OsmoBSC - Feature #1606

hand-over for load distribution among BTSs

02/23/2016 04:00 PM - laforge

Status:	Resolved	**Start date:**	02/23/2016
Priority:	High	**Due date:**	
Assignee:	neels	**% Done:**	80%
Category:	Handover		
Target version:			
Spec Reference:			

Description

We currently only do hand-over in case of a better available link/cell. However, hand-over can also be performed to move load from a high-load BTS to a lower-load BTS with sufficient signal level.

I believe jolly branches had some features along these lines, this needs investigation + merge.

Related issues:

Related to OsmoMSC - Feature #1778: avoid mismatching TCH/F vs TCH/H pchan types	**Resolved**	**07/21/2016**
Related to OsmoBTS - Bug #2790: dynamic timeslots: failure when all TS are us...	**Rejected**	**12/28/2017**
Related to OsmoBSC - Bug #2964: handover_decision_2: measurement report numbe...	**In Progress**	**02/19/2018**
Related to OsmoBSC - Bug #3002: HO2: handover decision for dynamic timeslots	**New**	**02/26/2018**
Related to OsmoBSC - Support #3487: comprehensive documentation for Handover ...	**Resolved**	**08/20/2018**
Related to OsmoBSC - Feature #1608: various handover improvements, meta-issue	**Rejected**	**02/23/2016**
Related to OsmoBSC - Feature #3638: handover decision 2: load balancing acros...	**New**	**10/09/2018**
Related to OsmoBSC - Feature #4189: TTCN-3 tests for load-based hand-over in BSC	**New**	**09/04/2019**
Blocked by Cellular Network Infrastructure - Bug #2963: Measurement Reports c...	**Resolved**	**02/19/2018**

History

#1 - 10/06/2017 02:28 PM - laforge

#2 - 10/06/2017 02:28 PM - laforge

- Checklist item [ ] implementation of load-based hand-over added
Checklist item [ ] automatic tests for load-based hand-over added
Checklist item [ ] jenkins/CI integration of automatic tests added

#3 - 11/07/2017 08:26 PM - laforge

- Assignee set to neels

#4 - 11/07/2017 08:28 PM - laforge

- Priority changed from Normal to High

#5 - 11/07/2017 09:33 PM - laforge

- Priority changed from High to Urgent

#6 - 11/08/2017 02:17 PM - neels

wow, another urgent task... let's talk about prioritization soon.

#7 - 11/08/2017 09:20 PM - laforge

On Wed, Nov 08, 2017 at 02:17:35PM +0000, neels [REDMINE] wrote:

> wow, another urgent task... let's talk about prioritization soon.

I basically juts use the levels for relative priority, so don't take "urgent"
literally, it just is easier to move a few to "urgent" rather than moving all

others from high to normal.

#8 - 11/20/2017 12:34 PM - neels

I'm not starting on this yet, please let me know in case I should start on this before completing #2618 / #2638

#9 - 11/24/2017 12:08 AM - neels

- Status changed from New to In Progress

Looking at openbsc.git's jolly/new_handover branch:

The branch

- applies various boilerplate patches, like
  - moving handover config to BTS level,
  - adding DHODEC logging category
  - allow triggering certain actions from VTY
- improves on "existing" handover code in various smaller patches, then moves all of it over to handover_1.c
- finally introduces a handover_2.c coined as improved handover decision code in a single large commit (>2k lines of code).
- adds an elaborate handover test suite

While both "old" and "new" HO algorithms remain configurable from VTY, apparently, the handover_2.c is the way to go from this branch. It exhibits three HO criteria levels that seem quite sane:

- A: on the target cell, HO must be enabled, channel types must exist, max count of handovers must not be exceeded.
- B: after the HO, the minimum of free channels as configured must still be available.
- C: the HO must bring an improvement, so that the target cell must still have less load than the originating cell after the HO.

The amount of implementation as well as test code is quite substantial, it will take a while of reading and testing to wrap my head around it, and to uncover remaining open issues, if any. I will try to contact jolly to get his last high-level view on the code, he can probably/hopefully still remember whether there are profound problems with the code that would need to be fixed before merging to master.

#10 - 11/24/2017 01:02 AM - neels

Looking at the openbsc.git fairwaves/master-rebase branch:

The branch has various topics not related to HO, like registration proxy, external SMS handling, USSD proxy, external HLR. Handover related patches seem to come up only near the end of the branch:

- instruct an external MNCC with an MNCC_RTP_MODIFY message, so that the MNCC knows about a change in TCH payload type.
- implement MNCC_RTP_MODIFY in internal MNCC (separately).
- improve on signalling for the actual handover (HO-Command, HO-Detect, HO-Complete/Fail/expired)

The signalling part seems to overlap / mirror the jolly/new_handover branch patches. Particularly, both implement a ho_queue that caches signalling messages during handover to be flushed after the HO is done.

There seems to be no implementation of **load** based handover.

My conclusion so far is that we would like to implement jolly's load-based handover_2 algorithm in osmo-bsc, and to match this up with the fairwaves handover signalling improvements to adopt those as well if required. We also want to adopt fairwaves' MNCC related handover bits.

#11 - 11/24/2017 01:37 PM - kluchnikov

Hello Neels,
I have pushed branch users/kluchnikov/wip-handover to openbsc repo.
You can find there patches which include basic changes of handover code for rtp proxy mode.
Unfortunately these patches are not finished and not well tested, but I hope this branch will be helpfull for you.
Also I will be happy to help with reviewing and testing.

#12 - 11/24/2017 02:02 PM - neels

kluchnikov wrote:

> I have pushed branch users/kluchnikov/wip-handover to openbsc repo.

Thanks! I'll take a look and adopt it over to osmo-bsc.git.

#13 - 11/24/2017 02:09 PM - neels

https://osmocom.org/projects/openbsc/wiki/Proposed_New_Handover_Algorithm
http://lists.osmocom.org/pipermail/openbsc/2017-November/011498.html

#14 - 11/27/2017 02:46 PM - neels

will continue to...

- rebase jolly's handover_2 onto current osmo-bsc master
- enrich with fairwaves' improvements on the HO signalling
- grok jolly's HO test suite

#15 - 11/27/2017 02:59 PM - neels

- Related to Feature #1778: avoid mismatching TCH/F vs TCH/H pchan types added

#16 - 11/27/2017 04:45 PM - neels

open question: what is synchronous vs asynchronous handover? 3GPP 48.058 4.1.3/4.1.4 explain what is done in each of them, but I don't yet get the bigger picture. What is the practical difference, when do we want which, which do we need to support? So far all handovers seem to be RSL_ACT_INTER_ASYNC. Background: a patch from jolly adds an 'async' flag to struct bsc_handover, yet that flag seems to instead reflect whether the ho is inter- or intra-cell. I'd like to understand whether that is a misnomer or intentional.

#17 - 11/27/2017 04:50 PM - neels

neels wrote:

> a patch from jolly adds an 'async' flag to struct bsc_handover, yet that flag seems to instead reflect whether the ho is inter- or intra-cell. I'd like to understand whether that is a misnomer or intentional.

http://git.osmocom.org/openbsc/commit/?id=b4b630baf2a4731a0433c6d79462d11591397a2b

#18 - 11/27/2017 07:40 PM - laforge

On Mon, Nov 27, 2017 at 04:45:49PM +0000, neels [REDMINE] wrote:

> open question: what is synchronous vs asynchronous handover? 3GPP
> 48.058 4.1.3/4.1.4 explain what is done in each of them, but I don't
> yet get the bigger picture.

