OsmoBSC - Feature #3682
Intra-domain connection of OsmoBSC to multiple MSCs
11/06/2018 02:14 PM - laforge

<table>
<thead>
<tr>
<th>Status:</th>
<th>In Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>neels</td>
</tr>
<tr>
<td>Category:</td>
<td>A interface</td>
</tr>
</tbody>
</table>

Start date: 11/06/2018
Due date: 
% Done: 10%

Description
3GPP TS 23.236 specifies the "Intra-domain connection of Radio Access Network (RAN nodes to multiple Core Network (CN) nodes".

This introduces two concepts:

- a deviation from the strict hierarchy of classic GSM architecture where a BSC is always only served by one MSC
- the introduction of "MSC pools" where a pool of multiple MSCs cover one geographic service area.

This appears to be functioning by dividing the TMSI range between different MSCs, using the higher-order TMSI bits to identify the MSC and the lower-order TMSI bits for the actual TMSI within that MSC.

Section 5.3 describes the functions required to be implemented in the BSC.

- NAS Node Selection Function
  - investigate Initial L3 message at RR establishment
  - if TMSI is present, extract NRI and route request to MSC for that NRI
  - if only IMEI or IMSI is present, or no route for that NRI, send to 'random' MSC (load distribution)
  - when any MSC sends a PAGING REQUEST, store the IMSI <-> MSC relation as long as the paging is active
  - any incoming response to that paging gets routed to the MSC that has issued the paging

Related issues:
- Related to OsmoPCU - Feature #4472: Intra-domain connection of OsmoPCU to multiple MSCs
- Related to OsmoBSC - Feature #3454: disable/constrain/hide "multiple msc" config

History
#1 - 01/25/2019 12:47 PM - laforge

#2 - 09/04/2019 09:01 AM - laforge
- File 142217065-MSC-Pool.pdf added
- Assignee deleted (laforge)

#3 - 03/29/2020 06:10 PM - laforge
- Checklist item changed from splitting of TMSI allocation range/spce to Function for deriving NRI from TMSI
- Description updated
- Spec Reference set to TS 23.236

#4 - 03/29/2020 06:11 PM - laforge

#5 - 03/29/2020 06:16 PM - laforge

In terms of the test suite, I would expect this has to cover:

- different NRI bit-width
- routing of TMSI with active MSC configured for NRI
- routing of TMSI with disconnected MSC configured for NRI

05/16/2020
- routing of TMSI without any MSC configured for NRI
- routing of TMSI with NULL NRI
- routing of PAGING RESPONSE back to correct MSC that requested paging

#6 - 03/29/2020 06:25 PM - laforge
- Related to Feature #4472: Intra-domain connection of OsmoPCU to multiple SGSNs added

#7 - 03/29/2020 07:11 PM - laforge
- Related to Feature #3454: disable/constrain/hide "multiple msc" concept added

#8 - 05/12/2020 12:40 AM - neels
Most parts of 23.236 are quite clear to me, except these:

NAS Node Selection Function
The BSC (RNC) selects an MSC to service new MS (UE). It says this selection is implementation specific. Should osmo-bsc be able to take into account how loaded each MSC is? If yes, how?

- We could do round-robin between configured MSCs for each not yet assigned MS.
- The BSC could try to keep counters of Location Updatings and IMSI Detach. However, it would also have to track each IMSI with its periodic LU period to not lose implicit detach from periodic LU timeout. That would amount to a second VLR in the BSC, probably not what we want.
- The BSC could somehow communicate with the MSC, e.g. receive load indications from the MSC (am not aware of such messages specified by 3GPP).
- Also, 23.236 names "low access priority" MS (Machine Type Comms), i.e. a way to direct specific kinds of subscribers to a particular MSC. We could easily make provision for marking specific MSCs as low access priority, but the question would be how to identify those MS: by IMSI regular expression? by communication with the HLR via MSC (am not aware of such messages specified by 3GPP)? We could make provision for IMSI regexes to directly select specific MSCs by osmo-bsc.cfg, if that makes any sense.

