

OsmoBSC - Bug #3637

handover decision 2: properly check available codecs

10/09/2018 02:31 AM - neels

Status:	Feedback	Start date:	10/09/2018
Priority:	Normal	Due date:	
Assignee:	neels	% Done:	0%
Category:			
Target version:			
Spec Reference:			
Description			
in handover_decision_2.c, I'm fairly sure we should have more than this for speech codec checks:			
<pre>static bool codec_type_is_supported(struct gsm_subscriber_connection *conn, enum gsm0808_speech_codec_type type) { int i; struct gsm0808_speech_codec_list *clist = &conn->codec_list; if (!conn->codec_list.len) { /* We don't have a list of supported codecs. This should never happen. */ LOGPHOLCHAN(conn->lchan, LOGL_ERROR, "No Speech Codec List present, accepting all codecs\n"); return true; } for (i = 0; i < clist->len; i++) { if (clist->codec[i].type == type) return true; } LOGPHOLCHAN(conn->lchan, LOGL_DEBUG, "Codec not supported by MS or not allowed by MSC: %s\n", gsm0808_speech_codec_type_name(type)); return false; }</pre>			
Testing with SCCPlite, I also get above "this should never happen".			
Compare match_codec_pref() which checks a lot more than just the gsm0808 SCL.			
Related issues:			
Related to OsmoBSC - Bug #3529: osmo-bsc does not send the correct S0-S15 bit...		Resolved	09/06/2018
Related to OsmoBSC - Bug #3503: osmo-bsc: codec-list VTY cfg implementation n...		Resolved	08/27/2018

History

#1 - 10/09/2018 02:33 AM - neels

- Status changed from New to Feedback

- Assignee set to dexter

[dexter](#), I believe you're fairly familiar with who dictates what codecs, can you provide some input or even an improved implementation of above codec_type_is_supported()?

#2 - 10/09/2018 07:43 AM - dexter

- Assignee changed from dexter to neels

Hmm. I wonder how this works. This is about handover, so the connection is already made and the codec is decided but apparently the function compares against the whole codec list that was used during the assignment. That would be wrong but probably I just don't understand it right now.

I think what should happen is that the current codec that is in use on the conn should be checked against the various codecs the new BTS supports. So the one selected codec from the conn gets compared to the various codecs the new BTS supports. For the regular codecs this should be just

looking at `bts->codec` if the codec is supported by the BTS. For AMR there comes additional difficulty, `bts->mr_full` and `bts->mr_half` would also need some checking. (However, but those struct members still need some discussion anyway, see also [#3529](#))

You can generate a list with supported codecs using `gen_bss_supported_codec_list()` from `codec_pref.c`. Technically we do not need to go through `msc->audio_support[]` again, but using this function may have some advantage since you get a struct `gsm0808_speech_codec_list` as result and you won't have to go through all those struct members. The function would do that for you. (Also I already have in mind that this function also should go through all `lchans` and see which ones are free. So once we have that, you would only get the codecs that are supported in the very moment the function gets called.)

For AMR we also still need to finish the implementation about the active set. That's also related to [#3529](#). Basically we need to take the S0-S15 from the old BTS and intersect them with the S0-S15 of the new BTS. However, I think we also must intersect the result with the active set that is currently in use since the MSC side will expect only rates from the active set it has decided on. The topic is a little nasty, probably we need to discuss a bit about that before we move on.

#3 - 10/09/2018 09:45 AM - neels

On Tue, Oct 09, 2018 at 07:43:00AM +0000, dexter [REDMINE] wrote:

Hmm. I wonder how this works. This is about handover, so the connection is already made and the codec is decided but apparently the function compares against the whole codec list that was used during the assignment. That would be wrong but probably I just don't understand it right now.

[dexter](#), well, actually I think it's rather probably wrong right now and you know how to fix it =)

The remaining comments are very helpful, thanks!
I'll get back to you on details...

#4 - 10/09/2018 09:45 AM - neels

- Related to Bug #3529: *osmo-bsc does not send the correct S0-S15 bits for AMR in the Assignment Compl Message.* added

#5 - 10/10/2018 11:38 AM - neels

- Related to Bug #3503: *osmo-bsc: codec-list VTY cfg implementation not filtering correctly* added