This is currently not supported/implemented. Having 11bit RACH gives us more information on the cause of channel establishment, especially when GSM + GPRS are used together.

Related issues:
- Related to OsmoBTS - Bug #1854: 11-bit RACH support breaks default 8-bit RACH...
  - Resolved 11/18/2016
- Related to OsmoTRX - Feature #3054: Extended (11-bit) RACH support in OsmoTRX
  - Stalled 03/10/2018
- Related to OsmoPCU - Feature #3014: fix re-apply patches reverted by #3013, r...
  - Resolved 02/27/2018
- Related to OsmoPCU - Bug #4338: Add EGPRS tests to TTCN3 PCU_Tests_RAW
  - Resolved 12/23/2019
- Has duplicate OsmoPCU - Bug #1834: Extended (11-bit) RACH is not properly han...
  - Closed 10/25/2016

Associated revisions

Revision 32e5641d - 12/11/2017 10:36 AM - max
Add functions for extended RACH coding
- convolutional code with puncturing
- encoding/decoding routines
- corresponding tests

Change-Id: I85a34a82d5cd39a594ee89d91a2338226066ab5d
Related: OS#1548

Revision 41ce6753 - 03/31/2020 12:26 PM - Vadim Yanitskiy
tests/coding: add 11-bit Access Burst samples from a real phone
This change adds several soft-bit (-127 .. 127) sequences containing EGPRS Packet Channel Request message (11-bit, payload only) sent by an EGPRS capable phone, and captured on the BTS/PCU side using a tool from the TRX Toolkit - trx_sniff.py.

As can be seen from the test output, none of decoded RA11 values looks like a valid EGPRS Packet Channel Request message (see table 11.2.5a.2 of 3GPP TS 44.060). All test sequences contain the same message with several random bits:

```
< EGPRS Packet channel request message content > ::= 
  < Signalling : 110011 < RandomBits : bit (5) > >;
```

since the phone was trying to perform Attach Request. It seems the bit order of decoded messages is somehow wrong.

Change-Id: ld80e471d252b9416217b56f4c8c0a8f5f1289fee
Related: OS#1548

Revision 9e713f38 - 03/31/2020 06:09 PM - Vadim Yanitskiy
coding: fix bit ordering in 11-bit RACH coding functions
According to 3GPP TS 44.004, figure 7.4.a.b, the format of 11-bit RACH uplink / Uplink access burst block is as defined follows:
As was (correctly) assumed in [1], the bit ordering in 11-bit RACH coding functions is wrong. The problem is that neither of generic functions from bit16gen.h can be used to load / store the RA11 value (regardless of the endianness), because they assume that the payload is 16 bit long.

With this patch applied, RA11 values from [1] look correct:

```
< EGPRS Packet channel request message content > ::= 
  < Signalling : 110011 < RandomBits : 00111 > > | 
  < Signalling : 110011 < RandomBits : 00110 > > | 
  < Signalling : 110011 < RandomBits : 01111 > > | 
  < Signalling : 110011 < RandomBits : 01100 > > | 
  < Signalling : 110011 < RandomBits : 00111 > > | 
  < Signalling : 110011 < RandomBits : 10110 > > ;
```

[1] ld80e471d252b9416217b56f4c8c0a8f51289fee
Change-Id: I43d30611dd69f77f2b3b46f4b56056a8891d3c24
Related: OS#1548

History

#1 - 11/09/2016 10:01 AM - laforge
- Status changed from New to In Progress
- Assignee set to arvind.sirsikar

11bit RACH is actually being worked on, assigning this to aravind. It might even be complete for some BTS models by now.

#2 - 11/09/2016 12:13 PM - arvind.sirsikar

Currently it is supported and integration tested for NuRAN LC1.5 and LC1.0. However there is no support for osmo-trx.

#4 - 08/17/2017 06:53 AM - laforge
- Assignee changed from arvind.sirsikar to sysmocom

#5 - 10/11/2017 01:12 AM - laforge
- Status changed from In Progress to New
- Assignee changed from sysmocom to msuraev

#6 - 10/11/2017 08:34 AM - laforge
- Priority changed from Normal to High

#7 - 11/28/2017 05:23 PM - msuraev
- Status changed from New to In Progress
- % Done changed from 0 to 10

Encoding/decoding support for 11-bit RACH for libosmocoding is available in gerrit 5062.

#8 - 12/04/2017 11:02 AM - msuraev
- Related to Bug #1854: 11-bit RACH support breaks default 8-bit RACH: collisions are possible added

#9 - 12/11/2017 02:51 PM - msuraev
- Status changed from In Progress to Stalled
- % Done changed from 10 to 20
Gerrit 5062 has been merged. Have to verify whether 11-bit RACH is supported on sysmoBTS.

#10 - 02/21/2018 11:36 PM - msuraev
Test with "gprs 11bit_rach_support_for_egprs" failed - seems like smth wasn't merged/missing. See related BTS bug.

#11 - 03/01/2018 11:13 PM - laforge
- Assignee changed from msuraev to sysmocom

#12 - 04/10/2018 05:34 PM - laforge
- Assignee changed from sysmocom to lynxis

#13 - 09/25/2018 07:16 PM - fixeria
The following patch should implement 11-bit RACH support of osmo-bts-trx:
https://gerrit.osmocom.org/#/c/osmo-bts/+/6315/
but should be properly tested.

