SGW**), S4 (**nodo de servicio del servicio GPRS (SGSN) - nodo del servicio del gateway GPRS (GGSN)**)) mantiene un contador del reinicio que se diseñe para alertar al par si el nodo de origen ha recommenzado de modo que ese par pueda tomar la acción apropiada. Mientras que un nodo puede utilizar un contador único para cada par que habla con, el diseño tiende a ser utilizar el MISMO contador para todos los pares, puesto que cuando el nodo recommienza, el contador necesita ser incrementado para todos los pares, y así que el seguimiento de un valor separado sobre una base del par no sirve ninguna ventaja adicional puesto que ellos toda la necesidad de aumentar en una de todos modos. De hecho usando lo mismo recommienza al revés para todos los pares es útil en resolver problemas una Red grande en donde todos los pares pueden ser marcados para asegurarse que los contadores del reinicio recibidos tienen el mismo (solo) valor para el nodo que esos pares están hablando con.

El contador del reinicio SE PUEDE incluir en todos los tipos de mensajes del control, mientras que el contador del reinicio SE DEBE incluir en todos los pedidos de eco y respuestas. Tan pronto como un cambio del contador del reinicio ocurra en un nodo, informan a los diversos pares sobre él vía los mensajes siguientes del control enviados a los pares respectivos. Cuando un reinicio ocurre, por la especificación, el lado de recepción tiene la opción temporalmente de mantener las llamadas que tiene para un cortocircuito mientras que o en la caída de ellas, este último que es

típicamente cuál es el comportamiento configurado.

En el caso de usar los pedidos de eco, el método es enviar los mensajes en un intervalo configurado, y si no se recibe ninguna respuesta después de que un número configurado de comprobaciones (cada uno después de un descanso configurado), la trayectoria se considera para estar abajo y la acción apropiada tomada. La configuración permite para que el eco sea enviado opcionalmente, y ésta se aplicaría al par también, cuyo acercamiento no tiene que necesariamente ser lo mismo, especialmente cuando el par podría ser de un diverso vendedor y/o de un diverso portador. De hecho en lo que respecta a las respuestas de la generación de eco, un nodo se obliga solamente para responder a un pedido de eco - si recibe el eco o no y con qué frecuencia no es significativa - apenas necesita enviar una respuesta de la generación de eco para cada petición, y tomar medidas si el contador del reinicio recibió los cambios. Por supuesto según lo mencionado ya, Si también está enviando los pedidos de eco y no recibe las respuestas oportunas, necesita tomar medidas para ese escenario también.

Observe que la detección de falla del trayecto se puede también hacer sobre la conexión GTP-U (plano del usuario) vía la generación de eco, pero SOLAMENTE vía el descanso, puesto que no hay contador del reinicio incluido en los mensajes GTP-U como hay para GTP-C como discutido previamente. GTP-U puede o no puede utilizar la misma trayectoria que por los mensajes del control. Si es la misma trayectoria, después la detección GTP-U puede no estar como objeto de valor puesto que la detección GTP-C debe ser suficiente.

En resumen, aquí están las maneras que la detección de falla del trayecto puede ser hecha:

- Cambio del contador del reinicio (vía el req/Rsp de los msg del control o la generación de eco)
- Tiempo del req de la generación de eco GTPC hacia fuera
- Tiempo del req de la generación de eco GTPU hacia fuera
- Controle el descanso del mensaje del req (no configurado generalmente)

Comandos relevant

Para resolver problemas cualquier clase de problemas de la trayectoria, familiarice con los siguientes comandos. Lo que sigue es una explicación de los detalles relevantes de cada uno.

