Description

Some big patches from wireshark.git epan/disseectors/packet-gsm_rlcmac.c/h still need to be checked for inclusion into osmo-pcu, since it was not clear if they benefit us. Those patches usually come as a bunch of 2-3 commits and later fixed many times. It actually makes more sense to get it from wireshark.git master directly.

One can check log of related patches with git log -u epan/disseectors/packet-gsm_rlcmac.*

First bunch of patches: support decode FDD CELL_INFORMATION of "UTRAN FDD Description"
dea5452b95daf18e38670a8e2b3b389f175fddf
774be29de0b4d93d01aeceb1518c2144017f557101a9
51c31cc3d8e8195a9f99f0a8af466ad84e9e6a8
51c31f842e81ec5f2370b13efa11f4e256e3091
(and other commits fixing stuff later)

Second bunch of patches: Enhancemnt to GSM RLCMAC dissection adding dissection of data block headers
12e5e625a6ce7f214d1111e48f12d5104144b53
cc6d434165e2e8df48f650ac0f236ec0dd844
(and others fixing later)

Third patch: 2g->3g/4g PS handover: 428ee66a1e524b49f9043729b1fibe9b4f52f409

History

#1 - 02/13/2020 07:01 PM - fixeria
- File vector1.dec added


```
vector1 = 4016713c094270ca2ae57ef90906aa0fc001f80222b
=========Start DECODE=================
+++++++++Finish DECODE (0)++++++++++
=========Start ENCODE================
+++++++++Finish ENCODE (0)++++++++++++
```

```
vector1 = 40 16 71 3d c0 94 27 0c a2 ae 57 ef 90 90 06 aa 0f c0 00 1f 80 22 2b
vector2 = 40 16 71 3d c0 94 27 48 a2 ae 57 ef 90 90 06 aa 00 00 1f 80 00 3f 00
vector1 == vector2 : FALSE
```

I have no idea why, but I hope Pycrate can help us to narrow down the problem.

```
from pycrate_csn1dir import uplink_rlc_mac_control_message
from binascii import unhexlify

# NOTRE the first octet should be skipped (MAC header)
vector1 = unhexlify('16 71 3d c0 94 27 0c a2 ae 57 ef 90 90 06 aa 0f c0 00 1f 80 22 2b'.replace(' ', ''))
vector2 = unhexlify('16 71 3d c0 94 27 48 a2 ae 57 ef 90 90 06 aa 00 00 1f 80 00 3f 00'.replace(' ', ''))

msg1 = uplink_rlc_mac_control_message.uplink_rlc_mac_control_message.clone()
msg2 = uplink_rlc_mac_control_message.uplink_rlc_mac_control_message.clone()

msg1.from_bytes(vector1)
print(msg1.show()) # See vector1.dec attached
msg2.from_bytes(vector2) # decoding fails: pycrate_core.charpy.CharpyErr: bitlen overflow: 1, max 0
```
I have been doing a second iteration over wireshark commits for rlcmac and csn1 files, and submitted a couple more patches:

remote: https://gerrit.osmocom.org/c/osmo-pcu/+/17628 rlcmac: support decode FDD_CELL_INFORMATION of 'UTRAN FDD Description
remote: https://gerrit.osmocom.org/c/osmo-pcu/+/17629 rlcmac: add dissection of 2G->3G/4G PS handover

I also did some fixes in csn1/rlcmac of both osmo-pcu and wireshark:
remote: https://gerrit.osmocom.org/c/osmo-pcu/+/17630 csn1: Fix Several typos & whitespace
remote: https://gerrit.osmocom.org/c/osmo-pcu/+/17631 csn1: verify enough bits present to decode whole CSN_UINT_ARRAY
remote: https://code.wireshark.org/review/36589 GSM RLC/MAC: fix typo in comment
remote: https://code.wireshark.org/review/36590 GSM RLC/MAC: Drop extra empty line
remote: https://code.wireshark.org/review/36591 CSN.1: verify enough bits present to decode whole CSN_UINT_ARRAY
remote: https://code.wireshark.org/review/36592 CSN.1: Optimize update of remaining_bits_len dissecting CSN_UINT_ARRAY
remote: https://code.wireshark.org/review/36593 CSN.1: Properly verify CSN_BITMAP length

I identified patches for 2 different topics which I didn't back-port because I have the feeling they are of no use to us:

Enhancemnt to GSM RLC/MAC dissection adding dissection of data block headers
12e5e625a6ce7f214d111f8e46f12d5104144b53
cc6d4341e65ef2e8d8488fe0ac0f236ece0dd844

So unless somebody reasons in a different way, I think we are done with porting stuff from packet-gsm_rlcmac.* and packet-csn1.* from wireshark.git (at wireshark.git commit 2f6a692fc027a6eb332bc3e2d546ac5c14d951 at the time of writing).

92ef3d0b55a37f042e1ce7e208e7f39cb49b9bce GSM RLC/MAC: add dissection of EC-GSM-IoT control messages