"synchronized" (not synchronous?) handover means that the target BTS already knows at which particular timing advance (distance) the phone will hand into the new cell. This way one can avoid using RACH bursts on the new channel to establish the timing advance (distance) of the phone to the new cell, which is done in non-sync handover.

It's called "synchronized" as it implies that the MS will already be synchronized to the timeslot sturcture of the target BTS by having the correct TA value to ensure the bursts arrive exactly inside the timeslot at the BTS.

> What is the practical difference, when do we want which, which do we
> need to support? So far all handovers seem to be RSL_ACT_INTER_ASYNC.

Correct, treating all hand-overs as asynchronous is the safe choice, unless your BSC has some detailed knowledge on how exactly the cells are located on a map and can hence make good estimates about the new TA in a target cell. So from the Osmocom point of view, sync HO is not interesting at all for the time being.

> Background: a patch from jolly adds an 'async' flag to struct
> bsc_handover, yet that flag seems to instead reflect whether the ho is
> inter- or intra-cell. I'd like to understand whether that is a
> misnomer or intentional.

in an intra-cell hand-over you of course know the TA: The distance between phone and BTS didn't change, as the new cell == old cell, and the phone will not move significantly (in terms of TA) during the hand-over procedure. So in that case you can actually implement sync HO as an optimization. I haven't studied the code in detail, but it seems that this is what the code might be doing?

#19 - 11/29/2017 12:58 PM - neels

- Project changed from OpenBSC to OsmoBSC

- Category deleted (libbsc)

#20 - 11/29/2017 01:15 PM - neels

laforge wrote:

> "synchronized" (not synchronous?)

48.058 calls it "4.1.3 Activation for Asynchronous Handover" and "4.1.4 Activation for Synchronous Handover"

Thanks, the TA explanation makes perfect sense.

BTW it also seems from the spec that one can send a synchronous channel activation msg type, but as long as no power or TA values are supplied along with it, it will actually behave like an async one.

> in an intra-cell hand-over you of course know the TA

While the patch does rsl_chan_activate_lchan(RSL_ACT_INTER_ASYNC) for inter-BTS, for intra it simply calls rsl_chan_activate_lchan(0). The zero amounts to RSL_ACT_INTRA_IMM_ASS, same as for the usual chan act. So it appears that the "intra handover" isn't really a big deal signalling wise, just re-assign and be merry?

The 0 instead of a proper RSL constant may indicate that that wasn't being tested properly; on the other hand the patch adds the 0 explicitly where it was always INTRA_ASYNC before. Anyway, I'll see during testing.

#21 - 12/04/2017 02:14 PM - neels

- % Done changed from 0 to 20

#22 - 12/04/2017 02:17 PM - neels

- Related to Feature #1749: add activation timeout in bsc_handover_start() added

#23 - 12/04/2017 03:59 PM - neels

implementation detail: jolly's HO code takes the MS's Bearer Capabilities into consideration: before looking for a cell to handover to, it matches the MS's supported codecs up with the cells' offered timeslots and permitted codecs. The Bearer Capabilities are typically included in CC messages, which we parse on the libmsc layer. Though OsmoBSC naturally passes these through to the MSC, we so far don't decode them AFAICT, and hence we don't know the Bearer Capabilities in OsmoBSC. I think I'll first continue without Bearer Capabilities and add CC decoding in OsmoBSC later. I will until then not limit the target cells by MS capabilities.

#24 - 12/04/2017 04:10 PM - laforge

On Mon, Dec 04, 2017 at 03:59:09PM +0000, neels [REDMINE] wrote:

> implementation detail: jolly's HO code takes the MS's Bearer Capabilities into consideration: before looking for a cell to handover to, it matches the MS's supported codecs up with the cells' offered timeslots and permitted codecs. The Bearer Capabilities are typically included in CC messages, which we parse on the libmsc layer. Though OsmoBSC naturally passes these through to the MSC, we so far don't decode them AFAICT, and hence we don't know the Bearer Capabilities in OsmoBSC. I think I'll first continue without Bearer Capabilities and add CC decoding in OsmoBSC later. I will until then not limit the target cells by MS capabilities.

Erm, the MSC is decoding the bearer capabilities and is putting them into the A interface ASSIGNMENT COMMAND as the list of permitted speech codecs. All the BSC needs to do is to keep that information/state and whenever attempting hand-overs look at that information.

I don't think the BSC should peek into L3 CC to obtain this.

Regards,
Harald

#25 - 12/05/2017 10:53 AM - neels

I was suspecting such but hadn't investigated properly. Thanks.

#26 - 12/05/2017 03:11 PM - neels

This question is a bit of step three before step one, but would be good to get an understanding of it in the meantime. I'll dig in the specs, if you already know details they are most welcome:

So,

- the MSC sends "Codec List (MSC Preferred)" during the BSSAP Assignment Request to the BSC.
- In the Assignment Complete, the BSC sends the "Speech Codec (Chosen)" back to the MSC;
  - along with a "Codec List (BSS Supported)" if "an intra-BSS cell change has occurred during the assignment."

I'm not entirely clear on what this means in practice:

- When the BSC sends to the MSC a "Codec List (BSS Supported)" "if an intra-BSS cell change has occurred during the assignment", what does that imply, that the BSS has chosen any one of those during the initial assignment? Possibly, that we are free to switch between these codecs?
- We tell the MSC our chosen codec during Assignment Complete. Hence if we later do intra-BSS handover during an ongoing call, are we required to stay with exactly that codec?
- Does the MSC not care about the speec codec / channel type happening on the BTS side, as long as OsmoBSC's OsmoMGW transcodes to whichever codec we sent as "Speech Codec (Chosen)"?

Notably, in the 48.008 PDF that we have in our software-lab network share (3GPP TS 48.008 version 6.9.0 Release 6 from 2005), these IEs are not listed. I see them documented in 3GPP TS 48.008 version 8.10.0 Release 8 from 2012.

#27 - 12/05/2017 06:40 PM - laforge

Hi Neels,

On Tue, Dec 05, 2017 at 03:11:05PM +0000, neels [REDMINE] wrote:

> This question is a bit of step three before step one, but would be good to get an understanding of it in the meantime. I'll dig in the specs, if you already know details they are most welcome:

I don't know the details off my head, sorry.

> Notably, in the 48.008 PDF that we have in our software-lab network share (3GPP TS 48.008 version 6.9.0 Release 6 from 2005), these IEs are not listed. I see them documented in 3GPP TS 48.008 version 8.10.0 Release 8 from 2012.

In classic GSM, the TRAU is co-located with the BSC. So the A interface (and hence the MSC) always only sees G.711 uncompressed audio on 64k ISDN B-channels. Hence, in original GSM, before the introduction of 3GPP AoIP, the MSC did never have to worry about speech codecs itself. Also, the BSC could of course freely choose whatever it wanted on the RAN side, as it had to transcode before hitting the A interface anyway.

I recommend to always use the latest version/release of the specs to avoid surprises.

#28 - 12/10/2017 08:31 PM - laforge

- Category set to Handover

#29 - 01/07/2018 04:08 PM - neels

- Related to Bug #2790: dynamic timeslots: failure when all TS are used as PDCH and the phone requests a TCH added

#30 - 01/10/2018 02:14 AM - neels

- % Done changed from 20 to 50

I have succeeded to get all unit tests from jolly/new_handover to work on current master.

The immediate tasks are...

- squash commits into useful chunks
- verify that this actually works in a real world BTS setup:
  - by writing ttcn3 tests;
  - by physical setup, possibly in the osmo-gsm-tester
- for automatic testing it would make sense to offer a CTRL interface to modify handover config
- add tests for dynamic channels, particularly TCH/F_TCH/H_PDCH open up interesting situations where have **either** 6 TCH/F available **or** 12 TCH/H, and deciding on /F or /H also affects congestion counts; two half used TCH/H joined together make room for a TCH/F and so forth. Jolly so far only considered TCH/F_PDCH dyn TS, which are only TCH/F and don't have this duality.