Non-Broadcast LAI
For Load Re-Distribution, the sequence outlined in the specs is:

- configure the BSC to no longer assign new MS to an MSC that should be unloaded.
- switch the MSC to unloading mode.
- wait several periodic LU periods, so that the MSC responds to periodic LU requests with a NULL NRI, as well as a Non-Broadcast LAI.
- in CS, the "terminal" [sic], because of the Non-Broadcast LAI, immediately re-issues another LU.
- the RAN then sees the NULL NRI in the LU and picks another MSC for the LU.

How does this Non-Broadcast LAI work?
The spec says that each MSC must have an individual LAI assigned to be its own unique Non-Broadcast LAI, i.e. it is a configured item per-MSC. However, the above sequence suggests that the MS is the one re-issuing another LU request right away (the "terminal"?). How should the MS know the individually configured Non-Broadcast LAIs?
Is this simply a LAI that mismatches the cell's LAI somehow and thus the MS rejects the LU Accept message?

Re-Distribution: communicate between MSCs
Apparently the above Non-Broadcast LAI should be used to communicate between the two MSCs:

Each CN node in the pool has to be aware of the non-broadcast LAI/RAI assigned to the other CN nodes in the pool, because in case of re-distribution the 'target CN node' will retrieve data (e.g. IMSI, security context, MM & PDP contexts) from the 'offloaded CN node' based on non-broadcast LAI/RAI.

We have no provision made for OsmoMSCs to communicate in this way. Will we implement some Osmocom-specific inter-MSC GSUP communication for this (and need to correlate Non-Broadcast LAIs to MSCs’ GSUP IPA names in osmo-msc.cfg)?
Will we not do Load Re-Distribution at all?
Should we still have a good plan to add it in the future?

(Also I am thinking, if the MSCs involved in the re-distribution do communicate with each other, why even send the NULL NRI and not directly the new MSC’s NRI?
I'm guessing to leave the decision for which other MSC is chosen in the RAN...)

#9 - 05/12/2020 01:17 AM - neels

Another open question:
The spec shows two bits of the TMSI as "CS/PS", and a number of bits (e.g. 5) for 'VLR-restart'.

When we implement assigning TMSIs with NRI in osmo-msc, should we also make provision for these parts of a TMSI?

**CS/PS**

Since TMSI and pTMSI live in separate domains, I see no benefit in keeping the upper two bits constant per domain.
I'm guessing this is for scenarios where CS and PS are routed more intimately, as some parts of 23.236 hint at.

**VLR-restart**

It is not apparent from 23.236 how the VLR-restart bits are useful to a CN. So far I am guessing:

It may avoid TMSI collisions where a restarted VLR assigns a TMSI that another MS still assumes to possess. That MS would try to Layer-3-Complete with its TMSI, would be accepted as another MS, but then the auth should fail, causing re-attach by IMSI. If this collision is avoided, that means that an Identity Request for IMSI is issued as part of the first LU.

The suggested 5 bits for VLR-Restart seem to suggest that a VLR likely restarts up to 31 times before MS re-attach and get new TMSIs? or that each time a VLR restarts, the VLR-restart bits are chosen randomly, in the hope to pick a different value every time??

Either way, seems to me a lot of TMSI bits sacrificed for a minor improvement, but what do I know.

---

#10 - 05/12/2020 12:06 PM - neels
- Checklist item [x] Function for deriving NRI from TMSI set to Done

#11 - 05/12/2020 12:13 PM - neels
- Checklist item [] figure out how to test with multiple MSCs in ttcn-bsc-tests added

We need multiple MSCs in the ttcn-bsc-tests

#12 - 05/12/2020 12:15 PM - neels
- Status changed from New to In Progress
- Assignee set to neels

#13 - 05/12/2020 12:15 PM - neels
- % Done changed from 0 to 10

#14 - 05/12/2020 12:30 PM - laforge

neels wrote:

> Should osmo-bsc be able to take into account how loaded each MSC is? If yes, how?

I would go for a simple implementation first, e.g.

- We could do round-robin between configured MSCs for each not yet assigned MS.

exactly.

If somebody needs a more complex scheme, we could always add that as a separate second stage.