- **muestre [egtpc-servicio | gtpu-servicio] todo**
- **muestre a pares del egtpc el [egtp-service <service>]**
- **gtp-versión {1 de la generación de eco de la prueba del egtpc | address> del <peer de la dirección de peer del address> del <service del src-direccionamiento 2}**
- **muestre las estadísticas del egtpc [[[sgw-direccionamiento | PGW-direccionamiento | [verbose] del [debug-only] del [demux-only] del <address> del mme-direccionamiento)]] | trayectoria-error-razones]**
- **muestre el [peer-address <peer>] de las estadísticas del gtpu**
- **muestre el [smgr-instance <instance>] del [ggsn-service <service>] de las estadísticas del gtpc**
- **muestre el <egtpinmgr de las estadísticas del demux-mgr | egtpegmgr | gtpumgr> todo (SGW-salida | GGSN/PGW, ingreso /SGW | GTPU)**
- **[Clear] de EGTPCPathFail, SNMP traps del [Clear] de EGTPUPathFail**
- **muestre los motivos de desconexión de la sesión - gtpc-trayectoria-error, gtpu-trayectoria-error, falla del trayecto**

Otros comandos para contar a los suscriptores se relacionaron con GTP:

muestre el <service> sub del GGSN-servicio del [summary]

muestre el <service> sub del gtpu-servicio del [summary]

muestre el [egtp-service <service>] de las sesiones del egtpc

El comando básico de marcar el estatus y la configuración de todos los servicios EGTP es **“egtpc-servicio de la demostración”**. La mayor parte de la salida debe ser qué se sabe ya vía la configuración, así como el estatus debe SER COMENZADO. La información única espigada es el “reinicio contrario” que no se puede encontrar en cualquier parte. El Comando relacionado **“gtpu-servicio de la demostración todo el”** vuelve los configurables para los servicios GTPU que se asocian a los servicios EGTPC.

“los pares del egtpc de la demostración” son muy útiles en la identificación de todos los pares, incluyendo los estatuses, reinicio contradicen, y el suscriptor actual/máximo cuenta.

- Observe que este comando cubre a los pares SGW y SGSN, mientras que en el pasado había un comando separado para SGSN/GGSN.
- Puesto que este comando incluye todos los tipos de peer, el tipo de peer puede ser filtrado encendido vía el nombre del servicio (aunque el egtpc-servicio de la palabra clave también toma engañoso un nombre del servicio GGSN).
- Si no hay suscriptores con el par, después el estatus es generación de eco inactiva y GTPC inhabilitada. En ese estado, para EGTPC PUEDE haber algunos pares para quienes el contador del reinicio está señalado, mientras que la mayoría puede ser 0. Esto ocurre para las situaciones donde el par está enviando activamente los pedidos de eco a los cuales ASR5K debe responder. El contador del reinicio se puede determinar funcionando con el comando test EGTPC (discutido después), aunque en el estado inactivo, el contador no será puesto al día en “los pares del egtpc de la demostración” aunque fue recibido. Mientras tanto para el GGSN, el contador del reinicio nunca se olvida.

Observe a los pares de un par debajo de ése tienen cuentas mucho más arriba que los otros pares. Éstos están para SGWs que están situadas físicamente en el mismo sitio o el sitio de la hermana del PGW al cual el MME concentra la mayoría de las llamadas.

Observe eso que funciona con este comando en una Redundancia del servicio del Inter-chasis (ICSR) que el nodo espera es solamente relevante para los peeres activos, en este caso los números están cercanos, por ejemplo en los pares siguientes:

Pares... 20:51:52 del egtpc de la demostración
PGW> (ACTIVE)

Pares... 20:51:51 del egtpc de la demostración
PGW-ICSR> (RECURSO SEGURO)

	Par	RC	# sesiones	Sess máximo	Par	RC	# sesiones	Sess máximo	
AESKS	192.0.2.5		2	31	443 ADSKS	192.0.2.5	2	31	443
AESKS	192.0.2.2		3	117	123 ADNKS	192.0.2.2	3	117	123
	4				4				
AESKS	192.0.2.3		15	4223	6889 ADSKS	192.0.2.3	15	4223	6889
	6				6				
AESKS	192.0.2.4		7	119	331 ADSKS	192.0.2.4	7	119	331
	4				4				
AESKS	192.0.2.6		34	6174	9798 ADSKS	192.0.2.6	34	6171	9797
	8				8				
AESKS	192.0.2.6		7	2311	5027 ADSKS	192.0.2.6	7	2311	5027
	9				9				
AESKS	192.0.2.8		17	3914	8880 ADSKS	192.0.2.8	17	3913	8874
	8				8				
AESKS	192.0.2.1		16	833705	1298070 ADSKS	192.0.2.1	16	832037	1294412
	64				64				
AESKS	192.0.2.1		3	1438418	2153317 ADSKS	192.0.2.1	3	1435461	2147280
	84				84				
AESKS	192.0.2.1		21	11447	26577 ADSKS	192.0.2.1	21	11443	26568
	93				93				
AESKS	192.0.2.1		5	48	162 ADSKS	192.0.2.1	5	48	162

