

libosmo-abis - Bug #3622

Clear API mess in sign_link_up callback

10/02/2018 07:32 PM - pespin

Status:	New	Start date:	10/02/2018
Priority:	Normal	Due date:	
Assignee:	pespin	% Done:	0%
Category:			
Target version:			
Spec Reference:			
Description			
<pre><pespin_> ipa sign_link_up() callback is so fucked up. It has so many hidden details on how it is called and handled, lots of intrinsic assumptions based on who uses it and which messages it expects to receive. <LaF0rge> pespin_: welcome to an architecture designed for E1 which was later extended with Abis/IP support without introducing too many changes <LaF0rge> pespin_: also, written when understanding nothing about Abis/IP as there's no documentation on it... <pespin_> LaF0rge, it's fine, I'm just trying to share my understanding and find possible solutions to make it more maintainable <pespin_> 1st parameter is a void* which is only used by osmo-bsc during IPA_ID_RESP, and in that case is passed a "struct ipaccess_unit *" <pespin_> from which it takes unit ID (including TRX ID) <pespin_> Then 3rd parameter is "enum elinp_sign_type type", which can be basically either E1INP_SIGN_OML or E1INP_SIGN_RSL <pespin_> but in case it's E1INP_SIGN_RSL, it's actually encoded as E1INP_SIGN_RSL+trx_num <pespin_> and that only for expected BTS users, that is for message IPA_ID_GET/REQ <pespin_> for BSC messages, as trx_num is passed inside the first parameter, that trick is not used <pespin_> I'm willing to fix this, but of course it means breaking compat between osmo-bts/osmo-bsc and libosmo-abis. <pespin_> we should basically send always a "struct ipaccess_unit *" correctly filled in the first param, and stop using this RSL+trx_num hack in the 3rd parameter <pespin_> another option would be adding a new sign_link_up2 callback, and implement it in libosmo-abis + osmo-bsc and osmo-bts <pespin_> and then deprecate sign_link_up</pre>			
Related code map for BTS:			
<pre>log "Rx IPA RSL CONNECT IP=%s PORT=%u STREAM=0x%02x" --> elinp_ipa_bts_rsl_connect_n(trx_nr) elinp_ipa_bts_rsl_connect [BTS connects TCP conn to BSC as requested by BSC] elinp_ipa_bts_rsl_connect_n ipa_client_conn_create (set priv_nr E1INP_SIGN_RSL+trx_nr, read_cb=ipaccess_bts_read_cb up down_cb=ipaccess_bts_updown_cb [BSC SENDS US A ID GET/REQ] ipaccess_bts_read_cb ipaccess_bts_handle_ccm ops->sign_link_up (on ID_GET)</pre>			
<p>On the other hand, sign_link_up on BSC expects E1INP_SIGN_RSL to not contain the trx_num, that's why ipaccess_rcvmsg in libosmo-abis/src/input/ipaccess.c:168 calls it this way:</p>			
<pre>line->ops->sign_link_up(&unit_data, line, E1INP_SIGN_RSL);</pre>			

History

#1 - 10/02/2018 07:34 PM - pespin

While looking for a fix for [#3612](#) and understanding the code, I already did an initial cleanup of related parts regarding access to line ts and index

handling:

<https://gerrit.osmocom.org/#/c/libosmo-abis/+11203> ipaccess: Simplify handling of e1line ts