**Non-Broadcast LAI**

For Load Re-Distribution, the sequence outlined in the specs is:

- configure the BSC to no longer assign new MS to an MSC that should be unloaded.
- switch the MSC to unloading mode.
- wait several periodic LU periods, so that the MSC responds to periodic LU requests with a NULL NRI, as well as a Non-Broadcast LAI.
- in CS, the "terminal" [sic], because of the Non-Broadcast LAI, immediately re-issues another LU.
- the RAN then sees the NULL NRI in the LU and picks another MSC for the LU.

How does this Non-Broadcast LAI work?

It is a LAI that is never broadcast, i.e. it is never advertised in SYSTEM INFORMATION, but you return it only in the LU ACCEPT on the dedicated (hence non-broadcast) channel to the MS.

The spec says that each MSC must have an individual LAI assigned to be its own unique Non-Broadcast LAI, i.e. it is a configured item
per-MSC. However, the above sequence suggests that the MS is the one re-issuing another LU request right away (the "terminal"?).

Yes, the MS is doing that if the LU ACCEPT contains a LAI that is != the one of the SI of the cell it camps on.

How should the MS know the individually configured Non-Broadcast LAIs?
Is this simply a LAI that mismatches the cell's LAI somehow and thus the MS rejects the LU Accept message?

Exactly.

We have no provision made for OsmoMSCs to communicate in this way.
Will we implement some Osmocom-specific inter-MSC GSUP communication for this (and need to correlate Non-Broadcast LAIs to MSCs' GSUP IPA names in osmo-msc.cfg)?
Will we not do Load Re-Distribution at all?
Should we still have a good plan to add it in the future?

This feature is about OsmoBSC. OsmoMSC is out of scope.

#15 - 05/12/2020 12:34 PM - laforge

neels wrote:

Another open question:

The spec shows two bits of the TMSI as "CS/PS", and a number of bits (e.g. 5) for 'VLR-restart'.
When we implement assigning TMSIs with NRI in osmo-msc, should we also make provision for these parts of a TMSI?

OsmoMSC is out of scope for this feature development. It may only be required for testing (if TTCN-3 based testing is not sufficient and/or one wants to do real end-to-end testing. So whatever we extend on OsmoMSC, it should be minimized in terms of effort as the BSC is what's in scope, not the MSC.

CS/PS

Since TMSI and pTMSI live in separate domains, I see no benefit in keeping the upper two bits constant per domain.
I'm guessing this is for scenarios where CS and PS are routed more intimately, as some parts of 23.236 hint at.

The question is whether or not this is mandatory as per spec. If it's mandatory, we have to do it. IF not, we can do whatever is easy for us.

VLR-restart

It is not apparent from 23.236 how the VLR-restart bits are useful to a CN. So far I am guessing:

I also don't know without re-reading. But why is this relevant for developing the related code in OsmoBSC? The BSC cannot influence which TMSIs are allocated in the MSC anyway.

#16 - 05/13/2020 12:55 AM - neels

In BSC_Tests.ttcn, I have managed to duplicate the BSSAP connections.
However, I now realize that we will have the same issue that I've faced in the MSC_Tests.ttcn for inter-BSC handover testing:
Each testing function will run on a separate MSC_ConnHandler, and thus I will need concurrent testing functions, one for each MSC.

For example, in MSC_Tests.ttcn, TC_ho_inter_bsc(), I coordinate an inter-BSC handover by running f_tc_ho_inter_bsc0() and f_tc_ho_inter_bsc1() concurrently,
and waiting for certain events to occur (or have a sufficient delay) to orchestrate between the two functions.

That can work fine, only I just realized that the tests will not be strictly sequential and might be a bit hard for the reader to follow.

- Three RSL L3-Complete requests for different NRI from a BTS, three separate test functions each expect one L3-Complete for their NRI.
- Paging: three test functions, each on a different MSC, each sending out a Paging for a different identity and expecting their own Paging Responses.
- NULL NRI: one test function responds upon LU with a NULL NRI, a second test function merely waits for a LU and accepts it.
- ...  