I also see a large field opening for future improvements. I'm getting to know this new algorithm and am also noticing some potential problems.
The congestion check seems to fall short of modeling the actual complexity; of course the question is whether it is a good enough simplification.
Let me mention what comes to mind...

1) When a cell is found to be congested, at most one lchan is handed over per congestion check. So even if N handovers would qualify at a check, it looks like only one will be handed over per BTS and congestion check timeout. The next one will be triggered on the next congestion check, after another timeout. Instead, we could trigger handover of two or more lchans consequtively (after all, there is a config item "max nr of handovers", set to 9999 in the test).

2) The concept of a timeout for congestion check seems a bit counter intuitive: I'm thinking, if we already know that assigning a new lchan will cause congestion, we might as well assign at a neighbor cell directly. (I'm still too noob to know whether I'm making good sense here, need to figure out to what extent we need measurement reports for resolving congestion / already have measurement reports at time of lchan assignment). Or we might as well trigger congestion resolution immediately. Currently, there is an N second timeout for congestion checks per BTS; intuitively it makes sense to me to trigger congestion checks on events that change the channel configuration / update the measurements, instead of a timer. OTOH, a timer might serve to reduce the computational load, so that we don't traverse all data structures on every little lchan change...

3) The Handover chapter in "Performance Enhancements in a Frequency Hopping GSM Network" [Nielsen,Wiegard,2002] suggests that it is more desirable to service ongoing calls in good quality than to accept new calls. So rather reject a new assignment instead of interrupting an ongoing call. The current algorithm tries to keep N TCH/F and TCH/H free in each cell at all times to avoid congestion. So if an lchan is degrading, but handover would cause congestion in the neighboring cell, it appears to me that it will actually refrain from handing it over, so that avoiding congestion is much stronger than ensuring call quality. Congestion is implemented as a boolean concept, i.e. if <N TCH/x are free, it is congested, without a gradient. Considering ongoing calls to be more important than new calls, I think we need to strengthen the case for handover into a congested cell.

4) Instead of trying to keep each cell at a fixed number of free channels, we should probably consider distributing congestion evenly. For example, if cell A has only 1 TCH/F free, and cell B has 3 TCH/F free, common sense would tell us that moving from A to B would reduce congestion. The current algorithm would see though that e.g. B wants 3 TCH/F free at all times, and will not move from A to B even though A is much worse off. So congestion could be a gradient more than a boolean of "are N available or not".

5) Intuitively it makes sense to me to decide everything by weighting gradients. For example, we would strive to evenly distribute lchans across neighboring cells at all times, if all RXLEV/RXQUAL were equal everywhere. If an MS experiences much better RX at another cell, we should allow it to handover there, even if that causes a degree of congestion. So my idea is that congestion could contribute to a gradient towards triggering handovers, and RXLEV and RXQUAL each add to the same gradient, with different weighting depending on how grave the situation is and on how much improvement a neighboring cell would provide. So each MS "naturally" gravitates to its optimal cell... If the resulting overall gradient surpasses a given threshold, handover is triggered for an MS; overall gradients between MS become comparable. Ideally, the algorithm would notice that e.g. two MS could swap the cells to minimize the overall gradient, instead of considering each MS on its own and doing nothing since the other cell is already considered congested. A drawback of such weighting is that the numbers may become hard to grok, and configuration could become esoterical. The current algorithm has config that is easy to understand. This is quite hand wavy and obviously a huge academic and non-trivial field. I might have some fun trying to model a weighting algorithm, but it would probably rather be a handover_decision_3.c (the jolly/new_handover is currently coined handover_decision_2.c).

...

1 thru 4 may be worth investigating by modeling many more tests with common sensical situations. But for now I'll try to not drift into 5), into inventing a new algorithm altogether.

I'll also make the tests keep going a little longer to make sure there are no flip flop situations (even passing the penalty timers) -- ideally, a decision to handover should lead to a situation where we don't want to handover again unless things change, and the tests should ensure that.

In short, testing testing testing. This stuff has infinite permutations and we need to at least trace the landscape of it.

#31 - 01/10/2018 02:31 AM - neels

forgot to mention, I need a little codecs 101; jolly's code used Bearer Capabilities, but now with the A interface we have a Codec List from the MSC. It amounts to the same thing, but the naming is slightly different. Also, back in the days we used to have bsc_codec_conf.afs and .ahs, now we have only .amr, besides efr and hr. I need to translate code like

```
                /* only select AFS/AHS, if supported by the MS */
                bcap = &lchan->conn->bcap;
                for (i = 0; bcap->speech_ver[i] >= 0; i++) {
                        if (bcap->speech_ver[i] == 4 && bts->codec.afs)
                                requirement |= REQUIREMENT_A_TCHF;
                        if (bcap->speech_ver[i] == 5 && bts->codec.ahs)
```

```
                                requirement |= REQUIREMENT_A_TCHH;
                }
```

I figured out how to translate speech_ver to FR1..HR3,
but not sure whether the distinction between FR and HR is possible or needed with bts->codec.amr
or whether I need to check other bts->codec.*.

In the Speech Codec List, e.g.

```
        lchan->conn->codec_list = (struct gsm0808_speech_codec_list){
                .codec = {_
                        { .fi=true, .type=GSM0808_SCT_FR1, },
                        { .fi=true, .type=GSM0808_SCT_FR2, },
                        { .fi=true, .type=GSM0808_SCT_FR3, },
                        { .fi=true, .type=GSM0808_SCT_HR1, },
                        { .fi=true, .type=GSM0808_SCT_HR3, },
                },
                .len = 5,
        };
```

I don't understand the .fi, .pi, .pt, .tf relevance yet.

(currently assuming all codecs are supported as a dev hack)

#32 - 01/10/2018 02:54 AM - ipse

I just want to add from my personal experience with the network we're currently installing. I observed a situation when there were three cells with a quite poor signal but one of them was providing good call quality while the other two were not (up to a point of complete audio loss) and the call was constantly handed over between these three cells which made the quality completely suck - firstly because handover event itself introduce a quite noticeable click in these conditions and secondly because the call was moved from a cell with good voice quality (Q) to a cell with worse Q. I haven't had a chance to investigate this further (partially because we don't have handover logs so it's difficult to dig into issues retroactively). But it feels like we're currently missing a condition to avoid handover to a stronger cell if the current voice quality is ok, or somehow else avoid cyclic handover.

#33 - 01/10/2018 10:50 AM - laforge

On Wed, Jan 10, 2018 at 02:54:33AM +0000, ipse [REDMINE] wrote:

> I just want to add from my personal experience with the network we're currently installing. [...]

This is using which handover algorithm? The one in osmocom master or the one from jolly?

> I haven't had a chance to investigate this further (partially because we don't have handover logs so it's difficult to dig into issues retroactively).

I would **presume** you can find our which handover has happened from where to where by enabling the related logging family and parsing the regular log files?

If not, then the log file is missing inportant information and the LOGP statements need to be amended with whatever is missing.

> But it feels like we're currently missing a condition to avoid handover to a stronger cell if the current voice quality is ok, or somehow else avoid cyclic handover.

Handover has no notion of "voice quality". It operates at a much lowe level, it only knows about RxLev/RxQual. Not doing handovers because we've not had success with a given MS + BTS combination some minutes ago is also wrong, as we cannot assume that the MS didn't move meanwhile.

