

## OsmoBSC - Bug #3503

### osmo-bsc: codec-list VTY cfg implementation not filtering correctly

08/27/2018 12:44 PM - pespin

<b>Status:</b>	Resolved	<b>Start date:</b>	08/27/2018
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	dexter	<b>% Done:</b>	100%
<b>Category:</b>			
<b>Target version:</b>			
<b>Spec Reference:</b>			

#### Description

Found while running following scenario in osmo-gsm-tester: -s voice:trx-b200+mod-bts0-ts-tchh+cfg-codec-fr1 -T -l dbg

So basically we set all timeslots to TCH/H, and we set in osmo-bsc.cfg "codec-list fr1". Which means any phone call should fail because fr1 requires a TCH/H (or dynamic) channel. But instead of failing, code in lchan\_select.c decides to pick a TCH/H channel to serve the call and the call succeeds, which is wrong, because that would mean using "hr1" which is not accepted in "codec-list".

```
20180827132722402 DLSCCP <0020> sccp_scoc.c:1581 SCCP-SCOC(4) [0x612000006820]{ACTIVE}: Received Event RCOC-DT1.ind
20180827132722402 DLSCCP <0020> sccp_user.c:156 Delivering N-DATA.indication to SCCP User 'msc-0'
20180827132722402 DMSC <0007> osmo_bsc_sigtran.c:238 N-DATA.ind(4, 00 20 01 0b 08 01 0a c2 a1 91 81 a5 05 7c 06 0a 2a 2a 03 0f a2 7d 0b 89 01 83 ff 57 82 80 84 3f 07 81 )
20180827132722402 DMSC <0007> a_reset.c:204 A-RESET(msc-0) [0x612000009e20]{CONN}: Received Event EV_N_CONNECT
20180827132722403 DMSC <0007> osmo_bsc_bssap.c:855 Rx MSC DT1 BSSMAP ASSIGNMENT REQ
20180827132722403 DMSC <0007> osmo_bsc_bssap.c:726 Found matching audio type: full rate SPEECH_V1 for channel_type = { ch_indctr=0x1, ch_rate_type=0xa, perm_spch=[ 42 21 11 01 25 05 ] }
20180827132722403 DMSC <0007> osmo_bsc_bssap.c:757 SUBSCR_CONN(conn4) [0x612000006b20]{ACTIVE}: Received Event ASSIGNMENT_START
20180827132722403 DMSC <0007> bsc_subscr_conn_fsm.c:342 SUBSCR_CONN(conn4) [0x612000006b20]{ACTIVE}: state_chg to ASSIGNMENT
20180827132722403 DMSC <0007> assignment_fsm.c:304 Assignment: incrementing rate counter: assignment:attempted Assignment attempts.
20180827132722403 DAS <0012> fsm.c:299 assignment(conn4) [0x6120000066a0]{WAIT_LCHAN_ACTIVE}: Allocated
20180827132722403 DAS <0012> fsm.c:329 assignment(conn4) [0x6120000066a0]{WAIT_LCHAN_ACTIVE}: is child of SUBSCR_CONN(conn4) [0x612000006b20]
20180827132722403 DRLL <0000> lchan_select.c:159 (bts=0) lchan_select_by_type(TCH_F)
20180827132722403 DRLL <0000> lchan_select.c:71 looking for lchan TCH/F: (bts=0,trx=0,ts=0,pchan=CCH+SDCCH4,state=IN_USE) is != TCH/F
20180827132722403 DRLL <0000> lchan_select.c:71 looking for lchan TCH/F: (bts=0,trx=0,ts=1,pchan=SDCCH8,state=UNUSED) is != TCH/F
20180827132722403 DRLL <0000> lchan_select.c:71 looking for lchan TCH/F: (bts=0,trx=0,ts=2,pchan=TCH/H,state=UNUSED) is != TCH/F
20180827132722403 DRLL <0000> lchan_select.c:71 looking for lchan TCH/F: (bts=0,trx=0,ts=3,pchan=TCH/H,state=UNUSED) is != TCH/F
20180827132722403 DRLL <0000> lchan_select.c:71 looking for lchan TCH/F: (bts=0,trx=0,ts=4,pchan=TCH/H,state=UNUSED) is != TCH/F
