Cellular Network Infrastructure - Feature #2623

SCCP/M3UA: detect restart of osmo-msc and osmo-sgsn

11/07/2017 11:25 PM - neels

Status:	New	**Start date:**	11/07/2017
Priority:	High	**Due date:**	
Assignee:	laforge	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Spec Reference:			

Description

Connecting osmo-bsc and osmo-hnbgw to the MSC and SGSN via an OsmoSTP instance, it is currently not possible to detect that the MSC or SGSN has restarted.

Scenario: using a sysmoBTS as a NITB, change MSC config, restart MSC -- now osmo-bsc happily continues to run and does not even notice that it is an entirely new MSC instance running in the core net now.

In the old days, the SCCPlite link would go down, but since now OsmoSTP is in-between and has no concept of who depends on who, no-one is notifying BSC or HNBGW that MSC or SGSN have gone down. Find out how this is intended to be solved if at all, and devise a way how osmo-bsc will restart and/or reconnect to a new MSC instance, and so forth.

History

#1 - 12/03/2017 09:07 PM - laforge

- Assignee set to laforge

neels wrote:

> Connecting osmo-bsc and osmo-hnbgw to the MSC and SGSN via an OsmoSTP instance, it is currently not possible to detect that the MSC or SGSN has restarted.

The specified way to treat this is the A interface RESET procedure (and I'm sure Iu has the same?). So the MSC should perform a RESET procedure towards the BSC after it has started new, to erase all state in the BSC.

What's problematic here is that with our "dynamically accept any BSC from any point code" approach, the re-started MSC has no clue about where BSCs might be. One possible (but ugly) approach would be to simply flood this RESET to an entire range of point codes that's configurable at the MSC.

> Scenario: using a sysmoBTS as a NITB, change MSC config, restart MSC -- now osmo-bsc happily continues to run and does not even notice that it is an entirely new MSC instance running in the core net now.

I presume you're hinting that the "MSC config change" included a change of the MSC's point code?

One could implement the classic SCCP messsages / primitives for infomring the BSC that the MSC is no longer reachable at the old point code. On the MTP-level, this is a MTP-STATUS.ind from the MTP up into the SCCP stack. The SCCP stack then would use N-PCSTATE.ind (Q.711 6.3.2.3.3)

The BSC would then receive a N-PCSTATE.ind and thus know the MSC is (at least temporarily) gone. However, an intermittent failure of the intermediate signaling network would look exactly the same, so there would probably need to be some kind of timeout, i.e. if the MSC is not again reachable shortly after it is gone, we behave as if we received an implicit RESET.

Another way to move forward is for the MSC to keep local state as to which BSCs were connected, so that after a crash it can send RESET to all of those point codes.

> In the old days, the SCCPlite link would go down, but since now OsmoSTP is in-between and has no concept of who depends on who, no-one is notifying BSC or HNBGW that MSC or SGSN have gone down. Find out how this is intended to be solved if at all, and devise a way how osmo-bsc will restart and/or reconnect to a new MSC instance, and so forth.

Well, with an entire (routed!) signaling network between BSC and MSC, the status of the signaling link (M3UA connection) has nothing to do anymore with whether or not the MSC is reachable. Let's imagine one or multiuple STPs in between: Any of them can go down, or any of the links can temporarily fail and recover without the BSC or MSC ever being down or losing their state.

So I guess the best we can do is for the BSC to detect "MSC unavailability" by timeout on any of its SCCP connections or connectionless procedures. If and when the MSC detects any message from an (unknown) BSC, the MSC will generate a RESET procedure and remove all state. Isn't this what's already happening now?

#2 - 12/04/2017 12:11 PM - neels

laforge wrote:

> The specified way to treat this is the A interface RESET procedure (and I'm sure Iu has the same?). So the MSC should perform a RESET procedure towards the BSC after it has started new, to erase all state in the BSC.
>
> What's problematic here is that with our "dynamically accept any BSC from any point code" approach, the re-started MSC has no clue about where BSCs might be. One possible (but ugly) approach would be to simply flood this RESET to an entire range of point codes that's configurable at the MSC.
>
> > Scenario: using a sysmoBTS as a NITB, change MSC config, restart MSC -- now osmo-bsc happily continues to run and does not even notice that it is an entirely new MSC instance running in the core net now.
>
> I presume you're hinting that the "MSC config change" included a change of the MSC's point code?

