OsmoBSC - Bug #3529

osmo-bsc does not send the correct S0-S15 bits for AMR in the Assignment Compl Message.

09/06/2018 02:56 PM - dexter

Status:	Resolved	**Start date:**	09/06/2018
Priority:	Normal	**Due date:**	
Assignee:	dexter	**% Done:**	100%
Category:			
Target version:			
Spec Reference:			

Description

AMR supports a variety of fine rates and settings. Those are set via S0-S15 in the speech codec list that is sent by the MSC to the BSC. the BSC may then choose a configuration and return it back via the ASSIGNMENT COMPLETE message.

In the documented case the S0-S15 were set to 0x0200, but we return 0xFF57. This looks problematic and when checking the source code one can see in assignment_fsm.c:send_assignment_complete() that the speech codec element is computed using gsm0808_speech_codec_from_chan_type(). This function only has perm_spc as input which is an integer and basically only sets the codec type and S0-S15 are populated with standard values. This is presumably not enough, we should take the S0-S15 from the speech codec list in the ASSIGNMENT REQUEST and work with it.

Related issues:

Related to OsmoBSC - Bug #3637: handover decision 2: properly check available...	**Feedback**	**10/09/2018**

History

#1 - 09/21/2018 04:04 PM - dexter

- File 01_COMPLETE_LAYER_3_INFO.png added

- File 02_ASSIGNMENT_REQUEST.png added

- File 03_ASSIGNMENT_COMPLETE.png added

- Status changed from New to In Progress

- % Done changed from 0 to 70

The S0-S15 for AMR are no longer hardcodec in osmo-bsc. We now take the configuration bits from the ASSIGNMENT REQUEST and do an intersection between our BSS capabilities. This intersected bits are then returned with the ASSIGNMENT COMPLETE message in speech codec (choosen).

Attached one finds the wireshark screenshots of the speech codec data as they look now. One can see, we first advertise a variety of options, then the MSC limits the options and we agree. However, in this particular example the 0x0200 looks suspicious. I checked the bit ordering against the spec multiple times and I am confident that our endieness is correct, but 0x0200 does not make much sense to me. We probably need to discuss this next week.

See also: https://gerrit.osmocom.org/#/c/osmo-bsc/+/11060 codec_pref: handle S0-S15 in ASSIGNMENT REQUEST

#2 - 09/25/2018 09:51 AM - dexter

Note: https://gerrit.osmocom.org/#/c/osmo-bsc/+/11060/ is still in review, TC_assignment_codec_amr_f and TC_assignment_codec_amr_h will pass when the patch is merged.

#3 - 09/25/2018 02:30 PM - dexter

I have now fixed the remaining bits. So far the result of S0-S15 was not used in lchan_fsm.c, instead lchan_fsm was accessing the mr configuration directly. Now the negotiated S0-S15 are used. Unfortunately that required an additional helper function in libosomocore, see also:

https://gerrit.osmocom.org/#/c/osmo-bsc/+/11060/
https://gerrit.osmocom.org/#/c/libosmocore/+/11082/

#4 - 09/25/2018 03:40 PM - dexter

- File amr_ass_fail.pcapng added

Unfortunately since we now forward the negotiated AMR configuration to RSL the BTS we get a Channel Activation Nack. I can not really see why. All that is different from before is that we select 5,15 and 5,90 instead of 5,90 only. I can not see why this should be rejected. I always thought that we could select up to four rates, so there should be room for even more selections. Attached one finds the trace. I am a bit confused at that point.

#5 - 09/25/2018 04:00 PM - laforge

On Tue, Sep 25, 2018 at 03:40:13PM +0000, dexter [REDMINE] wrote:

> I always thought that we could select up to four rates, so there should be room for even more selections. Attached one finds the trace. I am a bit confused at that point.

according to spec, yes. Whether OsmoBTS actually supports this, you have to check the code.

#6 - 09/27/2018 02:19 PM - dexter

- % Done changed from 70 to 80

I have checked the code on osmo-bts and the parser rejects the IE early, which is correct since the IE we are sending is not spec compliant. The problem is that when more then one rate is in the active set, one will have to specify hysteresis and threshold values for each of the rates. Unfortunately struct gsm48_multi_rate_conf is not specified this way at the moment and on top of that the layout of the data will be different for each of the three cases. See also 3GPP TS 04.08, 10.5.2.21aa.