20180827132722403 DRLL <0000> lchan_select.c:71 looking for lchan TCH/F: (bts=0,trx=0,ts=5,pchan=TCH/H,state=UNUSED) is != TCH/F
20180827132722403 DRLL <0000> lchan_select.c:71 looking for lchan TCH/F: (bts=0,trx=0,ts=6,pchan=TCH/H,state=UNUSED) is != TCH/F
20180827132722403 DRLL <0000> lchan_select.c:71 looking for lchan TCH/F: (bts=0,trx=0,ts=7,pchan=TCH/H,state=UNUSED) is != TCH/F
20180827132722404 DRLL <0000> lchan_select.c:71 looking for lchan TCH/H: (bts=0,trx=0,ts=0,pchan=CCH+SDCCH4,state=IN_USE) is != TCH/H
20180827132722404 DRLL <0000> lchan_select.c:71 looking for lchan TCH/H: (bts=0,trx=0,ts=1,pchan=SDCCH8,state=UNUSED) is != TCH/H
20180827132722404 DRLL <0000> lchan_select.c:86 looking for lchan TCH/H: (bts=0,trx=0,ts=2,pchan=TCH/H,state=UNUSED) ss=0 is available
20180827132722404 DCHAN <0010> lchan_select.c:253 lchan(0-0-2-TCH_H-0) [0x612000008aa0]{UNUSED}: (type=TCH_H) Selected
```

```

20180827132722404 DAS <0012> assignment_fsm.c:372 assignment(conn4_0-0-2-TCH_H-0) [0x6120000066a0]{
WAIT_LCHAN_ACTIVE}: (bts=0,trx=0,ts=2,ss=0) Starting Assignment: chan_mode=SPEECH_V1, full_rate=1,
aoip=yes MSC-rtp=10.42.42.3:4002
20180827132722404 DAS <0012> assignment_fsm.c:374 assignment(conn4_0-0-2-TCH_H-0) [0x6120000066a0]{
WAIT_LCHAN_ACTIVE}: state_chg to WAIT_LCHAN_ACTIVE
20180827132722404 DCHAN <0010> lchan_fsm.c:301 lchan(0-0-2-TCH_H-0) [0x612000008aa0]{UNUSED}: Recei
ved Event LCHAN_EV_ACTIVATE
20180827132722404 DCHAN <0010> lchan_fsm.c:475 lchan(0-0-2-TCH_H-0) [0x612000008aa0]{UNUSED}: state
_chg to WAIT_TS_READY
20180827132722404 DCHAN <0010> lchan_fsm.c:501 lchan(0-0-2-TCH_H-0) [0x612000008aa0]{WAIT_TS_READY}
: (type=TCH_H) Activation requested: FOR_ASSIGNMENT voice=yes MGW-ci=new type=TCH_H tch-mode=SPEEC
H_V1
20180827132722404 DTS <0011> lchan_fsm.c:505 timeslot(0-0-2-TCH_H) [0x61200000aea0]{UNUSED}: Receiv
ed Event TS_EV_LCHAN_REQUESTED
20180827132722404 DTS <0011> timeslot_fsm.c:319 timeslot(0-0-2-TCH_H) [0x61200000aea0]{UNUSED}: sta
te_chg to IN_USE
20180827132722404 DCHAN <0010> timeslot_fsm.c:104 lchan(0-0-2-TCH_H-0) [0x612000008aa0]{WAIT_TS_REA
DY}: Received Event LCHAN_EV_TS_READY
20180827132722404 DCHAN <0010> lchan_fsm.c:519 lchan(0-0-2-TCH_H-0) [0x612000008aa0]{WAIT_TS_READY}
: state_chg to WAIT_ACTIV_ACK
20180827132722404 DRSL <0003> abis_rsl.c:475 (bts=0,trx=0,ts=2,pchan=TCH/H,state=IN_USE) Tx RSL Ch
annel Activate with act_type=INTRA_NORM_ASS
20180827132722404 DCHAN <0010> fsm.c:299 lchan_rtp(0-0-2-TCH_H-0) [0x612000006520]{WAIT_MGW_ENDPOIN
T_AVAILABLE}: Allocated
20180827132722404 DCHAN <0010> fsm.c:329 lchan_rtp(0-0-2-TCH_H-0) [0x612000006520]{WAIT_MGW_ENDPOIN
T_AVAILABLE}: is child of lchan(0-0-2-TCH_H-0) [0x612000008aa0]
20180827132722404 DCHAN <0010> lchan_rtp_fsm.c:117 lchan_rtp(0-0-2-TCH_H-0) [0x612000006520]{WAIT_M
GW_ENDPOINT_AVAILABLE}: state_chg to WAIT_MGW_ENDPOINT_AVAILABLE

```

I attach a full osmo-gsm-tester with that test to have the complete information, but the above log of osmo-bsc should be enough.