It's a hard optimization problem, and one that you can spend an indefinite amount of tuning and implementing algorithims. Which is why I originally introduced the separation between handover decision making and handover execution, allowing anyone to implement their own version of a handover algorithm. Patches are always welcome, but if nobody even submits what has been implemented in their private branches, then we of course won't eve benefit from this. I'm also happy for more modularity here, I'd even merge patches for a plug-in architecture.

#34 - 01/10/2018 10:50 AM - laforge

Hi Neels,

On Wed, Jan 10, 2018 at 02:31:05AM +0000, neels [REDMINE] wrote:

> but not sure whether the distinction between FR and HR is possible or needed with bts->codec.amr

I'm not quite sure how one relates to the other? You are speaking of GSM 08.08 in your message, but now you are referring to something else, like the AMR codec configuration? Why would a distinctin between HR and FR not be needed?

> or whether I need to check other bts->codec.*.

As always when a BSC allocates a channel, it must consider

- the capabilities of the BTS hardware/firmware/software which it **should** know either from the BTS attributes fetched at runtime (Max implemented a mechanism for this, but will work only on OsmoBTS), or based on some kind of static compiled-in information (such as the bts_model for nanobts knowing that nanoBTS doesn't implement HRv1, or a BS-11 not supporting AMR/F or AMR/H)
- the codec list provided by the MSC at ASSIGNMENT time
- any administrative restrictions imposed by the sysadmin on either the BTS level (vty) or the global level (for this MSC or for the BSC globally).

I have a feeling I'm discussing this topic once every week with another person, which makes me rather worried. After all, I believe we're talking about something as simple as the bit-wise AND of four bit-masks in the BSC? Am I missing something?

Regards,
Harald

#35 - 01/10/2018 11:18 AM - neels

The immediate practical situation is I have code from jolly that uses bts->codec.{afs,ahs}.
bts->codec has changed and no longer features afs,ahs. So I need to figure out what those meant, whether amr == afs | ahs...
I need to tap knowledge on the topic to make good choices.

Now you mention BTS attributes / knowing what the BTS can do: if it's not reflected by bts->codec.*, then jolly's code needs to be extended for that.

#36 - 01/10/2018 12:13 PM - laforge

On Wed, Jan 10, 2018 at 11:18:29AM +0000, neels [REDMINE] wrote:

> The immediate practical situation is I have code from jolly that uses bts->codec.{afs,ahs}.

I have no memory of such a data structure/record. Either I'm forgetting more things that I'd want to, or those fields simply never existed in regular osmocom upstream.

> bts->codec has changed and no longer features afs,ahs.

a 'git log -u | grep' through the entire openbsc.git history doesn't show them, so I don't think that anything has changed.

To the contrary, it seems you might be missing some of jolly's changes on which the load handover code depends?

> whether amr == afs | ahs... I need to tap knowledge on the topic to make good choices.

AFS is AMR on full speed channels and AHS is AMR on half speed channels, sure. That's standard 3GPP spec language.

You will find plenty of spec references of a TCH/AFS and a TCH/AHS to indicate a TCH/F or TCH/H with related AMR voice payload inside.

> Now you mention BTS attributes / knowing what the BTS can do: if it's not reflected by bts->codec.*, then jolly's code needs to be extended for that.

Sorry, I don't know what is reflected where, I just know how I assume it works / how it should work.

I'm really surprised that there's so much discussion around this topic. As indicated, all we have to make sure is that we have those ~4 different bit-masks and that we bit-wise AND them, and that all parts of the code in their respective area use/create/update thsoe bit-masks.

#37 - 01/10/2018 04:09 PM - neels

Re, the bts->codec.{afs,ahs} vs. .amr "code change":

- On jolly/new_handover, the bts->codec was added in e9d06be699f53d9e5117a9b31f822a2f61ab715c, containing afs and ahs.
- On master, the bts->codec was added in a83d511b618c1e18b324e04db433a2fd111b2d6f, with amr instead of afs and ahs. So there isn't a visible change in the history, just two separate additions; a diff would show it.
- I see now that these items get populated by the vty 'bts' / 'codec-support fr...' config item. So I believe interpreting the presence of amr as supporting both AFS and AHS is a sound choice.
  If we would like to offer more fine grained 'codec-support' config for afs and ahs separately, we can add it later.

Re bitwise-AND: sounds really trivial, but the Speech Codec List received during Assignment is not compatible with the Bearer Capabilities, compare enum gsm48_bcap_speech_ver (used in Bearer Cap, gsm/protocol/gsm_04_08.h) and enum gsm0808_speech_codec_type (used in Speech Codec List, gsm/protocol/gsm_08_08.h). In osmo-bsc I'll just stick to using the gsm0808_speech_codec_type enum values.

BTW, you make it sound like I'm raising a discussion, while I'm just trying to figure out what all those names mean and logging my progress here; just in case someone has info that can cut short my learning curve...

#38 - 01/11/2018 01:10 AM - neels

first manual tests with physical equipment aren't working out yet...

- the RTP stream gets muted, MGW says "RTP from wrong IP address"; the MGW apparently does get an MDCX that should switch the RTP source to the other BTS, but either the RTP is still coming from the old BTS or the MDCX is doing something wrong...
- it seems the old lchan keeps sticking around after HO.

Will pursue avenues:

- ask pmaier on expected status after BSC uses MGW.
- Get a trivial manual HO triggered by VTY command to work reliably.
- incorporate fairwaves' HO branch, since that is precisely about making handovers work more reliably.

#39 - 01/11/2018 03:48 PM - laforge

On Wed, Jan 10, 2018 at 04:09:30PM +0000, neels [REDMINE] wrote:

> Re bitwise-AND: sounds really trivial, but the Speech Codec List received during Assignment is not compatible with the Bearer Capabilities, compare enum gsm48_bcap_speech_ver (used in Bearer Cap, gsm/protocol/gsm_04_08.h) and enum gsm0808_speech_codec_type (used in Speech Codec List, gsm/protocol/gsm_08_08.h). In osmo-bsc I'll just stick to using the gsm0808_speech_codec_type enum values.

I suggest those different types have to be translated into one common representation. I mean, after all we are talking about a set of five (!) codecs, that need to be translated from their particular protocol/layer specific format into a common representation and then matched. That common representation could even come with its own set of VTY commands, and all code in the osmocom code could be converted to use the common representation.

In fact, there would be two different common representations:

a) a simple bit-mask for capability of network element XYZ
b) an ordered list in order of preference to express policy. This policy then is read at the MSC from the VTY and handed down into the A interface as codec preference, where it is applied to the merged masks of bts/bsc?

#40 - 01/11/2018 11:28 PM - neels

To verify, I have tested manual handover and assignment using the current osmo-bsc master branch -- results:

- Assignment (handover within the same BTS) works well. The voice call continues on another lchan, RTP is redirected by MDCX to another port on the same BTS.
- Handover between two BTS does not work. RTP is redirected by MDCX to the other BTS, but the new BTS never starts to send RTP; RTP continues to flow from the old BTS and lchan.

I will go on to compare the two and try to pinpoint where exactly things break. And will test with fairwaves' branch to find out whether the failure is solved with their patches.

#41 - 01/12/2018 04:29 AM - neels

neels wrote:

> - Handover between two BTS does not work. RTP is redirected by MDCX to the other BTS, but the new BTS never starts to send RTP; RTP continues to flow from the old BTS and lchan.

I was quite sure of that, but subsequent tests showed that on osmo-bsc master handover works fine between two BTS as well. Maybe the build was somehow stale...