As a quick solution I now mask out all rates except the highest. This will give us a spec compliant IE, but we lack the lower codec rates from the active set. We probably should deal with the problem in another patch in order to get this one through quickly.

#7 - 09/27/2018 04:49 PM - dexter

There is the open question from where exactly the codec rate parameters
currently come and where the VTY settings for threshold and hysteresis end up.

The configuration in the lchan is set in lchan_fsm.c by the function
lchan_mr_config() The only input parameter this function gets is of type
struct gsm48_multi_rate_conf.

The function populates two fields, lchan->mr_ms_lv and lchan->mr_bts_lv, with
the same data. Both fields are raw data and will also contain the length field
of the TLV struct.

The first operation is to set the TLV fields length fields in the first byte of mr_ms_lv and mr_bts_lv. The pointers are of type struct gsm48_multi_rate_conf, so this will be constantly 2 (see also the definition of gsm48_multi_rate_conf in gsm_04_08.h of libosmocore).

What follows is some pointer casting and the assignment of the struct contents. The input struct is not copied. Only the rate flags are picked. ver and icmi will be constantly set to 1. During the struct population there is even some filtering that makes sure .m10_2 and .m12_2 are 0 when HR is used.

lchan_mr_config() gets its input from &info->for_conn->sccp.msc->amr_conf. This is the configuration that from vty:

```
msc 0
 amr-config 12_2k forbidden
 amr-config 10_2k forbidden
 amr-config 7_95k forbidden
 amr-config 7_40k forbidden
 amr-config 6_70k forbidden
 amr-config 5_90k allowed
 amr-config 5_15k forbidden
 amr-config 4_75k forbidden
```

Apparently there is no filtering done. When two codec rates are set to allowed it fails.

In gsm_04_08_rr.c there is the function gsm48_multirate_config() which seems to be able to render a complete version of a multi rate configuration. This function is called when VTY settings on hysteresis and threshold are made.

bsc_vty.c:
gsm48_multirate_config(lchan->mr_ms_lv, &mr, mr.ms_mode)
gsm48_multirate_config(lchan->mr_bts_lv, &mr, mr.bts_mode)

Now lchan_mr_config() also writes to lchan->mr_ms_lv and lchan->mr_bts_lv, which seems to be not right.

To me this looks like having the VTY generating the bitfields that are sent via RSL is an older method. Now we have a more dynmic way to generate the active set (S0-S15). I think we need to record the info from the VTY and then use that as input to regenerate lchan->mr_ms_lv and lchan->mr_bts_lv depending on the current situation of the call. So regenerating the bitfield in lchan_mr_config() is right, but we need to include the VTY settings. Those are mainly threshold and hysteresis.

All in all, since we already have the confusion about lchan->mr_bts_lv and lchan->mr_ms_lv in current master, we wouldn't loose anything when we merge I2d8ded51b3eb4c003fe2da6f2d6f48d001b73737, but still this is something that needs to be fixed.

#8 - 09/28/2018 07:54 AM - dexter

When looking in bsc_vty.c at function lchan_set_single_amr_mode(), which is is the only caller of gsm48_multirate_config() one can see that we apparently never used more then one codec rate at a time, even though gsm48_multirate_config() should be capable of encoding multiple rates.

I need to read more into bsc_vty.c to find out how exactly the configuration of the codec rates works.

Basically we have the following sources that make up the final codec rate configuration:

- codec list in ASSIGNMENT REQUEST
- local configuration set via VTY (multiple, per BTS and per MSC)

We need to make sure that the code is SCCP-lite sympatic, which basically means that we can only rely on what we have configured in the VTY. Since the underlying logic relies on S0-S15 the value has to be computed, regardless if we are on AoIP or SCCP-lite. At the moment this is the case. match_codec_pref() will always compute chan_mode, full_rate and s15_s0.

#9 - 09/28/2018 10:41 AM - dexter

Looks like my assumption about lchan_set_single_amr_mode() and gsm48_multirate_config() is not correct. lchan_set_single_amr_mode() is called by lchan_act_cmd, so it is a helper function that is used for debugging but has nothing to do with the regular operation.

#10 - 09/28/2018 11:14 AM - dexter

I have now analyzed the VTY options in more detail to find out what struct members are actually set. Here are some of my thoughts. Actually it looks less problematic than than it looked to me yesterday.