	95				95				
AESKS	192.0.2.2	21	1962	3931	ADSKS	192.0.2.2	21	1962	3931
	26					26			
AESKS	192.0.2.2	6	4465	9332	ADSKS	192.0.2.2	6	4466	9332
	27					27			

“la generación de eco de la prueba del egtpc” se utiliza para marcar a un par específico para considerar si es accesible o no y se debe ejecutar en el contexto en donde se define el servicio.

- Utilice la gtp-versión 1 para las conexiones GGSN-SGSN
- Usando el comando ping no es una prueba válida, aunque si es acertado, se sabe que hay un cierto nivel de accesibilidad.
- Se visualiza el contador del reinicio del par (recuperación).
- Funcionar con este comando incrementará los contadores de la respuesta del pedido de eco del tx/de la generación de eco del rx en “las estadísticas del egtpc de la demostración”
- El contador del reinicio del par está señalado después de la palabra clave de la recuperación (11 en el ejemplo abajo)
- Advertencia: aunque esto no debe suceder en las Circunstancias normales, si una generación de eco de la prueba se realiza en un par y mide el tiempo hacia fuera por la razón que sea (si toda entonces lo trabaja mide el tiempo hacia fuera), todas las llamadas e caída (aunque las cosas trabajaban apenas muy bien hasta la punta de hacer la prueba).

“las estadísticas del egtpc de la demostración” están para EGTPC (v2) y los informes sobre el éxito/los errores de muchos mensajes del control EGTPC (petición/respuesta, Tx/Rx) incluyendo crean la sesión, modifican el portador, sesión de la cancelación, los datos del link descendente notifican, liberan el portador del acceso, crean el portador, ponen al día el portador, modifican el portador, suspenden, etc., y las respuestas asociadas. Incluye todos los contadores de la Administración de la trayectoria para el Tx/Rx del pedido de eco/de la respuesta, y basado en los temporizadores, el crecimiento de estos temporizadores se puede predecir y utilizar para resolver problemas si no incrementar como se esperaba.

El echos y las respuestas de las demostraciones del siguiente ejemplo que eran iniciados y respondieron de los ambos lados. Observe que el SGW se ha configurado para enviar una vez por el minuto mientras que el PGW se ha configurado por una vez cada tres minutos (los ambos lados no necesitan tener la misma configuración) y así que cada tres minutos allí son dos veces el número de los paquetes (6:16:33, 6:19:33). Tenga presente que aunque los ambos lados estén en el mismo “ciclo” en este ejemplo, ellos no tiene que ser y puede funcionar con a la independiente de uno a. Los contadores del reinicio se han capturado en una columna también. Observe el contador del reinicio de 11 de la respuesta de la generación de eco SGW hace juego el valor de su respuesta a la generación de eco de la prueba PGW desde arriba (la prueba fue hecha en el momento diferente que la captura de paquetes pero para el mismo peer de entidades pares).

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

Demux-only la opción captura todos los contadores de la Administración de la trayectoria (pedidos de eco/respuestas) así como los mensajes del control que nunca alcanzan un sessmgr, que es quien crea generalmente el pedido de sesión/la respuesta. Demux procesa el etpinmgr que reside en el Demux que el indicador luminoso LED amarillo de la placa muestra gravedad menor es responsable de manejar todo el pedido de eco/respuesta y de la dirección inicial de todos los mensajes de control de llamadas antes de distribuirlos a los sessmgrs apropiados (aleatoriamente para las nuevas llamadas, y específicamente para las llamadas existentes aseguradas en un sessmgr específico). **Sessmgr-only** la opción cuenta inversamente solamente los mensajes que consiguen dirigidos por los sessmgrs, que no incluye los contadores de la Administración de la trayectoria. No especificar a cualquier calificador es el valor por defecto de contar todos los valores.