On handover2 code (aka jolly's branch rebased onto master), I also got both Handover and Assignment to work now:

jolly's code attempts to do a re-assignment within the same BTS with an RR Assignment Command and an Immediate Assignment channel activation.
On the master branch, we successfully use Handover Command and Async Chan Act successfully, for both HO and AS.
When going back to using Handover Command and Async Chan Act on jolly's code, Assignment is fixed.
My first guess is that the Assignment Command might not actually be intended for re-assigning an lchan, only for initial assignment? Will need to read up in the specs to confirm.
Handover was broken only because there was a swap (probably edited in by me) that swapped HO with Assignment code paths (and sent RR Assignment Command for handover instead of Handover Command; now both do essentially the same and there is only one code path, except for logging "Assignment" instead of "Handover").

Right, so now I'm in a code state where manually triggered handover and assignment still work with all of jolly's code rebased onto master, and I can concentrate back on the actual load based handover decision making. (osmo-bsc.git branch neels/ho)

Re: the HO patches in the fairwaves/master-rebase branch in openbsc.git: the interesting bits are almost exclusively in openbsc.git's libmsc, and when applied to osmo-bsc master are simply lost in the rebase.
The HO bits from openbsc.git/fairwaves/master-rebase are in osmo-bsc.git neels/fairwaves_ho_orig (branching off a point where libmsc was still around).

- The patches in libmsc: mostly the DTAP cache that stores DTAP while handover is ongoing, to flush the cache when HO is done. Jolly's branch has a similar DTAP cache that is implemented in libbsc and was easy to apply to osmo-bsc master.
- Then there is MNCC related handover stuff, doing RTP-Modify negotiation via MNCC, and some early returns / value assignments in select places that I can't figure out (comments would have been good). At first I'm thinking that HO within the BSC should not concern MNCC in any way, but if a BSC HO actually changes the codec, we might want to somehow tell the call router about it ... or transcode?
- There is an rsl_ipacc_mdcx() placed in an S_LCHAN_HANDOVER_DETECT signal callback, which I haven't figured out yet: was it somehow missing? Is it a safeguard for a specific situation?
- There are a handful of logging improvements that I've applied.

I'll ask Ivan from fairwaves with a summary of the rebase results to see whether I'm missing some important points.

#42 - 01/12/2018 05:21 AM - ipse

laforge wrote:

> On Wed, Jan 10, 2018 at 02:54:33AM +0000, ipse [REDMINE] wrote:
>
> > I just want to add from my personal experience with the network we're currently installing. [...]
>
> This is using which handover algorithm? The one in osmocom master or the one from jolly?

Master one.

> > I haven't had a chance to investigate this further (partially because
> > we don't have handover logs so it's difficult to dig into issues
> > retroactively).

> I would **presume** you can find our which handover has happened from where to where by enabling the related logging family and parsing the regular log files?
>
> If not, then the log file is missing inportant information and the LOGP statements need to be amended with whatever is missing.

I assume it's possible but I'm not sure - I haven't looked into extracting those from log files as it's a bit cumbersome to implement on a permanent basis.

> > But it feels like we're currently missing a condition to avoid handover to a stronger cell if the current voice quality is ok, or somehow else avoid cyclic handover.
>
> Handover has no notion of "voice quality". It operates at a much lowe level, it only knows about RxLev/RxQual.

Yes, RxQ is what I mean by "Quality (Q)".

> Not doing handovers because
> we've not had success with a given MS + BTS combination some minutes ago
> is also wrong, as we cannot assume that the MS didn't move meanwhile.

My suggestion is opposite - don't do handover if the quality is good on the current cell.

> It's a hard optimization problem, and one that you can spend an indefinite amount of tuning and implementing algorithims. Which is why I originally introduced the separation between handover decision making and handover execution, allowing anyone to implement their own version of a handover algorithm. Patches are always welcome, but if nobody even submits what has been implemented in their private branches, then we of course won't eve benefit from this. I'm also happy for more modularity here, I'd even merge patches for a plug-in architecture.

I agree with this. What I heard about "big vendor" implementations is that handover is moved completely out of the core into a separate function which uses a DB with handover conditions/rules - in many cases specific to each cell. This allows an operator's engineer to tweak handover decisions without touching any code. This looks quite appealing to me - allow an external decision making instead of hardcoding and spending a lot of time trying to optimize for every possible case (and still failing as conditions vary greatly). This could be a DB or Lua (which I assume can talk to a DB) or something else?

#43 - 01/12/2018 01:10 PM - neels

ipse wrote:

> "big vendor" implementations is that handover is moved completely out of the core into a separate function which uses a DB with handover conditions/rules - in many cases specific to each cell. This allows an operator's engineer to tweak handover decisions without touching any code.

Technically we can configure each cell individually from the osmo-bsc VTY without touching any code. Would probably be nice to have the same config API on the CTRL interface.
(On master it is even mandatory to copy handover config for each "bts N", on my current development branch there are defaults on the network level which can or can not be overridden by each individual cell config)

> This looks quite appealing to me - allow an external decision making instead of hardcoding

We also have functionality to manually trigger handovers. So I guess if we add a CTRL TRAP that feeds measurement reports and lchan events to an external program, that program could trigger handovers externally; implemented as a handover_decision_external.c or something, patches welcome :)

A clear advantage of this would be that we don't use osmo-bsc cycles to calculate handover. osmo-bsc is a single-threaded program, and if we were to do extensive complex calculations for each cell in the (upcoming) congestion check, we could affect response time of actual subscriber signalling. That would need to be a huge number of cells and/or an exponential algorithm to really become noticeable.

But I'm also thinking that in osmo-bsc code, all the data structures and cb triggers are already present, and an external program would have to duplicate those. So, currently, my call would be to rather implement the handover decision within the osmo-bsc code base, and offer parameters that can be adjusted externally in such a way that an operator doesn't need to touch code. The congestion check might take care not to hog the main thread for too long, e.g. by checking at most N cells at a time.

#44 - 01/13/2018 07:20 PM - laforge

On Fri, Jan 12, 2018 at 04:29:42AM +0000, neels [REDMINE] wrote:

> My first guess is that the Assignment Command might not actually be intended for re-assigning an lchan, only for initial assignment? Will need to read up in the specs to confirm.

Please always specify the protocol (which then implies the interface) for the command.

In RR, there's ASSIGNMENT CMD and IMMEDIATE ASSIGNEMNT CMD. In BSSMAP there's only one ASSIGNMENT CMD.

- The patches in libmsc: mostly the DTAP cache that stores DTAP while handover is ongoing, to flush the cache when HO is done. Jolly's branch has a similar DTAP cache that is implemented in libbsc and was easy to apply to osmo-bsc master.

In a split BSC/MSC network, you have two levels of this:

- one queue in the BSC which is used for MT L3 Messages received while a intra-BSC HO is ongoing
- one queue in the MCS which is used for MT L3 Messages received (e.g. from MNCC) while an inter-BSC HO is ongoing.

- Then there is MNCC related handover stuff, doing RTP-Modify negotiation via MNCC, and some early returns / value assignments in select places that I can't figure out (comments would have been good). At first I'm thinking that HO within the BSC should not concern MNCC in any way, but if a BSC HO actually changes the codec, we might want to somehow tell the call router about it ... or transcode?

I completely agree that hand-over should not be exposed on MNCC. It's an internal property of the cellular network, and no external entity should have to deal with it. The extenral MNCC handler (such as osmo-sip-connector) must provide/express the supported codecs via MNCC, and then the entire GSM network must make sure to provide the call within those permitted/supported codecs. Whether it decides to transcode or simply never hand-over to a RAN path that doesn't support any of the permitted codecs is an implementation/policy detail.

> I'll ask Ivan from fairwaves with a summary of the rebase results to see whether I'm missing some important points.

thanks.