Not really. I mean if the MSC changes configuration items that affect the BSC (though nothing comes to mind); actually I think I also meant that subscribers are still considered attached though the restarted MSC does not. The point is, usually we restart programs when their "server" restarts, like when the BSC goes down, we restart the BTS and hence are sure they are in sync. If the MSC goes down, the BSC currently doesn't ever get notified, because of the above mentioned: we only do the BSSAP Reset dance when the BSC re-attaches.

> One could implement the classic SCCP messsages / primitives for infomring the BSC that the MSC is no longer reachable at the old point code. On the MTP-level, this is a MTP-STATUS.ind from the MTP up into the SCCP stack. The SCCP stack then would use N-PCSTATE.ind (Q.711 6.3.2.3.3)
>
> The BSC would then receive a N-PCSTATE.ind and thus know the MSC is (at least temporarily) gone. However, an intermittent failure of the intermediate signaling network would look exactly the same, so there would probably need to be some kind of timeout, i.e. if the MSC is not again reachable shortly after it is gone, we behave as if we received an implicit RESET.

who would trigger that, the OsmoSTP?

> Another way to move forward is for the MSC to keep local state as to which BSCs were connected, so that after a crash it can send RESET to all of those point codes.

i.e. keep persistent state. That would solve it, but we would need persistent state ;)

> So I guess the best we can do is for the BSC to detect "MSC unavailability" by timeout on any of its SCCP connections or connectionless procedures. If and when the MSC detects any message from an (unknown) BSC, the MSC will generate a RESET procedure and remove all state. Isn't this what's already happening now?

hmm, need to check

#3 - 12/04/2017 12:50 PM - laforge

Hi Neels,

On Mon, Dec 04, 2017 at 12:11:42PM +0000, neels [REDMINE] wrote:

> actually I think I also meant that subscribers are still considered
> attached though the restarted MSC does not.

This is what the RESET procedure is for.

> The point is, usually we restart programs when their "server"
> restarts, like when the BSC goes down, we restart the BTS and hence
> are sure they are in sync.

BTS+BSC is different: They cannot function without each other. A BTS cannot even allocate a radio channel without a BSC. So there's very tight integration.

Between BSC and MSC it's a bit different. At least with modern features like MOCN, you can actually have a BSC talking to multiple MSCs (even of different operators) and there is not such a strict inter-dependency.

> If the MSC goes down, the BSC currently doesn't ever get notified,
> because of the above mentioned: we only do the BSSAP Reset dance when
> the BSC re-attaches.

The BSC should get notified as soon as it sends the first packet to the MSC, which triggers a RESET procedure from the MSC as it doesn't know any state about.

This should be rather quick. The only situation in which this takes a long time is if there's absolutely no activity from any MS.

> > One could implement the classic SCCP messsages / primitives for
> > infomring the BSC that the MSC is no longer reachable at the old
> > point code. On the MTP-level, this is a MTP-STATUS.ind from the MTP
> > up into the SCCP stack. The SCCP stack then would use
> > N-PCSTATE.ind (Q.711 6.3.2.3.3)
> >
> > The BSC would then receive a N-PCSTATE.ind and thus know the MSC is
> > (at least temporarily) gone. However, an intermittent failure of
> > the intermediate signaling network would look exactly the same, so
> > there would probably need to be some kind of timeout, i.e. if the
> > MSC is not again reachable shortly after it is gone, we behave as if
> > we received an implicit RESET.
>
> who would trigger that, the OsmoSTP?

Yes, the STP (or also the local libosmo-sigtran on the client side) would generate such messages whenever point codes become available or unavailable.

> Another way to move forward is for the MSC to keep local state as to which BSCs were connected, so that after a crash it can send RESET to all of those point codes.

> i.e. keep persistent state. That would solve it, but we would need persistent state ;)

well, this is why **normally** one configures the point code of each BSC
in the MSC. At that point the MSC can send a RESET after start to each
of them :)

> > So I guess the best we can do is for the BSC to detect "MSC
> > unavailability" by timeout on any of its SCCP connections or
> > connectionless procedures. If and when the MSC detects any message
> > from an (unknown) BSC, the MSC will generate a RESET procedure and
> > remove all state. Isn't this what's already happening now?
>
> hmm, need to check

yes, if that's not the case we're definitely broken.