Some more related information from IRC discussion:

```

first it says "looking for lchan TCH/F", then "looking for lchan TCH/H". Actually lchan_select_by_
type() does fall back to TCH/H
in practice it makes sense if an entity asks for TCH/F but can also manage TCH/H. Historically lch
an_select() (formerly called chan_alloc()) was the place to decide such, but yeah, I think that's
the wrong place.
lchan_select_by_type() should return "no", and then the caller should decide whether to also try T
CH/H
so in lchan_select_by_type(), we historically since always fall back to TCH/H if TCH/F is not avai
lable
but before we do so, the code should check whether it even wants to permit that
and this check doesn't belong in lchan_select_by_type() but in the caller
I think it's a loophole left over from before the time when we started strictly checking the codec
config
I think lchan_select_by_type() should do exactly what it says, select exactly that type, and the c
allers should iterate through permitted types in order of preference
falling back from TCH/H to TCH/F is probably fine, but still, the caller might then pick a half ra
te codec on TCH/F even though a full-rate codec might be permitted
yes, picking dynamic TS is the right place there, but changing TCH kinds is not

```

#### Related issues:

Related to OsmoBSC - Bug #3637: handover decision 2: properly check available...	<b>Feedback</b>	<b>10/09/2018</b>
Related to OsmoBSC - Bug #3645: internal handover: when handover changes the ...	<b>Resolved</b>	<b>10/11/2018</b>

#### History

**#1 - 09/30/2018 10:47 AM - laforge**

- Assignee set to dexter

I know [dexter](#) has been working in this area recently. In any case, it needs to be fixed, if it's still present.

**#2 - 10/09/2018 04:28 PM - dexter**

I have now analyzed the problem and I think that lchan\_select.c is not behaving correctly.

Basically what happens is the following: The MSC offers a lot of unsolicited codecs, which I think is not exactly forbidden, but it could have respected what we already told it in the L3 complete message. However, the logic that selects the codec does select FR1, which is exactly what we configured (BTS implicitly has no codec limitation configured, the MSC is restricted to FR1). Until that point everything works.

But when the BSC is looking for an lchan the function lchan\_select\_by\_type() does not find a TCH/F (since we only configured TCH/H). But instead of failing the function tries again with TCH/H, and succeeds. This leads into a TCH/H lchan to be selected and eventually leads into the wired ASSIGNMENT COMPLETE that happily tells the MSC that a GSM-HR has been selected as codec, even though this codec has never been allowed by the configuration.

I think The function lchan\_select\_by\_type() should not implement such a fallback. The fallback is definitely intended as there is a comment /\* If we don't have TCH/F available, fall-back to TCH/H \*/ in there, but I think this should not be. Lower layers must not jump over what the higher layers have negotiated.

Then we also should add some lchan scanning to codec\_pref.c. It should scan through the lchan configuration and see if there is indeed an lchan for the codec that is selected.

### #3 - 10/10/2018 10:57 AM - neels

On Tue, Oct 09, 2018 at 04:28:33PM +0000, dexter [REDMINE] wrote:

But when the BSC is looking for an lchan the function lchan\_select\_by\_type() does not find a TCH/F (since we only configured TCH/H). But instead of failing the function tries again with TCH/H, and succeeds. This leads into a TCH/H lchan to be selected and eventually leads into the wired ASSIGNMENT COMPLETE that happily tells the MSC that a GSM-HR has been selected as codec, even though this codec has never been allowed by the configuration.

That fall-back code has been there "from the start" of openbsc, from before we had any codec matching.

**Something** has to do a fallback, though, in case multiple codecs are permitted and the first one isn't available.

Then we also should add some lchan scanning to codec\_pref.c. It should scan through the lchan configuration and see if there is indeed an lchan for the codec that is selected.

exactly. Iterate the allowed codecs in order of preference, and call lchan\_select() for each one; remove the fall-back to other voice codecs from lchan\_select().

This will be needed by assignment\_fsm.c and handover\_fsm.c, so it might as well be a wrapper like lchan\_select\_by\_codec\_prefs(ms\_codecs, bts\_codecs, msc\_codecs) or something, in lchan\_select.c, called by both. Or maybe rather add the codec pref args to lchan\_select\_by\_chan\_mode() and do the matching in there?

The RSL Chan Required / Immediate Assignment code already does that, it falls back to TCH/\* if lchan\_select\_by\_type() has no SDCCH left. See abis\_rsl.c rsl\_rx\_chan\_rqrd().