#14 - 10/17/2018 10:26 AM - laforge
- Assignee changed from lynxis to msuraev

#15 - 11/01/2018 01:41 PM - msuraev
- Has duplicate Bug #1834: Extended (11-bit) RACH is not properly handled added

#16 - 11/01/2018 01:42 PM - msuraev
- Related to Feature #3054: Extended (11-bit) RACH support in OsmoTRX added

#17 - 02/14/2019 10:26 AM - laforge
- Assignee changed from lynxis to msuraev

#18 - 02/14/2019 04:37 PM - fixeria
- Related to Feature #3014: fix re-apply patches reverted by #3013, related: UL and DL packet assignment, and Timing Advance added

#19 - 09/17/2022 10:00 AM - fixeria
- Assignee deleted (fixeria)

Both osmo-trx and osmo-bts-trx do support 11-bit encoded Access Bursts now. We also have a TTCN-3 test case for that (see BTS_Tests.TC_pcu_ext_rach_content) and it passes. However, all my attempts to test the network with the real phones have been unsuccessful. Unfortunately, I have no time to investigate further now.

#20 - 01/21/2020 01:23 PM - laforge
- Assignee set to daniel
- Priority changed from High to Normal

#21 - 01/22/2020 02:52 PM - pespin
- Related to Bug #4338: Add EGPRS tests toTTCN3 PCU_Tests_RAW added

#22 - 01/22/2020 02:54 PM - pespin
I'm trying to add some TTCN3 tests with 11bit RACH support for osmo-pcu in PCU_Tests_RAW as part of #4338. I'd say that's all what's missing here.

#23 - 01/22/2020 07:32 PM - fixeria
I'd say that's all what's missing here.

I don't think so. OsmoPCU does not seem to handle 11-bit RACH correctly when running with osmo-trx and osmo-bts-trx.

#24 - 03/24/2020 09:44 PM - fixeria
- Spec Reference set to 3GPP TS 44.004, section 7.4a; 3GPP TS 44.060, sections 11.2.5 and 11.2.5a

#25 - 03/31/2020 06:39 PM - fixeria

OsmoPCU does not seem to handle 11-bit RACH correctly when running with osmo-trx and osmo-bts-trx.

https://gerrit.osmocom.org/c/libosmocore/+/17692/ tests/coding: add 11-bit Access Burst samples from a real phone
https://gerrit.osmocom.org/c/libosmocore/+/17693/ coding: fix bit ordering in 11-bit RACH coding functions

With this fix applied, OsmoPCU works much better when running with osmo-bts-trx.
I also have a (draft) TTCN-3 test case for OsmoPCU.

#26 - 04/01/2020 07:45 PM - fixeria

I also have a (draft) TTCN-3 test case for OsmoPCU.

https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/17704/ library/RLCMAC_CSN1_TYPES: add EGPRS Packet Channel Request definition
https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/17706/ PCU: add test cases for EGPRS Packet Channel Request

For some reason, TITAN decodes different extended RA value in IA Rest Octets...
Any help would be appreciated. Wireshark shows that everything is ok (as expected).

#27 - 05/05/2020 11:01 AM - fixeria
- % Done changed from 20 to 60

TTCN-3 test cases have been merged and, as can be seen, they're passing in Jenkins.

Though I am still not satisfied by some code parts handling 11 bit EGPRS Packet Channel Request in osmo-pcu, for example:

```c
static inline uint16_t mslot_class_from_ra(uint16_t ra, bool is_11bit)
{
    if (is_11bit)
        return (ra & 0x3e0) >> 5;
    /* set multislot class to 0 for 8-bit RACH, since we don't know it yet */
    return 0;
}
```

The way we parse the multislot class does not seem to be correct, because it's optional and its presence depends on type of EGPRS Packet Channel Request. I believe we should implement proper parsing, and here is a draft change for that: https://gerrit.osmocom.org/c/osmo-pcu/+/17718.

#28 - 05/12/2020 12:03 PM - laforge
- Assignee changed from daniel to fixeria

#29 - 05/21/2020 08:13 AM - fixeria
- Status changed from Stalled to In Progress

I just realized that RR Immediate Assignment Reject (IAR Rest Octets) should also contain Extended RA, when sent in response to EGPRS Packet Access Request. More details can be found in 3GPP TS 44.018, section 10.5.2.17.

#30 - 05/21/2020 08:20 AM - fixeria

Actually, Encoding::write_immediate_assignment_reject() does support writing IAR Rest Octets with Extended RA. We just need a TTCN-3 test case.
Actually, Encoding::write_immediate_assignment_reject() does support writing IAR Rest Octets with Extended RA. We just need a TTCN-3 test case.

https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/18420/ library/GSM_RR_Types: fix definition of FeatureIndicator
https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/18422/ library/GSM_RR_Types: add receive templates for IMM ASS Reject
https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/18423/ PCU: introduce TC_egprs_pkt_chan_req_reject_content
https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/18424/ PCU: introduce TC_egprs_pkt_chan_req_reject_emergency

While working on EGPRS Packet Channel Request coding support, I discovered and fixed some problems in CSN.1 codec:

https://gerrit.osmocom.org/c/osmo-pcu/+/18430/ csn1: fix M_CHOICE: restrict maximum length of the choice list
https://gerrit.osmocom.org/c/osmo-pcu/+/18431/ csn1: fix csnStreamEncoder(): also check length of the choice list
https://gerrit.osmocom.org/c/osmo-pcu/+/18432/ csn1: fix csnStreamEncoder(): always check the choice index
https://gerrit.osmocom.org/c/osmo-pcu/+/18433/ csn1: fix: never use enumerated types in codec structures

All previously submitted changes have been merged. These two new changes are waiting for review:

https://gerrit.osmocom.org/c/osmo-pcu/+/18386/ bts: refactor handling and parsing of RACH.ind
https://gerrit.osmocom.org/c/osmo-pcu/+/18387/ BTS::parse_rach_ind(): properly handle EGPRS Packet Channel Request

as soon as they're merged, we can finally close this ticket.

All patches have been merged.