OsmoBSC - Feature #3102

clean up and use a_reset.c in osmo-bsc

03/23/2018 01:23 PM - neels

Status:	Resolved	**Start date:**	03/23/2018
Priority:	Low	**Due date:**	
Assignee:	dexter	**% Done:**	100%
Category:			
Target version:			
Spec Reference:			

Description

After issue #3041 was fixed by the new gscon FSM, it became apparent that the a_reset.c FSM is actually no longer used at all: a_reset_conn_fail() is never invoked.

I also noticed in osmo_bsc_reset.c, we have an evil twin of a_reset.c still in the code base.
Its events also seem to never be invoked.
(BTW, in osmo-msc, we also never invoke a_reset_conn_fail())

In summary, the Reset FSMs are in desperate need of some tender loving care, they are currently abandoned and useless, apparently.

The proper trigger to invoke a_reset_conn_fail(): it was mentioned that specific N-Disconnect causes indicate a failure versus a normal disconnect.
a_reset_conn_fail() should only be invoked upon receiving such failure cause.

If possible, some ttcn3 test should simulate such a failure disconnect.

Also debatable is the conn_loss_count: if there are only two SCCP connections, and both of them go up in flames, the reset FSM should probably not wait for a third erratic disconnect; it should fire a BSSMAP Reset as well?

Related issues:

Related to OsmoMSC - Feature #3103: use a_reset.c in osmo-msc	**Resolved**	**03/23/2018**
Related to OsmoBSC - Bug #3041: osmo-bsc initiates BSSMAP Reset after every f...	**Rejected**	**03/07/2018**

History

#1 - 03/23/2018 01:27 PM - neels

- Description updated

#2 - 03/23/2018 01:29 PM - neels

- Related to Feature #3103: use a_reset.c in osmo-msc added

#3 - 03/23/2018 01:30 PM - neels

- Related to Bug #3041: osmo-bsc initiates BSSMAP Reset after every fourth BSSMAP Clear added

#4 - 03/23/2018 02:02 PM - dexter

The file osmo_bsc_reset.c turned out to be dead code (was not not even compiled). I have removed the file now: https://gerrit.osmocom.org/7476

#5 - 04/12/2018 09:24 AM - dexter

- Status changed from New to Feedback

- Assignee changed from dexter to neels

I have checked the problem back. The most convenient way to call a_reset_conn_fail() is when we notice that an SCCP connection indeed failes (CR is not answered). From the logical perspective and when I understand the things right, I should get an OSMO_SCU_PRIM_N_DISCONNECT in osmo_bsc_sigtran.c:sccp_sap_up(). Unfortunately I do not see any OSMO_SCU_PRIM_N_DISCONNECT indication at all. Shouldn't the SCCP stack

make sure that there actually is a connection made and inform in case the connection attempt failes?

Apart from this it is also debatable if we need the error detection at all. The reason why osmo-msc and osmo-bsc work fine even with a dysfunctional error detection is because both parties are sending an BSSMAP reset when they start up. So if one side goes down, the opposite side is informed when the side that was down is up again. However, we could run into problems when we use a third party MSC that does not do a bssmap RESET when it is starting up (I do not know how strict the spec is in this case and how common the MSC sideded reset is)

#6 - 04/12/2018 12:30 PM - laforge

Hi Philipp,

On Thu, Apr 12, 2018 at 09:24:07AM +0000, dexter [REDMINE] wrote:

> Unfortunately I do not see any OSMO_SCU_PRIM_N_DISCONNECT indication
> at all. Shouldn't the SCCP stack make sure that there actually is a
> connection made and inform in case the connection attempt failes?

yes. The following timer from libosmo-sccp/src/sccp_scoc.c applies:

/* Appendix C.4 of Q.714 (all in milliseconds) */
#define CONNECTION_TIMER»·····(1 * 60 * 100)

So basically after you send a OSMO_SCU_PRIM_N_CONNECT.req, and there is
no OSMO_SCU_PRIM_N_CONNECT.resp within 60 seconds, you should get a
OSMO_SCU_PRIM_N_DISCONNECT.ind

The scoc fsm is in the "S_CONN_PEND_OUT" state during those 60 seconds.

Plesae double check, possibly with debugg logging on "lsccp", if you
really don't get that DISCONNECT.ind without timeout.

It's also something for which it would be rather easy to write a test:
Send the CONNECT.req, wait more than 1 minute and check for result.

However, as this is an API test of libosmo-sigtran, it would have to be
a C-language test case, and not a TTCN-3 test.