“las estadísticas del gtpc de la demostración” son el comando análogo para los servicios SGSN/GGSN (v1) GTPC (CPC, contexto de la actualización PDP, los etc.)

“muestre stats de los informes de las estadísticas del gtpu” en el tráfico del plano del usuario manejado por los servicios GTP-U que se asocian a los servicios EGTPC. Un contador interesante es el Tx/Rx de la indicación de error que serían enviados cuando el nodo de recepción no tiene ningún expediente del suscriptor que debe ser asociado al identificador de punto final de terminal (TEID) del paquete en la pregunta que podría suceder por varias razones (el esquema/la variable de los bulkstats es `errar-IND-tx GTPU1/`). Si el MME/SGW no se ha informado que el PGW tiene no más un atascamiento al suscriptor, después las llamadas pueden necesitar ser borrado manualmente en el MME/SGW (o esperarlo al descanso) para resolver completamente tal aplicación las llamadas dejadas huérfano.

Las versiones específicas del par de los comandos stat antedichos son necesarias al resolver problemas a los pares específicos, que es a menudo el caso.

Detección de falla del trayecto

El desvío de **EGTPCPathFail** es la clave para saber cuando ha ocurrido una falla del trayecto. Entre algunos valores básicos que uno asumiría ya, señala el número de llamadas caídas, de los viejos y nuevos contadores del reinicio, y la razón del error. Además, incrementarán al `gtpc-trayectoria- error del` motivo de desconexión de la sesión.

El desvío de **EGTPCPathFailClear** indicará el contador del reinicio junto con la razón del claro de la trayectoria. El StarOS más reciente v17 se redacta mejor para decir la “razón clara” en vez de la “razón del error” (borró, no falló!) para evitar la confusión.

Los siguientes son averiados las cuentas de la falla del trayecto que están señaladas **“de las trayectoria-error-razones stat del gtpc de la demostración”**

Notas:

- El contador de mensaje de control del reinicio en el sessmgr podría ser cualquier tipo de mensaje de control (enumerado no no todo aquí)
- El contador cambiado del reinicio del SGSN vía el siguientes crea la petición del contexto PDP después que cambian (**el CPC-reinicio-contador-**
- Contador cambiado del reinicio de los SGSN vía la petición siguiente del contexto de la actualización PDP después que cambian (que sería manos de otro SGSN) (**UPC-reinicio-contador-cambio**)
- El error claro de la trayectoria que la razón sesión-agrega es un término de engaño - es un no error sino un éxito, e indica que una nueva sesión ha venido adentro establecer la conexión
- Si los valores señalados para el reinicio contradicen y las cuentas de sesiones para los desvíos de PathFail y de PathClear no parecen alinearse, abra un boleto con Cisco para la explicación adicional.

Es importante entender cómo EGTPC/GTPC y a una detección de falla del trayecto del grado menor EGTPU/GTPU (pues la detección de falla del plano del usuario no se configura siempre)

se implementa para monitorear la conexión entre un GGSN o un PGW y al diverso par SGSN y SGWs. Las conexiones dentro de una red de los proveedores de servicio son normalmente bastante estables, en este caso las fallas del trayecto son infrecuentes. En el aspecto negativo, la experiencia ha mostrado que las trayectorias GTPC entre la itinerancia de SGSNs que conecta con los GGSN pueden no ser tan estables, y así que con las malas conexiones el resultado es averiado fallas del trayecto que suceden en una base bastante frecuente.