#45 - 01/15/2018 09:11 PM - neels

laforge wrote:

> On Fri, Jan 12, 2018 at 04:29:42AM +0000, neels [REDMINE] wrote:
>
> > My first guess is that the Assignment Command might not actually be intended for re-assigning an lchan, only for initial assignment? Will need to read up in the specs to confirm.
>
> Please always specify the protocol (which then implies the interface) for the command.

oh I thought I had written RR, must have gone missing in an edit...

So what I need to find out is: if a conn has an lchan assigned, can we reassign to another lchan on the same cell by doing an GSM48_MT_RR_ASS_CMD? AFAICT using the GSM48_MT_RR_HANDO_CMD works fine in that case, and am not sure why jolly's code changes that to an RR Assignment (and whether it worked for him). (If anyone knows answers that'd be nice, otherwise I'll dig in the specs when I get to it)

> In a split BSC/MSC network, you have two levels of this:
>
> - one queue in the BSC which is used for MT L3 Messages received while a intra-BSC HO is ongoing
> - one queue in the MCS which is used for MT L3 Messages received (e.g. from MNCC) while an inter-BSC HO is ongoing.

Yes, I guess jolly's DTAP cache will end up in osmo-bsc, and Ivan's will end up in osmo-msc.

> I completely agree that hand-over should not be exposed on MNCC. It's an internal property of the cellular network, and no external entity should have to deal with it. The extenral MNCC handler (such as osmo-sip-connector) must provide/express the supported codecs via MNCC, and then the entire GSM network must make sure to provide the call within those permitted/supported codecs. Whether it decides to transcode or simply never hand-over to a RAN path that doesn't support any of the permitted codecs is an implementation/policy detail.

Right, so if MNCC supports a range of HR and FR codecs, and now the BSC decides to handover from TCH/F to TCH/H due to load, then from the sip-connector's point of view the RTP will just all of a sudden change from FR to HR. My question is, is that permissible, or does MNCC need an explicit notification that the payload changed? The fairwaves branch sends an MNCC_RTP_MODIFY, which will be hard to do if the MSC never

knows when the BSC has done an Intra-BSC HO. For Inter-BSC, the MSC of course would know about it and could comfortably do the MNCC_RTP_MODIFY.

#46 - 01/15/2018 10:13 PM - laforge

On Mon, Jan 15, 2018 at 09:11:37PM +0000, neels [REDMINE] wrote:

> So what I need to find out is: if a conn has an lchan assigned, can we reassign to another lchan on the same cell by doing an GSM48_MT_RR_ASS_CMD? AFAICT using the GSM48_MT_RR_HANDO_CMD works fine in that case, and am not sure why jolly's code changes that to an RR Assignment (and whether it worked for him). (If anyone knows answers that'd be nice, otherwise I'll dig in the specs when I get to it)

well, per definition it's an assignment inside a cell and a hand-over to another cell, so his code seems correct. Technically, I think not a lot is different between a synchronized HO and an assignment.

> > I completely agree that hand-over should not be exposed on MNCC. It's an internal property of the cellular network, and no external entity should have to deal with it. The extenral MNCC handler (such as osmo-sip-connector) must provide/express the supported codecs via MNCC, and then the entire GSM network must make sure to provide the call within those permitted/supported codecs. Whether it decides to transcode or simply never hand-over to a RAN path that doesn't support any of the permitted codecs is an implementation/policy detail.
>
> Right, so if MNCC supports a range of HR and FR codecs, and now the BSC decides to handover from TCH/F to TCH/H due to load, then from the sip-connector's point of view the RTP will just all of a sudden change from FR to HR.

Correct. But I would suppose that a lot of practical scenarios don't permit that, as the external SIP enttity won't suport all those codecs

> My question is, is that permissible, or does MNCC need an explicit notification that the payload changed?

I would presume one would want to know that event, yes.

> The fairwaves branch sends an MNCC_RTP_MODIFY, which will be hard to do if the MSC never knows when the BSC has done an Intra-BSC HO.

If the BSC changes codec (even among the permitted codecs) it has to send some kind of "intra-bss handover performed" message to the MSC, as far as I remember the specs.

So in inter-BSC HO, the BSC asks the MSC to do hand-over. In inter-BSC handover where any of the relevant parameters like ciphering/codec change it simply notifies the MSC.

> For Inter-BSC, the MSC of course would know about it and could comfortably do the MNCC_RTP_MODIFY.

ACK. But as indicated, even in intra-BSS handover the MSC will know.

#47 - 01/16/2018 01:07 AM - ipse

I'm not sure about every MNCC client but with SIP clients you can change RTP codec without any signaling, as long as the new codec is in the reciever's SDP. The receiver will know this by looking into the RTP Payload Type and match it against the SDP.

Changing IP is what actually requires SIP signalling, because in classic SIP implementation you would send your RTP stream to what your peer specified in the SDP. So in case of handover you need to do a SIP re-INVITE. Some implementations ignore the SDP IP after an initial packet is received from the other party and just send to the same IP/port this RTP was received from. This is done to circumvent various NATs and proxies seen in the wild Internet. But even then, I don't think RTP stream will follow changes in the source IP/port and you would need a re-INVITE anyway.

#48 - 01/16/2018 11:20 AM - neels

ipse wrote:

> I'm not sure about every MNCC client but with SIP clients you can change RTP codec without any signaling, as long as the new codec is in the reciever's SDP. The receiver will know this by looking into the RTP Payload Type and match it against the SDP.

That sounds interesting, though I'm still not clear on those payload types. I have the uninformed impression that the payload type numbers aren't clearly defined for all codecs... or are they?

> Changing IP is what actually requires SIP signalling

We pair an osmo-mgw with each osmo-bsc to re-route the RTP to the BTS in a way that is transparent to outside of the BSS. In short, the IP address and port towards the call router will not change during intra-bsc handover.

Right, now I have two pointers: RTP payload can just change without notification, and the MSC should be notified of handovers. I guess if we implement the MSC notification anyway, it's simple enough to do the MNCC_RTP_MODIFY as well.

#49 - 01/16/2018 11:52 AM - ipse

neels wrote:

> ipse wrote:
>
> > I'm not sure about every MNCC client but with SIP clients you can change RTP codec without any signaling, as long as the new codec is in the reciever's SDP. The receiver will know this by looking into the RTP Payload Type and match it against the SDP.
>
> That sounds interesting, though I'm still not clear on those payload types. I have the uninformed impression that the payload type numbers aren't clearly defined for all codecs... or are they?

SDP is what maps PT (payload type) to the actual codec. That's what's called a "dynamic RTP payload type" SDP is a really small standard - look into it and it'll clear a lot of your questions. Btw, SDP is used in some telco protocols as well.

Relevant RFCs:
https://tools.ietf.org/html/rfc4566
https://tools.ietf.org/html/rfc3264
https://tools.ietf.org/html/rfc4317

There are more RFCs about specific cases, but these ones are most important.

> Changing IP is what actually requires SIP signalling

> We pair an osmo-mgw with each osmo-bsc to re-route the RTP to the BTS in a way that is transparent to outside of the BSS. In short, the IP address and port towards the call router will not change during intra-bsc handover.

Ok. This didn't exist in NITB, so we had to do re-invites. And it seems we still have to do them for inter-BSC handovers.

> Right, now I have two pointers: RTP payload can just change without notification, and the MSC should be notified of handovers. I guess if we implement the MSC notification anyway, it's simple enough to do the MNCC_RTP_MODIFY as well.