> However, we could run into problems when we use a third party MSC that
> does not do a bssmap RESET when it is starting up (I do not know how
> strict the spec is in this case and how common the MSC sideded reset
> is)

I don't have any data on this either :/

#7 - 04/13/2018 03:29 PM - dexter

I think I found the reason why we do not get the N_DISCONNECT.ind. Its simply because the timer 993210 (20sek) is set expiring earlier than the CONNECTION_TIMER (60sek) in libosmo-sccp. When I increase 993210 to more than 60sek, then I get the N-DISCONNECT.ind with cause code 0.

```
Fri Apr 13 15:30:16 2018 DRLL <0000> chan_alloc.c:665 (bts=0) channel load average is 10.24%
Fri Apr 13 15:30:16 2018 DRLL <0000> chan_alloc.c:678 (bts=0) T3122 wait indicator set to 14 seconds
Fri Apr 13 15:30:17 2018 DMSC <0008> osmo_bsc_sigtran.c:217 N-DISCONNECT.ind(2, , cause=0)
Fri Apr 13 15:30:17 2018 DMSC <0008> osmo_bsc_sigtran.c:224 SUBSCR_CONN[0x561cf2183dc0]{WAIT_CC}: Received Eve
nt DISCONNET.ind
Fri Apr 13 15:30:17 2018 DMSC <0008> bsc_subscr_conn_fsm.c:979 SUBSCR_CONN[0x561cf2183dc0]{WAIT_CC}: Terminati
ng (cause = OSMO_FSM_TERM_REGULAR)
Fri Apr 13 15:30:17 2018 DMSC <0008> bsc_subscr_conn_fsm.c:297 SUBSCR_CONN[0x561cf2183dc0]{WAIT_CC}: tossing a
ll MGCP connections...
Fri Apr 13 15:30:17 2018 DMSC <0008> bsc_subscr_conn_fsm.c:1017 SUBSCR_CONN[0x561cf2183dc0]{WAIT_CC}: Releasin
g lchan
Fri Apr 13 15:30:17 2018 DRLL <0000> chan_alloc.c:540 (bts=0,trx=0,ts=0,ss=1) starting release sequence
```

So, libosmo-sccp seems to work fine here.

Here is a fix that uses the expiration of 993210 to inform the A-RESET FSM about a connection problem. When a certain number of connection errors occurred consecutively, then the BSC starts sending out BSSMAP RESET messages again.
https://gerrit.osmocom.org/7794

#8 - 04/16/2018 08:11 PM - dexter

- % Done changed from 0 to 40

We might have the following error situation:

Problem: The MSC restarts, but since it lost all its state it does not know about any BSC, eventually the MSC can not inform about the reset.

There are two possible approaches I see here:

1) We drop all input from an unknown (not yet known) BSC. The BSC will notice that the connection is bad and fall back into the disconnected state, sending continues BSSMAP resets until an answer is received.

2) When the MSC receives a message from an unknown BSC, it responds with a BSSMAP reset immediately.

Approach No. 1 is currently implemented. I wonder if No. 2 is cleaner, but maybe we should keep it as it is.

#9 - 04/17/2018 12:23 PM - neels

I share the impression that we might not need the FSM, given the Reset sent to a BSC on first seeing it, and the BSC sending Reset on restarting.
It would be good if you shared your test results. Have you played out various scenarios?
Like, what currently happens when we do:

- restart MSC
  - during "silence"
  - during ongoing communication.
- same for BSC.
- pull some ethernet cable between MSC and BSC
  - "left" of the STP
  - "right" of the STP
- re-connect ethernet cable after a while
- kill the STP, restart after a while

Does any lock-up or too long timeouts occur before the system becomes operational?
Would the a_reset FSM help with that?

#10 - 04/17/2018 12:23 PM - neels

- Assignee changed from neels to dexter

#11 - 05/07/2018 10:55 AM - dexter

Did some test, when restarting the MSC in total silence everything seems to be ok, but when I restart it in the middle of some communication things go problematic. I think this mostly has something to do with the phones which are not registered anymore but still think that they are registered.

Like with the MSC I also cleaned up the code that handles the BSSMAP reset. See also: https://gerrit.osmocom.org/#/c/8055/

#12 - 05/07/2018 05:10 PM - laforge

On Mon, May 07, 2018 at 10:55:46AM +0000, dexter [REDMINE] wrote:

> Did some test, when restarting the MSC in total silence everything
> seems to be ok, but when I restart it in the middle of some
> communication things go problematic.

which "communication things" go problematic in which specific way?

> I think this mostly has something to do with the phones which are not
> registered anymore but still think that they are registered.

In that case, there's a clear recovery path: Either
a) the next location update from a MS (periodic or due to location
change) will re-create the state in VLR+MSC, or
b) the next MO transaction will fail due to "subscriber unknown in VLR"

which will trigger the MS to perform a LU right away.

MT transactions are not possible until either of those two things happen.

#13 - 05/17/2018 02:02 PM - laforge

- Priority changed from Normal to Low

#14 - 05/28/2018 09:18 AM - dexter

- Status changed from Feedback to Resolved

- % Done changed from 40 to 100

The following two patches (already merged) finish up the cleanups to a_reset.c:

https://gerrit.osmocom.org/#/c/osmo-msc/+/8054/
https://gerrit.osmocom.org/#/c/osmo-bsc/+/8055/

The problem I noticed ("communication things") was caused by the usage of wrong cause codes. The wrong cause codes prevented the MS to try an LU when the network rejects the CM-Service request. This is now also fixed:

https://gerrit.osmocom.org/#/c/osmo-msc/+/9169/