El config siguiente cubre los temporizadores y la frecuencia de la transmisión para el control y los mensajes de eco que se pueden utilizar para detectar las fallas del trayecto. El mismo concepto se aplica a los servicios GGSN, SGW, del MME, y EGTP:

GGSN-servicio GGSN1	
retransmisión-descanso 20	Para los mensajes del control GTP (contra los mensajes de eco)
MAX-retransmisiones 5	¿para el control y los mensajes de eco GTP? cuántas veces de retransmitir antes de la detección de falla del trayecto accione
generación de eco-intervalo 120	Una generación de eco se envía cada 2 minutos (120 segundos).
generación de eco-retransmisión-descanso 5	Para el echos, si ninguna respuesta, él retransmite cada 5 segundos
generación de eco del gtp de la detección-directiva de la falla del trayecto	La detección de falla del trayecto es posible vía la generación de eco del gtp (pero no usando esto como método de detectar la falla del trayecto (pero todavía de el echos)).
gtpu-servicio GGSN1_gtpu del socio	Asocia el servicio Nombrado GGSN1_gtpu GTPU a este servicio GGSN
lazo ipv4-address 209.165.201.2	Los IP Addresses del lazo del <u>usuario y del avión del control</u> son <u>lo mismo</u> en e configs del siguiente ejemplo, pero no se pueden configurar necesariamente con tal con las implementaciones del proveedor del otro servicio.

Notas:

- Cualquier momento 0 que es cuando un ciclo del pedido de eco comienza (cada 2 minutos), si no hay respuestas después de que un total de 30 segundos (6 tentativas), la trayectoria falle (0 + 5sec + (5 * 5sec))
- El egtpinmgr de proceso (gtpcmgr desaprobado puesto que v14) que se ejecuta en el indicador luminoso LED amarillo de la placa muestra gravedad menor del demux es responsable de procesar los paquetes de eco
- ¿? ¿muestre las estadísticas del gtp? ¿, sección? ¿Mensajes de administración de la trayectoria? tiene estadísticas para los pedidos de eco/las respuestas para el trayecto de control en las ambas direcciones:

Otro método de detección de la trayectoria está en el plano del usuario en vez del avión del control, y esto es gobernada por los config siguientes.

Notas:

- ¿? ¿ningún generación de eco-intervalo? ¿inhabilita el envío de los pedidos de eco para el plano del usuario y por lo tanto el CLI? ¿generación de eco del gtp de la detección-directiva de la falla del trayecto? ¿no tiene ninguna importancia? No se hace NINGUNA detección de falla del trayecto en GTPU
- El GGSN todavía responde a los pedidos de eco del SGSN sin embargo
- ¿? ¿muestre el stat del gtpu? ¿y? ¿muestre el gtpumgr stat del demux-mgr? capture el stats

Diversos ejemplos de los desvíos y de las explicaciones del fall de la trayectoria

El contador del reinicio del par aumentará generalmente en uno cuando hay un reinicio. En este caso, un bug en el PGW lo hizo reemplazar el reinicio en dirección contraria el valor del RC de otro par (primer desvío), y tan cuando el próximo paquete llegó sobre la conexión, la trató como reinicio (en segundo lugar desvío) que pensaba que el RC había cambiado. La trayectoria entonces borra en cualquier paquete sucesivo (tercer desvío). La punta aquí no es saber sobre un bug específico pero bastante intentar y ser consciente de ninguna cambios contraria que no puedan ser explicados.

Este ejemplo es de un GGSN donde el par SGSN ha cambiado su RC a partir del 66 a 67 vía una petición del contexto del crear PDP:

Este ejemplo es de un PGW donde el ePDG del par ha cambiado su RC a partir del 69 a 70 vía un pedido de sesión del crear:

En este ejemplo, la salida de los diversos Nodos PGW conectados todo con la misma demostración SGW que el SGW paró el llegar a ser accesible de los PGW por un desvío de EGTPCPathFail en los diversos PGW dentro de un marco de breve periodo de tiempo de uno a. Observe que todos muestran el mismo viejo contador del reinicio que fue discutido anterior que normalmente un nodo utiliza el mismo contador para todos sus pares. En el caso de ninguna respuesta, el nuevo contador del reinicio visualizado es 9, pero el nuevo contador se pudo imprimir mejor como indefinido puesto que la conexión está abajo a partir de cuando se acciona el desvío y así que tener un contador del reinicio no tiene realmente sentido.