If you're doing this anyway, it's then surely easy to ignore those notifications when they are not needed :)

#50 - 01/16/2018 01:40 PM - laforge

On Tue, Jan 16, 2018 at 11:20:11AM +0000, neels [REDMINE] wrote:

> That sounds interesting, though I'm still not clear on those payload types. I have the uninformed impression that the payload type numbers aren't clearly defined for all codecs... or are they?

the entire point of SDP **is** to establish a mapping between the dynamically-allocated payload
type and the string identifying the media format. It's the same in all protocols using SDP,
whether MGCP, SIP, ...

#51 - 01/19/2018 03:49 AM - neels

- % Done changed from 50 to 70

After looking through handover_test.c from jolly's code in detail, I can clarify some comments above:

- As soon as rxlev drops below the configurable min_rxlev, handover is done despite congestion. So that's good. The algorithm also handles the case that a more congested cell can hand over to a less congested. So the congestion concept isn't all that boolean as I have described it above, much better than I thought at first. To illustrate...

- A "congested" cell is one that has less free timeslots than the min-free-slots setting (there are two, one for TCH/F and one for TCH/H).
- Only when an MS hits critically low RXLEV do we handover into a "congested" cell.
- If we set min-free-slots to 1, a cell would fill up, and only when it is completely full do we consider handing over 1 lchan due to load.
- As long as a cell is not considered congested, we freely allow handovers for any RXLEV, as long as another RXLEV is better. If the hysteresis were set to zero, that would mean a lot of handovers. Increasing the hysteresis reduces the amount of handovers.
- Setting the amount of desired free channels to the maximum (min-free-slots = amount of lchans) in all BTS would mean that RXLEV triggers handover only when it goes below the min-rxlev value, and it also means that we will constantly attempt to keep the amount of used channels equal on all cells in the neighborhood.
- So the min-free-slots setting is more like a threshold number above which we equalize load and below which we freely choose the best RXLEV cell. (when equalizing load, we still pick the best target RXLEV from the list of HO candidates, but would allow HO to less RXLEV than the current cell)

- This is still true: "1) When a cell is found to be congested, at most one lchan is handed over per congestion check. So even if N handovers would qualify at a check, it looks like only one will be handed over per BTS and congestion check timeout. The next one will be triggered on the next congestion check, after another timeout. Instead, we could trigger handover of two or more lchans consequtively (after all, there is a config item "max nr of handovers", set to 9999 in the test)." -- This max-nr-of-handovers refers to concurrent handovers per cell, so that if e.g. a measurement report at cell A triggers a handover to cell B, and another at cell C also to cell B, and then congestion check wants to handover to B as well while the other handovers aren't complete yet ... in that sense. We could still change the code so that more than one congestion handover is triggered per cell and congestion check interval; not sure if that's a big improvement really or might make it too trigger happy instead.

- in "2)" I argued against doing congestion at a timeout. When reconsidering, though, calculating all handover conditions and values on each and every lchan change would cause considerable load. It is better to do them periodically. It is simpler (= less bug surface) to not collect trigger events but just do the check every N seconds. In an intermediate code state I also had separate timeouts per each cell, so as to not run the entire congestion check in one go and not hog the process for too long, but that would have lead to huge complexity, for example to detect that a BTS comes into or goes out of operation, noticing a change of the trigger interval for each cell and so on. I've gone back to one single shot of checking all BTSes at once. If we hit an overload situation (actually unlikely with the present algorithm) we can limit the number of BTS per run, which is also not all that trivial (to ensure a round robin).

I found a few remaining problems in jolly's code. Namely, doing an RR Assignment within a cell didn't work for me, might come back to that another time. Also the measurement report count would grow indefinitely while the last_seen_nr was limited to 255, which wouldn't have worked for long ... things like that.

I have not managed to write TTCN3 tests at all yet. So far testing is covered by jolly's test suite and manual runs with two cells and two phones.

In summary, I think this algorithm is good enough as it is now to pass as beta testing basis. I have concrete ideas for improvement, but the current parameters allow for enough flexibility that one should be able to get good load distribution by tweaking them. So I will push the patches to gerrit now. Most of it has an effect only when osmo-bsc is configured to use 'handover algorithm 2'.

I have incorporated one improvement from the fairwaves branch, which applies in a completely changed way on osmo-bsc master, but does the same thing to fix recovery from failed handover. (Even with two BTS in the same room and the phones barely 2m away, I sometimes see HO timing out.) https://gerrit.osmocom.org/5891 makes it so that a failed HO goes completely unnoticed instead of silencing the call.
I have asked about various other fairwaves patches, but there was a lot asked and it might take some time to get answers.

During practical testing, I often find congestion balancing failing, and traced that to the fact that for some reason one of the phones often fails to send measurement reports for the other cell I'm testing with. I have checked it down to the SIs going out to the DSP by looking at BTS GSMTAP. The neighbor cell ARFCNs are definitely sent out properly. It's not precisely the one phone or precisely the one cell exhibiting this, they seem to swap roles every so often, but it is common that the measurement reports for one lchan include rxlev of the other cell, while the reports on the lchan of the other cell simply omit neighbor cell rxlev. I have reduced transmitting power, but since the phones are so close to the cells, with rxlev not dropping below -70, I don't see how a phone could decide to omit the cell. Often I first get reports for both cells, and after a short while one of them stops getting neighbors reported. This is of course independent from the new handover algorithm. It puzzles me a bit. When there are sufficient reports for the other cell, congestion handovers work out well.

It might be good to write documentation on the handover parameters in the way of a "how to choose good values and adjust to certain situations"; which is of course a huge academic field, but an introduction would be good.

I have another half finished patch in the queue: the handover logging prints the subscriber_conn's mobile identity along most of the time, but since osmo-bsc doesn't pick up that identity except during paging, this is mostly still NULL ("subscr unknown") except for MT, where an lchan was created from Paging. My patch picks up the mobile identity like the DTAP filter we already have in osmo-bsc and populates the bsc_subscr with it, so that logging shows the actual subscriber. Will follow later.

Not implemented so far is notification of the MSC that intra-bsc handover has happened / of an inter-bsc handover request.

For the future, there is a huge field of features that come to mind, like automatic analysis of the handover logs and/or pcaps to produce visualizations, or to produce simulation data to run through a virtual testbed, so that one can pick a trace from a live network and analyse and optimise for that in a reproducible and controlled environment. There was talk of an external HO mechanism, which (I realize now) would allow managing HO across several BSS in one process/scheme/database -- which would enable load balancing across BSS borders as well.

I will not consider any of the ongoing features or tweaks to be part of this issue -- once the first patches are merged, we'll probably see a lot more

issues on HO.

Automatic tests are still part of this issue though, there's "only" jolly's test suite so far.

#52 - 01/19/2018 03:55 AM - neels

Ah, and still barely covered is handover decision in the presence of dynamic timeslots. They are bound to add a lot more complexity to the lchan choices in handover_decision_2.c

#53 - 01/25/2018 04:31 PM - laforge

- Status changed from In Progress to Stalled

#54 - 02/16/2018 04:08 AM - neels

- File failing_meas_reports_after_a_while.pcapng added

I have checked working handover (with limitations) with the patches now submitted at https://gerrit.osmocom.org/#/q/topic:ho2
Replied with some questions to the review posted so far.

limitations:
In my current test setup, measurement reports omit certain neighbor cells seemingly from the first handover that occured.
This might be due to hardware failure (two BTS may clock drift, and one of the phones has weird rx issues),
but the reproducability of the problem actually suggests that it is a software bug of sorts.

The reports themselves are indeed empty, wireshark says
".... ...1 11.. .... = NO-NCELL-M: Neighbour cell information not available for serving cell (7)"

When I start the core network, the BTSes register, and for a while (during a call) I get nice measurement reports for both of the neighbors.
After that the meas reports lack neighbors, hence no load-based HO occurs anymore.