### #4 - 10/10/2018 11:38 AM - neels

- Related to Bug #3637: handover decision 2: properly check available codecs added

### #5 - 10/10/2018 12:03 PM - neels

neels wrote:

exactly. Iterate the allowed codecs in order of preference, and call lchan\_select() for each one [...]

This will be needed by assignment\_fsm.c and handover\_fsm.c

In assignment\_fsm.c, we want to pick one from a list of available configs. (It's up to the user to configure codecs so that we don't need transcoding support...)

In handover\_fsm.c, we currently want one specific existing lchan config that should be matched.

If the codec changes we're currently still broken in two ways: a) lack of transcoding b) no message to tell the MSC about it.

But ultimately, as soon as we support transcoding, we wouldn't mind to change the channel / codec type during handover. So basically, we can already implement towards the way Assignment does things: until transcoding is here, it is up to the user to configure so that codecs never mismatch, and we might as well pick any supported codec now; maybe mildly prefer to stay on the same codec the old lchan is on. When support for transcoding

is added, users can start configuring more freely.

- internal HO: so far we have an lchan with a {type,mode,mr\_cfg} and want to find an equivalent one.  
So far we just lchan\_select\_by\_type(), i.e. we're happy if the TCH/x kind matches, ignoring AMR stuff and codecs.  
So that needs to be less naive.
- incoming HO from remote BSS: we have a BSSMAP Handover Request message, with a Channel Type IE as well as Codec List (MSC Preferred).  
So far we do match\_codec\_pref(), apparently already heeding the request's IEs and AMR bits.  
Then we do lchan\_select\_by\_chan\_mode(). Looks ok?

#### #6 - 10/10/2018 12:21 PM - dexter

- Status changed from New to In Progress

- % Done changed from 0 to 50

I have now added checks that make sure a suitable pchan is available on the BTS. However we still may run into the odd situation that a wrong channel rate is used because that is problem is not addressed yet. But at least we are now preventing flawed configurations that can not work.

(neels thanks for your input. I will go through that later but I see that we indeed need the fallback, but we only need it does not contradict wat the MSC allowed with the ASSIGNMENT REQUEST)

See also:

<https://gerrit.osmocom.org/#/c/osmo-bsc/+/11295> codec\_pref: cosmetic: seperate half/full rate determination

<https://gerrit.osmocom.org/#/c/osmo-bsc/+/11296> codec\_pref: also check physical channels

#### #7 - 10/11/2018 02:10 PM - neels

- Related to Bug #3645: internal handover: when handover changes the speech codec, it should notify the MSC with BSSMAP Handover Performed added

#### #8 - 10/29/2018 01:32 PM - dexter

Since the logic that checks the configuration the bug got more difficult to reproduce. The only situation where a wrong channel assignment now can happen is when for example all TCH/F are occupied, but at least one TCH/H is still free. When then a TCH/F is requested in the assignment request, the BSC will assign a TCH/H. We also would see that in the assignment complete. I am working on way to test this.

#### #9 - 01/10/2019 05:59 PM - dexter

I have now analyzed the problem further. The channel allocator fails the assignment in the following situations:

- All FR channels full, but Assignment permits FR and HR (We should get at least an HR channel)
- All HR channels full, but Assignment permits FR and HR (We should get at least an FR channel)

When all HR channels are full and we purely ask for an HR channel we get an assignment complete for a FR channel.

I have now TTCN3 test cases and VTY debug commands that pinpoint those corner case situations. I will push the stuff for review when I have polished it a bit.

#### #10 - 01/15/2019 05:54 PM - dexter

I have added two more tests to check the if the BSC returns with the preferred rate (FR or HR). When both, FR and HR is requested and FR is preferred, then FR is returned. For the case where FR and HR is requested and HR is preferred, FR is returned as well, which is wrong.

I managed to pinpoint one part of the problem, but not everything yet. 3 of 10 tests still are still failing.

The current state can be found on pmaier/challoc (osmo-ttcn3-hacks and osmo-bsc)

#### #11 - 01/16/2019 04:17 PM - dexter

- % Done changed from 50 to 80

I think I have a good understanding now of what is going wrong at the BSC and I am now working on a solution. Actually the way the codec is currently selected does not honor the fact that an assignment command may ask for HR and FR (with a preference) at the same time, nor does decision include the current channel load.