El siguiente ejemplo muestra una falla del trayecto y el mismo nuevo contador del reinicio que en el ejemplo anterior, pero por otra parte demostraciones de una media hora posteriores cuando viene el par salvaguardia otra vez y el contador ha aumentado en el una a partir 15 a 16. En un cierto plazo los aumentos de la cuenta para ese par:

```
Wed Jul 22 05:07:08 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service EGTP1, interface type pgw-ingress, self address 209.165.201.13, peer address 198.51.100.162, peer old restart counter 15, peer new restart counter 15, peer session count 34, failure reason no-response-from-peer
```

```
Wed Jul 22 05:38:00 2015 Internal trap notification 1113 (EGTPCPathFailClear) context XGWin, service EGTP1, interface type pgw-ingress, self address 209.165.201.13, peer address 198.51.100.162, peer restart counter 16, peer session count 1, clear reason sessions-update
```

```
[local]PGW> show egtpc peers
Wednesday July 22 10:04:30 UTC 2015
```

Service ID	Peer Address	Restart Counter	No. of restarts	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)

Este ejemplo muestra una falla del trayecto que relanza debido al **reinicio CPC** para 3 pares específicos SGSN al GGSN. La salida debajo de las demostraciones apenas una de los pares 209.165.200.225, pero ellos toda tiene el mismo problema. El funcionamiento relanzado “de los pares del egtpc de la demostración” y “del sgsn-direccionamiento stat del gtpc de la demostración” también muestra los valores de alternancia. “el stat del gtpc de la demostración” no confirma ningún problema con la Conectividad en cuanto a los pedidos de eco y a las respuestas, que en este caso están siendo iniciados solamente por el GGSN (aceptable según lo discutido anterior). Una punta de la barra lateral que se hará aquí es que hay a menudo pedazos múltiples de datos que puedan ser recogidos que todos corroboran a la misma conclusión - una puede elegir recoger los datos que son los más concluyentes con la cantidad mínima de esfuerzo.

```
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, interface type ggsn, self address 209.165.202.129, peer address 209.165.200.225, peer
old restart counter 129, peer new restart counter 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, interface type ggsn, self address 209.165.202.129, peer address 209.165.200.225,
peer restart counter 132, peer session count 4, clear reason sessions-add
Mon Jun 22 22:20:55 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service
GGSN1, interface type ggsn, self address 209.165.202.129, peer address 209.165.200.225, peer
old restart counter 132, peer new restart counter 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, interface type ggsn, self address 209.165.202.129, peer address 209.165.200.225,
peer restart counter 129, peer session count 4, clear reason sessions-add
Mon Jun 22 22:20:58 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service
GGSN1, interface type ggsn, self address 209.165.202.129, peer address 209.165.200.225, peer
old restart counter 129, peer new restart counter 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, interface type ggsn, self address 209.165.202.129, peer address 209.165.200.225,
peer restart counter 132, peer session count 4, clear reason sessions-add
Mon Jun 22 22:21:02 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service
GGSN1, interface type ggsn, self address 209.165.202.129, peer address 209.165.200.225, peer
old restart counter 132, peer new restart counter 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, interface type ggsn, self address 209.165.202.129, peer address 209.165.200.225,
peer restart counter 129, peer session count 3, clear reason sessions-add
Mon Jun 22 22:21:03 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service
GGSN1, interface type ggsn, self address 209.165.202.129, peer address 209.165.200.225, peer
old restart counter 129, peer new restart counter 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, interface type ggsn, self address 209.165.202.129, peer address 209.165.200.225,
peer restart counter 132, peer session count 5, clear reason sessions-add
Mon Jun 22 22:21:06 2015 Internal trap notification 1112 (EGTPCPathFail) context XGWin, service
GGSN1, interface type ggsn, self address 209.165.202.129, peer address 209.165.200.225, peer
old restart counter 132, peer new restart counter 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, interface type ggsn, self address 209.165.202.129, peer address 209.165.200.225,
peer restart counter 129, peer session count 3, clear reason sessions-add