I have calibrated one sysmobts to the other using the sysmobts-calib tool and setting the phy instance's clock-calibration value, without change.

I'm not sure how else to approach this, but it feels to me like there still is a problem hidden in the code somewhere.
We should try it with different BTS and phones, and possibly with virtual BTS tests, to see how reproducible the problem is.

The first empty meas report is in packet 9031 in attached pcap (rtp and mgw logging cut out from the trace for size)

#55 - 02/19/2018 02:26 PM - neels

Today pmaier helped me, and we saw the same behavior of measurement reports ceasing to be useful, with a completely other set of hardware: his laptop, his BTSes, two TEMS phones instead of the S4m. We saw measurement reports ceasing to be useful after the first handover; in other tests I have seen measurement reports also being broken from the start of a call, but doing a handover seems to be at least one of the triggers to break measurement reports.

This is preventing me from properly testing load distribution handover, but actually it's a separate issue. I am fairly sure now that it is a bug somewhere in our code, so creating another issue for it.

#56 - 02/19/2018 02:29 PM - neels

- Blocked by Bug #2963: Measurement Reports cease to be useful some time into a voice call / after handover (not sure which project has the bug / bugs yet) added

#57 - 02/19/2018 03:53 PM - neels

- Related to Bug #2964: handover_decision_2: measurement report number compared to lchan->meas_rep_last_seen_nr often overflows to zero (255 range) and hence an old report may be seen as new every once per 255 cycle added

#58 - 02/20/2018 06:30 PM - neels

The patches based on jolly's branch are now merged to osmo-bsc master.
The handover for load distribution part is only used with a config of 'handover algorithm 2'.
It seems to be working well at first glance, yet proper heavy testing of it isn't possible until we've resolved #2963.

#59 - 02/26/2018 12:07 AM - neels

- Related to Bug #3002: HO2: handover decision for dynamic timeslots added

#60 - 03/12/2018 02:09 AM - neels

- Status changed from Stalled to In Progress

yay! With the measurement reports problem fixed, I can finally test ho properly and get back to these remaining issues.

#61 - 03/28/2018 02:20 AM - neels

- Related to deleted (Feature #1749: add activation timeout in bsc_handover_start())

#62 - 05/17/2018 11:37 AM - laforge

neels wrote:

> yay! With the measurement reports problem fixed, I can finally test ho properly and get back to these remaining issues.

what's that status on this? It has been without update for two months, at 70% - but still in progress. Also: No checklist items checked. If "only" test are missing (which I consider actually one of the most important tasks) then either create sub-tickets for that, or rename the ticket title to "no tests for ...".

#63 - 05/18/2018 01:19 PM - neels

this is stalled by the inter-bsc handover efforts, because they are slightly higher prio IIUC.
Anyhow, I am super annoyed by the slow progress at inter-bsc ho, because it touches way more code than I had initially anticipated
(and this month I am constantly being distracted from work by endless private business, which doesn't help at all).

An aspect is that inter-bsc HO will cause refactorings also in the intra-bsc HO code, and having more HO tests would also help there.
I assume thus that I will naturally reach the need to add tests more or less soon.

But my focus is indeed currently more on the handover action, not so much the load-based decision aspect.
My head is already heavy with the sheer scope of it, I'm kind of reluctant to extend that right now... :/

But if you say load-based HO testing is more important at the moment I could shelve inter-BSC ho to test load-based ho.

#64 - 05/18/2018 01:22 PM - neels

- Checklist item deleted (automatic tests for load-based hand-over)
Checklist item deleted (jenkins/CI integration of automatic tests)
Checklist item [ ] load-based handover for dynamic timeslots added
Checklist item [x] osmo-bsc/tests/ regression tests in C for load-based ho added
Checklist item [ ] load-based HO tests in ttcn3 added
Checklist item [ ] load-based HO tests in osmo-gsm-tester added
Checklist item [x] implementation of load-based hand-over set to Done

#65 - 05/25/2018 10:19 AM - laforge

neels wrote:

> this is stalled by the inter-bsc handover efforts, because they are slightly higher prio IIUC.

then please mark it as "stalled" at the point where you're no longer working on it. Having it two months "in progress" without an update is not reflecting reality and can be misleading.

> An aspect is that inter-bsc HO will cause refactorings also in the intra-bsc HO code, and having more HO tests would also help there.
> I assume thus that I will naturally reach the need to add tests more or less soon.

Ok, so we agree that both the intra- as well as the inter-BSC HO will need proper test coverage before inter-BSC-HO can be merged without unknown risks of regressions.

> But if you say load-based HO testing is more important at the moment I could shelve inter-BSC ho to test load-based ho.

I'm just saying: Please make sure your tickets reflect the actual state. Thanks!

#66 - 05/25/2018 12:09 PM - neels

- Status changed from In Progress to Stalled

Stalled: waiting for inter-BSC HO refactorings

#67 - 08/20/2018 02:38 PM - neels

inter-BSC handover and related refactorings are merged to osmo-bsc master, hence this can carry on.

#68 - 08/20/2018 02:56 PM - neels

The status here is: the code is merged in handover decision algorithm 2, it most probably lacks proper handling in the presence of dynamic timeslots. This also needs testing, manually as well as in the form of test suites for ttcn3 and osmo-gsm-tester.

#69 - 08/20/2018 03:00 PM - neels

- Related to Support #3487: comprehensive documentation for Handover configuration added

#70 - 08/20/2018 03:01 PM - neels

- Related to Feature #1608: various handover improvements, meta-issue added

#71 - 10/02/2018 06:52 PM - neels

- Checklist item [ ] inter-BSC: do load-based HO to neighbor-BSS cells? added

#72 - 10/10/2018 11:27 AM - neels

- Checklist item deleted (inter-BSC: do load-based HO to neighbor-BSS cells?)

- Status changed from Stalled to In Progress

- % Done changed from 70 to 80

load-based inter-BSC HO: I'm at the point where handover_decision_2.c handles remote-BSS neighbors properly,
which I'd suggest is all that we implement for now. There's also #3638 as a proper separate issue for this item.
(patches not merged yet)

#73 - 10/10/2018 11:32 AM - neels

- Checklist item deleted (load-based handover for dynamic timeslots)

dynamic timeslots have a separate issue: #3002

#74 - 10/10/2018 11:32 AM - neels

- Related to Feature #3638: handover decision 2: load balancing across BSS added

#75 - 10/15/2018 10:04 PM - neels

- Related to Feature #3656: inter-BSC handover outgoing: compose Cell Identifier List from several ARFCN+BSIC added

#76 - 10/15/2018 10:05 PM - neels

Maybe instead of adding these issue relations I should use a "Handover" tag...?

#77 - 10/15/2018 10:10 PM - neels

- Related to deleted (Feature #3656: inter-BSC handover outgoing: compose Cell Identifier List from several ARFCN+BSIC)

#78 - 10/15/2018 10:10 PM - neels

neels wrote:

> Maybe instead of adding these issue relations I should use a "Handover" tag...?

I meant to say this at #1608...

#79 - 07/18/2019 05:13 AM - laforge

- Priority changed from Urgent to High

#80 - 09/04/2019 09:10 AM - laforge

- Related to Feature #4189: TTCN-3 tests for load-based hand-over in BSC added

#81 - 09/04/2019 09:11 AM - laforge

- Checklist item deleted (load-based HO tests in ttcn3)

- Status changed from In Progress to Resolved

Files

failing_meas_reports_after_a_while.pcapng	2.55 MB	02/16/2018	neels