IIUC each concurrent test function will be able to interact with the RSL side, so the RSL part should probably be written in only one of the test functions for clarity?
I wonder if that will work out well.
the patch is http://git.osmocom.org/osmo-ttcn3-hacks/commit/?h=neels/mscppool&id=080aa41f2c179bed65664cfa8228f2bea7708764
"bsc: allow multiple MSCs"

BSC tests still pass with this after adjusting the BSC_Tests.cfg in docker-playground
http://git.osmocom.org/docker-playground/commit/?h=neels/mscppool&id=038c048d27c34b5eb6a2e433caa447a2b030529
"bsc: adjust cfg for multiple MSCs, msc: tweak comments in cfg"

#17 - 05/13/2020 01:02 AM - neels

On Wed, May 13, 2020 at 12:55:23AM +0000, neels [REDMINE] wrote:

In BSC Tests.ttcn, I have managed to duplicate the BSSAP connections.

great!

However, I now realize that we will have the same issue that I’ve faced in the MSC_Tests.ttcn for inter-BSC handover testing:
Each testing function will run on a separate MSC_ConnHdlr, and thus I will need concurrent testing functions, one for each MSC.

I thought contrary to hand-over, this is what you wanted for the MSC-pool work?

- Three RSL L3-Complete requests for different NRI from a BTS, three separate test functions each expect one L3-Complete for their NRI.

Shouldn’t this simply be three components, each running the same code (with different parameters such as TMSI)?

What is the need for coordination between them?

side note: Do you even need to run them concurrently? You could also run them sequentially?

- Paging: three test functions, each on a different MSC, each sending out a Paging for a different identity and expecting their own Paging Responses.

Again, same function/testcase on the same component type (ConnHdlr), each just executed with different parameters leading to using a different BSSAP emulation?

- NULL NRI: one test function responds upon LU with a NULL NRI, a second test function merely waits for a LU and accepts it.

Why is this not handled in one component?

Each component emulates both the MS/BTS side and one MSC. All logic always happens within that component.

IIUC each concurrent test function will be able to interact with the RSL side, so the RSL part should probably be written in only one of the test functions for clarity?

I cannot follow you here, sorry.

#18 - 05/13/2020 11:20 AM - laforge

On Wed, May 13, 2020 at 12:55:23AM +0000, neels [REDMINE] wrote:

In BSC Tests.ttcn, I have managed to duplicate the BSSAP connections.

great!

However, I now realize that we will have the same issue that I’ve faced in the MSC_Tests.ttcn for inter-BSC handover testing:
Each testing function will run on a separate MSC_ConnHdlr, and thus I will need concurrent testing functions, one for each MSC.

I thought contrary to hand-over, this is what you wanted for the MSC-pool work?

- Three RSL L3-Complete requests for different NRI from a BTS, three separate test functions each expect one L3-Complete for their NRI.

Shouldn’t this simply be three components, each running the same code (with different parameters such as TMSI)?

What is the need for coordination between them?

side note: Do you even need to run them concurrently? You could also run them sequentially?

- Paging: three test functions, each on a different MSC, each sending out a Paging for a different identity and expecting their own Paging Responses.

Again, same function/testcase on the same component type (ConnHdlr), each just executed with different parameters leading to using a different BSSAP emulation?

- NULL NRI: one test function responds upon LU with a NULL NRI, a second test function merely waits for a LU and accepts it.

Why is this not handled in one component?

Each component emulates both the MS/BTS side and one MSC. All logic always happens within that component.

IIUC each concurrent test function will be able to interact with the RSL side, so the RSL part should probably be written in only one of the test functions for clarity?

I cannot follow you here, sorry.

#19 - 05/14/2020 03:17 PM - neels

laforge wrote:

- Three RSL L3-Complete requests for different NRI from a BTS, three separate test functions each expect one L3-Complete for their NRI.

Shouldn’t this simply be three components, each running the same code (with different parameters such as TMSI)?

yes, that could work. actually it will probably even “disappear” into something like f_location_updating()...

- NULL NRI: one test function responds upon LU with a NULL NRI, a second test function merely waits for a LU and accepts it.

Why is this not handled in one component?
One MSC sends a NULL NRI and non-broadcast LAI, then the subscriber needs come back with that NULL NRI and be redirected to another MSC. Could of course be modeled in separate parts, too, but would be nice to have all in one scenario...
I guess I'll manage either way.