Relevant structs

libosmocore: gsm_04_08.h

```
/* Chapter 10.5.2.21aa */
struct gsm48_multi_rate_conf {
    uint8_t smod : 2,
         spare: 1,
         icmi : 1,
         nscb : 1,
         ver : 3;
    uint8_t m4_75 : 1,
         m5_15 : 1,
         m5_90 : 1,
         m6_70 : 1,
         m7_40 : 1,
         m7_95 : 1,
         m10_2 : 1,
         m12_2 : 1;
} __attribute__((packed));
```

osmo-bsc: gsm_data.h

```
struct amr_multirate_conf {
    uint8_t gsm48_ie[2]; //data layout as in struct gsm48_multi_rate_conf
```

```
    struct amr_mode ms_mode[4];
    struct amr_mode bts_mode[4];
    uint8_t num_modes; //never used!
};

struct amr_mode {
    uint8_t mode;
    uint8_t threshold;
    uint8_t hysteresis;
};
```

Currently used by codec_pref.c
struct gsm48_multi_rate_conf: bts->mr_full->gsm48_ie
struct gsm48_multi_rate_conf: bts->mr_half->gsm48_ie
struct gsm48_multi_rate_conf: msc->amr_conf Set in osmo_bsc_vty.c
struct gsm48_multi_rate_conf: msc->amr_conf->m12_2
struct gsm48_multi_rate_conf: msc->amr_conf->m10_2
struct gsm48_multi_rate_conf: msc->amr_conf->m7_95
struct gsm48_multi_rate_conf: msc->amr_conf->m7_40
struct gsm48_multi_rate_conf: msc->amr_conf->m6_70
struct gsm48_multi_rate_conf: msc->amr_conf->m5_90
struct gsm48_multi_rate_conf: msc->amr_conf->m5_15
struct gsm48_multi_rate_conf: msc->amr_conf->m4_75

```
msc 0
 amr-config 12_2k forbidden
 amr-config 10_2k forbidden
 amr-config 7_95k forbidden
 amr-config 7_40k forbidden
 amr-config 6_70k forbidden
 amr-config 5_90k allowed
 amr-config 5_15k forbidden
 amr-config 4_75k forbidden
```

VTY sets rate members in msc->amr_conf. Other flags are apparantly not set

Set in bsc_vty.c

struct gsm48_multi_rate_conf: bts->mr_full|half->gsm48_ie->m12_2
struct gsm48_multi_rate_conf: bts->mr_full|half->gsm48_ie->m10_2
struct gsm48_multi_rate_conf: bts->mr_full|half->gsm48_ie->m7_95
struct gsm48_multi_rate_conf: bts->mr_full|half->gsm48_ie->m7_40
struct gsm48_multi_rate_conf: bts->mr_full|half->gsm48_ie->m6_70
struct gsm48_multi_rate_conf: bts->mr_full|half->gsm48_ie->m5_90
struct gsm48_multi_rate_conf: bts->mr_full|half->gsm48_ie->m5_15
struct gsm48_multi_rate_conf: bts->mr_full|half->gsm48_ie->m4_75
(VTY only allows to set up to 4 flags)

struct gsm48_multi_rate_conf: bts->mr_full|half->gsm48_ie->icmi
struct gsm48_multi_rate_conf: bts->mr_full|half->gsm48_ie->smod

struct amr_multirate_conf: bts->mr_full|half.ms_mode[0..3].mode (unclear, set by VTY, but never accessed?)
struct amr_multirate_conf: bts->mr_full|half.ms_mode[0..3].threshold
struct amr_multirate_conf: bts->mr_full|half.bts_mode[0..3].hysteresis

```
network
 bts 0
  amr tch-f modes 1 2 6 7
  amr tch-f threshold ms 0 0 0
  amr tch-f hysteresis ms 0 0 0
  amr tch-f threshold bts 0 0 0
  amr tch-f hysteresis bts 0 0 0
  amr tch-f start-mode auto
  amr tch-h modes 1 2 4 5
  amr tch-h threshold ms 0 0 0
  amr tch-h hysteresis ms 0 0 0
  amr tch-h threshold bts 0 0 0
  amr tch-h hysteresis bts 0 0 0
  amr tch-h start-mode auto
```

Ideas and analysis

In the current master we only use the rate flags set by osmo_bsc_vty.c (info->for_conn->sccp.msc->amr_conf) in the lchan. All other settings are ignored. At least as far as the lchan is concerned. For example settin threshold and hysteresis will have no effect on the lchan.

