OsmoMSC - Bug #2348

AoIP: garbled RTP in call following a call to an unreachable subscriber

07/03/2017 08:14 PM - dexter

Status:	Closed	**Start date:**	07/03/2017
Priority:	Normal	**Due date:**	
Assignee:	dexter	**% Done:**	100%
Category:			
Target version:			
Resolution:		**Spec Reference:**	

Description

There is a problem somewhere in the call control, probably also connected to MGCP. The problem succours very rarely, but it can be provoked by doing the following:

1) Make a call to an unreachable subscriber

2) Make a call to a reachable subscriber

The other phone will ring. When picked up on the calling phone playing a metallic noise sound (garbeled RTP traffic).

The problem needs to be investigated further. It might be a race condition between the connection that is still open from the failed call and the new established connection.

In a_iface.c, see a_iface_tx_dtap(), the link_id is permanently set to 0x00. Looks like a candidate for the cause of the problem.

History

#1 - 07/04/2017 11:55 AM - neels

Please clarify: call an unreachable subscriber; when I dial an unsubscribed number, all I get is "invalid number" and the call aborts. Is that what you mean? Or do you mean the line has to be busy?

#2 - 07/05/2017 03:53 PM - dexter

- File capture.pcapng added

- Status changed from New to In Progress

Attached a pcap file with one successful call and after that follows a broken call.

#3 - 07/05/2017 03:57 PM - neels

- Subject changed from AoIP: Race condition problem to AoIP: garbled RTP in call following a call to an unreachable subscriber

#4 - 07/05/2017 04:36 PM - neels

It would help if you could explain the actions taken at specific points in the pcap (by packet number). Does it include the initial unsuccessful call? Was the **entire** set of core network programs restarted just before the pcap was taken?

Also interesting would be logs of the MSC (MM, PAG, RSL) as well as the MGCPGW log.

#5 - 07/05/2017 04:47 PM - dexter

The trace contains one successful call. The bug was triggered to make one call to 23101, which is the non reachable end. Immediately after that follows a call to 23006, which is the reachable end.

I also made the following observations so far:

With the following capture filter one can see what happens here: gsm_abis_rsl.msg_dsc == 63 || mgcp

The bug starts at packet 3957. We can see the interaction between the MSC and the MGCP-GW, followed by the IPACC interaction from between the

BSC and BTS. So far this looks normal. The following connection is negotiated:

BTS-Ports: RX 59686, TX 4002

Since the call fails there is not much done with that. However, after that we see the negotiation for the next call:

BTS-Ports: RX 42478, TX 4002

So far this looks like a double use of the same connection. The same port number is used because the MSC thinks that the call is already released and that it can re-use the same endpoint again, but the BSC and the BTS still think that there is an ongoing connection. I think this is why the call gets messed up and we hear the garbled noise.

#6 - 07/05/2017 05:00 PM - dexter

- File messed_up_call.pcapng added

Here is a capture that only contains the problematic part. A call is placed to the non reachable end, immediately after another call is placed to the reachable end.

#7 - 07/05/2017 05:04 PM - dexter

The call to the reachable end starts at packet 581.

#8 - 07/05/2017 05:24 PM - neels

in **the first** pcap, capture.pcapng, I see:

(packet index)

(201) call to MSISDN 23006 starts. Call is setup successfully for both ends, MGCP succeeds to bridge the calls (1566, 1567).
The call ends with CC Releases (3235..3242) and DLCX (3237, 3243).
However, the RTP packets continue to flow after that.

The first normal call uses SSRC=0x41850C60 and SSRC=0x18AC38B.
After the CC Release, SSRC=0x41850C60 continues, the other one happens only occasionally.

All RTP with these SSRCs ceases (3828).

Then "nothing" happens for about one minute.

A second call is started to MSISDN 23101 (3967), it gets canceled by "Unassigned (unallocated) number" (3969).
Some RTP happens for SSRC=0x4E2C8D4F, and ends (4508).

About one second passes.

A third successful call is established for MSISDN 23006 again (4538), MGCP is setup successfully and RTP stream happens for two different SSRCs.

It's not really apparent why that RTP would be erratic. I see the same MGCP ports being re-used (1@mgw, 2@mgw), but no RTP streams seem to mix.

#9 - 07/05/2017 05:38 PM - neels

In the second pcap, I also see no RTP streams crossing. ("don't cross the streams!")

One detail I notice is that the RTP payload type is modified by the MGCPGW.
Coming from the BTS, it is GSM 06.10, then our code overwrites it with DynamicRTP-Type-98.
That's due to openbsc/src/libmgcp/mgcp_network.c:mgcp_patch_and_count()

```
413: int payload = rtp_end->codec.payload_type;
[...]
540: rtp_hdr->payload_type = payload;
```

Not sure why the payload change is there, must be a specific hack.
I had a patch to switch this off once: http://git.osmocom.org/openbsc/commit/?id=1804823c92dcb34b100a89e95b7cc17476fb46d0

#10 - 07/07/2017 09:44 AM - dexter

I think one important part of the problem is the releasing of a channel. The MSC receives the Release Complete DTAP message, but it does not

respond with a clear command. The clear command has to follow immediately after the release has completed.

```
|<----------- Channel Activation ---------------|                                                |
|---------- Channel Activation Ack ------------>|                                                |
|------------ Assignment Command -------------->|                                                |
|------------- Release Complete --------------->|                                                | #96
|                                               |------------ DTAP/ Release Complete ---------->| The MSC shou
ld send a clear comande here?
|----------- Establish Indication ------------->|                                                |
|------------ Assignment Complete ------------->|                                                |
|<--------------- Release Req ------------------|                                                |
|<------------- ip.access CRCX -----------------|                                                |
```

#11 - 07/07/2017 12:02 PM - dexter

- File fixed_msc.log added

- File fixed_bsc.log added

- File fixed.pcapng added

- Status changed from In Progress to Resolved

- % Done changed from 0 to 100

I think the problem is fixed now. We now transmit the clear command when the subscriber connection is freed. This informs the BSC properly about the fact that the connection has ended. Also calling a non available subscriber does not harm anymore. Tried it several times and it worked for all testcalls.

#12 - 08/08/2017 07:06 PM - laforge

- Status changed from Resolved to Closed

Files

capture.pcapng	1.12 MB	07/05/2017	dexter
messed_up_call.pcapng	422 KB	07/05/2017	dexter
fixed_msc.log	63.7 KB	07/07/2017	dexter
fixed_bsc.log	26.1 KB	07/07/2017	dexter
fixed.pcapng	16.9 KB	07/07/2017	dexter