osmo-sip-connector - Bug #4310

sends BYE to self

12/05/2019 11:15 AM - neels

Status:	New	**Start date:**	12/05/2019
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:			
Target version:			
Resolution:		**Spec Reference:**	

Description

During analysis of a pcap of a short call, I noticed that the final SIP ACK as well as BYE goes back to its own address.
This is using osmo-sip-connector and Kamailio

This is a text representation of the entire SIP negotiation,
the point being the last three 'SIP->(self)' in the end.

(MO -> sipcon -> kamailio -> sipcon -> MT)

```
        SIP2    sip.***New:SIP2=192.168.178.74:5069 452
SIP             sip.***New:SIP=192.168.178.74:5060 452
SIP--->SIP2     sip.INVITE udp{'src'='192.168.178.74:5060', 'dst'='192.168.178.74:5069'} 452
SIP<---SIP2     sip.INVITE-trying udp{'src'='192.168.178.74:5069', 'dst'='192.168.178.74:5060'} 46
0
SIP<---SIP2     sip.INVITE udp{'src'='192.168.178.74:5069', 'dst'='192.168.178.74:5060'} 462
SIP--->SIP2     sip.INVITE-Trying udp{'src'='192.168.178.74:5060', 'dst'='192.168.178.74:5069'} 46
9
SIP--->SIP2     sip.INVITE-Ringing udp{'src'='192.168.178.74:5060', 'dst'='192.168.178.74:5069'} 1
200
SIP<---SIP2     sip.INVITE-Ringing udp{'src'='192.168.178.74:5069', 'dst'='192.168.178.74:5060'} 1
202
SIP--->SIP2     sip.INVITE-OK udp{'src'='192.168.178.74:5060', 'dst'='192.168.178.74:5069'} 1324
SIP<---SIP2     sip.INVITE-OK udp{'src'='192.168.178.74:5069', 'dst'='192.168.178.74:5060'} 1328
SIP->(self)     sip.ACK udp{'src'='192.168.178.74:5060', 'dst'='192.168.178.74:5060'} 1353
SIP->(self)     sip.BYE udp{'src'='192.168.178.74:5060', 'dst'='192.168.178.74:5060'} 1702
SIP->(self)     sip.BYE-OK udp{'src'='192.168.178.74:5060', 'dst'='192.168.178.74:5060'} 1706
```

(the final numbers are the packet index in the attached pcap;
output generated by 'osmo-gsm-shark -f short_call_flt.pcapng --filter-msg sip --show-traits udp')

I'm not entirely sure whether that is intended, i.e. that Kamailio connects both peers and takes itself out of the loop (like it does with RTP),
and I can in fact find a 'Contact URI' in the SIP package that might intend to do that, but it has no port number?
Or is it the 'Via' header ... ?

Figure this out to see whether osmo-sip-connector has a bug.

History

#1 - 01/09/2020 09:21 AM - keith

If Kamailio (or rather rtpengine) is not involved in the RTP, then there is no reason for it to be involved in the BYE. Regardless of that, I'm 99% sure that this has to do with your kamailio setup, which is not inserting the headers that would require the SIP UA to route the ACK and BYE via the proxy. I don't know if that's because of an omission in the config, or if with a default config kamailio is clever enough to figure out that SIP can reach SIP2 and it does not need to be involved in the ACK/BYE.

I've never seen this, but then my kamailio is multi-homed and SIP could never route directly to SIP2 (so in my case in fact this would break the BYE)

If rtpengine were involved, then you **would** want the proxy to see the BYE, as kamailio would instruct rtpengine to tear down the streams as a result of the BYE. (rtpengine will eventually detect lack of RTP and do so anyway, so no fatal harm done.)

BTW, did you know that kamailio is (mostly) stateless? - certain modules aside, it's just forwarding messages on a one by one basis without maintaining any concept of "call"

From a similar scenario; not a complete explanation, but does the accepted answer help?

EDIT: actually - I just noticed the link to this presentation on SIP routing: https://vimeo.com/140267478 *in the answer. I watched the first few minutes and it looks like a good 30 mins spent to not wonder anymore about Contact and Via and Route headers.*

https://stackoverflow.com/questions/57630610/how-do-i-ensure-that-bye-messages-bypass-a-sip-proxy

I think this is certainly not a bug in osmo-sip-connector.

Dialog stuff is being handled by sofia-sip defaults anyway.
osmo-sip-connector is not explicitly doing any SIP header insertion.

Files

short_call_flt.pcapng	586 KB	12/05/2019	neels