At the moment the bts specific settings look completely broken. The codec flags are not of little use. At the moment the current master uses them to determine the S0-S15 for the speech codec list in COMPLETE L3 INFO. But the values are never transfered via RSL to the BTS.

Since the VTY only allows up to 4 modes to be configured the bts specific settings were probably intended to model the active set of possible codecs, since via SCCP-Lite we are unable to specify the active it makes sense to be able to configure this statically. We now need to find a compatible way to use this with AoIP.

In AoIP we will must be able to configure all possible coec rates. Thats 8 for FR and 6 for HR. When using SCCP-Lite we don't get S0-S15 via the ASSIGNMENT REQUEST, so we implicitly assume that all codec rates should be permitted by the MSC. So in this situation we should not specify more then 4 codec rates in VTY because all coec rates we specifyed will become our active set.

Pseudo S0-S15 from ASSIGNMENT REQUEST (all codec rates set)
VTY settings in msc->amr_conf
VTY settings in bts->mr_full|half (not more then 4 codec rates)
Intersection: lchan->mr_ms|bts_lv

If we use AoIP the ASSIGNMENT REQUEST should give us up to 4 codec rates, but we will have to implement a check here. We probably will filter out the 4 highest, in case the MSC is selecting S0-S15 carelessly.

The difference now is that we may configure more then 4 codec rates in bts->mr_full|half. Intersected against the bits in the ASSIGNMENT REQUEST we will then still end up with the maximum of 4 codec rates in the active set.

S0-S15 from ASSIGNMENT REQUEST (up to 4 set)
VTY settings in msc->amr_conf
VTY settings in bts->mr_full|half (may have more then 4 set)
Intersection: lchan->mr_ms|bts_lv

So at least to my understanding now merging gerrit 11060 should not make it any worse, but we definetly should fix the problem and make use of the configuration options of the VTY.

See also: https://gerrit.osmocom.org/#/c/osmo-bsc/+/11060/

#11 - 10/09/2018 09:45 AM - neels

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

#12 - 10/22/2018 08:25 PM - dexter

- % Done changed from 80 to 90

Status update: The most problematic points from the VTY point of view are now resolved. One big problem was that the values the user could input were not really checked. For example the order in which the rate modes are entered is very important. Those bits are now checked. I have also improved the function gsm48_multirate_config() so that it runs some constancy checks on the input values. In order to make sure that VTY config is valid the VTY makes a testrun with gsm48_multirate_config() after each input.

The patchset is almost finished, however, the devil is in the details here. For example we allow in the msc node of the VTY to forbid certain rate modes. Unfortunately we can not articulate all of those propitiations on the A interface. So before we generate the multirate IE we must calculate another intersection. At the moment the resulting intersection does not pass the consistency checks in gsm48_multirate_config(), which is most likely because the checks are too restrictive (I already have an idea). Apart from that, I saw osmo-bsc sending a valid multirate IE via RSL, the call was successful, so I think we are very close to get this done.

#13 - 10/23/2018 04:48 PM - dexter

The related patches are now finished and up for review:

https://gerrit.osmocom.org/#/c/osmo-bsc/+/11441 gsm_data: set meaningful default values for amr modes
https://gerrit.osmocom.org/#/c/osmo-bsc/+/11442 codec_pref: also check amr codec rates in check_codec_pref()
https://gerrit.osmocom.org/#/c/osmo-bsc/+/11443 gsm_04_08: improve gsm48_multirate_config()
https://gerrit.osmocom.org/#/c/osmo-bsc/+/11444 bsc_vty: check amr mode parameters
https://gerrit.osmocom.org/#/c/osmo-bsc/+/11445 lchan_fsm: generate proper multirate configuration IE on RSL

#14 - 10/29/2018 01:56 PM - dexter

- Status changed from In Progress to Resolved

- % Done changed from 90 to 100

All patches are merged and things seem to be fine now. I think we can close this task.

Files

assignment_complete.png	32.5 KB	09/06/2018	dexter
assignment_request.png	57.4 KB	09/06/2018	dexter
02_ASSIGNMENT_REQUEST.png	36 KB	09/21/2018	dexter
03_ASSIGNMENT_COMPLETE.png	32.4 KB	09/21/2018	dexter
01_COMPLETE_LAYER_3_INFO.png	33.8 KB	09/21/2018	dexter
amr_ass_fail.pcapng	16.3 KB	09/25/2018	dexter