```

```
[XGWin]GGSN> show egtpc peers | grep 209.165.200.225
```

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

| | |
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

Mirando a otros pares en la subred, todos muestran un RC = 132 todo el tiempo, mientras que los pares quebrados alternan entre el 129 y 132 según lo apenas mostrado, que señala hacia estos pares SGSN como potencialmente paso del RC incorrecto en los escenarios quebrados:

Restart

Counter

AESKG 6 209.165.200.225 129 85115 8 57 <==IDNKG 6 209.165.200.226 132 21 0
3AESKG 6 209.165.200.227 132 60 47 92AESKG 6 209.165.200.228 132 53 24 59AESKG 6 209.165.200.229
132 69 92 126AESKG 6 209.165.200.232 132 66 21 30AESKG 6 209.165.200.236 132 43 8 21AESKG 6
209.165.200.238 132 49 18 32AESKG 6 209.165.200.239 132 39 1 6AENKG 6 209.165.200.240 132 17 1
5AESKG 6 209.165.200.241 132 67 6 43AESKG 6 209.165.200.242 132 56 22 29AENKG 6 209.165.200.243
132 40 2 9AENKG 6 209.165.200.246 129 17713 3 24 <==AESKG 6 209.165.200.247
132 64 29 45AESKG 6 209.165.200.250 132 62 42 56

Memoria de anterior, otro acercamiento a los problemas como esto es marcar otros GGSN (o los PGW para el LTE) para considerar si tienen el mismo problema con esos pares aparentemente problemáticos SGSN, y en este caso hacen también. Para GGSN3 la primera ejecución del comando mostró los valores 132 para ambos pares pero otro funcionamiento mostró el 129 para uno de los pares. Los funcionamientos múltiples (no mostrados aquí para la brevedad) mostrarían todos los GGSN que despiden entre el RCs dos para esos pares en un cierto plazo:

Restart

Counter

AESKG 6 209.165.200.225 129 85115 8 57 <==IDNKG 6 209.165.200.226 132 21 0
3AESKG 6 209.165.200.227 132 60 47 92AESKG 6 209.165.200.228 132 53 24 59AESKG 6 209.165.200.229
132 69 92 126AESKG 6 209.165.200.232 132 66 21 30AESKG 6 209.165.200.236 132 43 8 21AESKG 6
209.165.200.238 132 49 18 32AESKG 6 209.165.200.239 132 39 1 6AENKG 6 209.165.200.240 132 17 1

```

5AESKG 6 209.165.200.241 132 67 6 43AESKG 6 209.165.200.242 132 56 22 29AENKG 6 209.165.200.243
132 40 2 9AENKG 6 209.165.200.246 129 17713 3 24 <==AESKG 6 209.165.200.247
132 64 29 45AESKG 6 209.165.200.250 132 62 42 56

```

Finalmente confirmar el problema es el SGSN que cambia el RC, opción de menú del suscriptor del monitor **por las demostraciones del par SGSN que cambian el RCs**. Cuando cambia, todas las llamadas serían desconectadas con la **falla del trayecto** como el motivo de desconexión. Un tiro de pantalla de una captura de paquetes también muestra esto:

```

[XGWin]GGSN> mon subMonday 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-failureINBOUND>>>> 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 2015INBOUND>>>> 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

Este ejemplo muestra a falla del trayecto donde estaba el problema realmente en la trayectoria y no el PGW o el SGW. Como consecuencia, el RC sigue siendo lo mismo cuando la trayectoria restablece. En el segundo ejemplo del par el claro es un resultado del establecimiento del pedido

de eco:

```
[XGWin]GGSN> mon subMonday 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-failureINBOUND>>>> 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 2015INBOUND>>>> 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
```

Este ejemplo muestra a otra anomalía donde en varias ocasiones después de que el viejo contador del reinicio haya reajustado a 2, termina de alguna manera para arriba en 0 otra vez cuando el fall de la trayectoria borra. Éste resultó ser un bug. La lección aquí es otra vez si algo aparece impar, él debe ser señalada.

```
[XGWin]GGSN> mon subMonday 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-failureINBOUND>>>> 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 2015INBOUND>>>> 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
```