#### #12 - 01/17/2019 05:34 PM - dexter

- % Done changed from 80 to 90

The channel allocator problems seem to be fixed now. All the TTCN3 I have set up to address the problem do now pass. The patches and the TTCN3 tests are now in Gerrit:

<https://gerrit.osmocom.org/#/c/osmo-bsc/+12621> chan\_alloc: remove references to lchan\_alloc()  
<https://gerrit.osmocom.org/#/c/osmo-bsc/+12622> lchan\_select: dont allow half rate EFR to be selected  
<https://gerrit.osmocom.org/#/c/osmo-bsc/+12623> lchan\_select: Do not unsolicitedly select a TCH/F  
<https://gerrit.osmocom.org/#/c/osmo-bsc/+12624> bsc\_vty: add features to shun specific lchans  
<https://gerrit.osmocom.org/#/c/osmo-bsc/+12625> assignment\_fsm: fix channel allocator preferences  
<https://gerrit.osmocom.org/#/c/osmo-ttcn3-hacks/+12620/> WIP: BSC\_Tests: Add tests to check channel allocator

#### #13 - 01/28/2019 03:50 PM - dexter

The channel allocator fix and the necessary VTY changes are still under review. I have discussed with Neels recently how we can simplify the lchan disable/enable functionality and we came across the idea that it is actually very easy to send the lchan fsm from unused to broken state and vice versa. I am using this method now and the tests still behave as expected. I haven't pushed that to Gerrit yet since there still some review issues left.

#### #14 - 01/30/2019 09:37 AM - dexter

I am now done with the review issues. The patches can be found under:

<https://gerrit.osmocom.org/#/c/osmo-bsc/+12624> bsc\_vty: add features to disable specific lchans via vty  
<https://gerrit.osmocom.org/#/c/osmo-bsc/+12625> assignment\_fsm: fix channel allocator preferences  
<https://gerrit.osmocom.org/#/c/osmo-bsc/+12679> bsc\_vty: add vty command to display all lchans

The VTY patches are split up into two separate ones. The changes that just show lchan states for debugging are now separate. Also the lchan blocking is now done by using the broken state. Doing it that way requires no changes to the lchan FSM, its just the VTY that puts the lchan FSM from unused to broken state and vice versa.

#### #15 - 01/30/2019 09:39 AM - dexter

Since the VTY syntax for switching between broken and unused had to be slightly modified in order to be more expressive on what actually happens the related TTCN3 tests also need some adjustment:

<https://gerrit.osmocom.org/#/c/osmo-ttcn3-hacks/+12728> BSC\_Tests: fix vty interface for channel allocator tests

#### #16 - 02/14/2019 11:28 AM - laforge

#### #17 - 02/25/2019 11:16 AM - dexter

The patch is now merged, but unfortunately we now see the following two testcases failing:

TC\_assignment\_codec\_amr\_f  
TC\_assignment\_codec\_amr\_h

The reason for this is that S15-S0 are incorrectly stored and therefore the value is lost when the ASSIGNMENT complete is generated. The following patch fixes this:

<https://gerrit.osmocom.org/#/c/osmo-bsc/+13039> assignment\_fsm: use activate.info.s15\_s0 for ASS. COMPL.

#### #18 - 03/05/2019 08:57 AM - dexter

Unfortunately I am still a bit confused where to store the s15\_s0 bits I will discuss with Neels as soon as possible how the problem can be resolved.

The S0-S15 bits are pretty much the only thing that has to be stored for longer. All other parameters are deducted from the current lchan parameters.

The speech codec IE is generated from the following parameters:  
lchan->type, lchan->tch\_mode and conn->lchan->activate.info.s15\_s0

conn->lchan->activate.info.s15\_s0 is the problematic one, it needs a different storage position. I wonder if it could be lchan->s15\_s0 since the S-bits are a normal property of the lchan like the type or the tch\_mode? This is what needs to be discussed.

Currently the storing of the s15\_s0 happens somewhere where the value gets overwritten or not copied when the lchan changes so we get a constant 0x0000 here. This is also the reason why TC\_assignment\_codec\_amr\_f and TC\_assignment\_codec\_amr\_h are currently not passing.

#### #19 - 03/11/2019 05:47 PM - dexter

- Status changed from In Progress to Resolved

- % Done changed from 90 to 100

The problem is fixed now, however, the storage location of S15-S0 is not correct yet. Since the scope of this task is solved now and the related tests are all passing I have crated a new task for this. See also: [#3833](#)

## Files

---

run.tar.gz	256 KB	08/27/2018	pespin
------------	--------	------------	--------