

## OpenBSC - Bug #1971

### TCH/F\_TCH/H\_PDCH: BSC doesn't allocate TCH/F

03/08/2017 12:13 PM - fixeria

<b>Status:</b>	Rejected	<b>Start date:</b>	03/08/2017
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	fixeria	<b>% Done:</b>	100%
<b>Category:</b>	OpenBSC		
<b>Target version:</b>	Dynamic TCH/H TCH/F PDCH		
<b>Resolution:</b>		<b>Spec Reference:</b>	

#### Description

There are some old phones which don't support TCH/H. And since dynamic TS switching was implemented in OpenBSC, there is an opportunity to make everyone happy - to use dynamic timeslot switching. As I understand, OpenBSC switches the lchan type depending on which type is supported/requested by MS side. So, if a phone has no TCH/H support and indicates that in classmask, the network should switch TCH/F\_TCH/H\_PDCH into TCH/F mode.

I decided to simulate such situation using OpenBSC + OsmoBTS + OsmoTRX as network side and OsmocomBB as MS side. The mobile application of OsmocomBB allows to configure supported bands, codecs, channel capability and other features, moreover it provides great logging output.

#### TRX Channel configuration

```
timeslot 0
  phys_chan_config CCCH+SDCCH4
timeslot 1
  phys_chan_config SDCCH8
timeslot 2
  phys_chan_config TCH/F_TCH/H_PDCH
timeslot 3
  phys_chan_config TCH/F_TCH/H_PDCH
timeslot 4
  phys_chan_config TCH/F_TCH/H_PDCH
timeslot 5
  phys_chan_config TCH/F_TCH/H_PDCH
timeslot 6
  phys_chan_config TCH/F_TCH/H_PDCH
timeslot 7
  phys_chan_config TCH/F_TCH/H_PDCH
```

#### OsmocomBB mobile configuration (disabled TCH/H support):

```
...
codec full-speed prefer
no codec half-speed
support
...
channel-capability sdcch+tchf
full-speech-v1
full-speech-v2
no half-speech-v1
...
```

#### Call to number 995 (LCR)

#### OsmocomBB/mobile logs

```
<0009> mnccms.c:570 Make call to 995
<0009> mnccms.c:153 support TCH/F only
<0009> mnccms.c:174 support full rate v2
```

```
<0009> mnccms.c:178 support full rate v1
...
<0001> gsm48_rr.c:1447 CHANNEL REQUEST: e0 (Orig TCH/F)
...
<0001> gsm48_rr.c:3576 CHANNEL MODE MODIFY
<0001> gsm48_rr.c:3604 (chan_nr 0x12 ARFCN 100 TS 2 SS 0 TSC 7 mode 1)
<0001> gsm48_rr.c:270 Not supporting half-rate speech V1
<0001> gsm48_rr.c:898 RR STATUS (cause #9)
...
```

## Classmask obtained from Wireshark

```
Bearer Capability 1
  MS supports at least full rate speech version 1,
  but does not support half rate speech version 1.
```

## OpenBSC logs

```
...
<0000> chan_alloc.c:352 (bts=0,trx=0,ts=2,pchan=TCH/F_TCH/H_PDCH as PDCH) Allocating lchan=0 as TC
H_H
<0004> abis_rsl.c:1823 (bts=0,trx=0,ts=2,ss=0) Activating ARFCN(100) SS(0) lctype TCH_H r=OTHER ra
=0xea ta=1
<0004> abis_rsl.c:572 (bts=0,trx=0,ts=2,ss=0) saved rqd_ref=(nil) ta=0
<0004> abis_rsl.c:2451 (bts=0,trx=0,ts=2,pchan=TCH/F_TCH/H_PDCH as PDCH) starting switchover to TC
H/H
<0004> abis_rsl.c:1189 (bts=0,trx=0,ts=2,ss=0) state ACTIVE -> RELEASE REQUESTED
<0004> abis_rsl.c:853 (bts=0,trx=0,ts=2,ss=0) RF Channel Release
<0004> abis_rsl.c:159 (bts=0,trx=0,ts=2,pchan=TCH/F_TCH/H_PDCH switching PDCH -> TCH/H) Abis RSL r
x DCHAN: mismatching chan_nr=0x82
<0004> abis_rsl.c:924 (bts=0,trx=0,ts=2,ss=0) RF CHANNEL RELEASE ACK
<0004> abis_rsl.c:1189 (bts=0,trx=0,ts=2,ss=0) state RELEASE REQUESTED -> NONE
<0004> abis_rsl.c:971 (bts=0,trx=0,ts=2,pchan=TCH/F_TCH/H_PDCH switching PDCH -> TCH/H) Rx RSL Cha
nnel Release ack for lchan 0
<0004> abis_rsl.c:2522 (bts=0,trx=0,ts=2,pchan=TCH/F_TCH/H_PDCH switching PDCH -> TCH/H) switchove
r: release complete, activating new pchan type
<0004> abis_rsl.c:579 (bts=0,trx=0,ts=2,pchan=TCH/F_TCH/H_PDCH switching PDCH -> TCH/H) Tx RSL Cha
nnel Activate with act_type=INITIAL
<0004> abis_rsl.c:1189 (bts=0,trx=0,ts=2,ss=0) state NONE -> ACTIVATION REQUESTED
<0004> abis_rsl.c:1546 (bts=0,trx=0,ts=2,ss=0) CHANNEL ACTIVATE ACK
<0004> abis_rsl.c:1189 (bts=0,trx=0,ts=2,ss=0) state ACTIVATION REQUESTED -> ACTIVE
<0004> abis_rsl.c:2638 (bts=0,trx=0,ts=2,pchan=TCH/F_TCH/H_PDCH as TCH/H) switchover from PDCH com
plete.
<0000> abis_rsl.c:2013 (bts=0,trx=0,ts=2,ss=0) SAPI=0 ESTABLISH INDICATION
<0000> gsm_04_08.c:3960 Dispatching 04.08 message, pdisc=5
<0002> gsm_04_08.c:998 <- CM SERVICE REQUEST serv_type=0x01 MI(TMSI)=1799138458
<0002> gsm_04_08_utils.c:648 -> CM SERVICE ACK
<0000> abis_rsl.c:2012 (bts=0,trx=0,ts=2,ss=0) SAPI=0 DATA INDICATION
<0000> gsm_04_08.c:3983 Dispatching 04.08 message, pdisc=3
<0001> gsm_04_08.c:3877 (bts 0 trx 0 ts 2 ti 8 sub 20134) Received 'SETUP' from MS in state 0 (NUL
L)
<0001> gsm_04_08.c:3882 Unknown transaction ID 8, creating new trans.
<0001> transaction.c:71 subscr=0x173b3c0, net=0x15facd0
<0001> gsm_04_08.c:1605 new state NULL -> INITIATED
<0001> gsm_04_08.c:2285 Subscriber 1010000000000 (20134) sends SETUP to 995
<0001> gsm_04_08.c:1667 (bts 0 trx 0 ts 2 ti 8 sub 20134) Sending 'MNCC_SETUP_IND' to MNCC.
<0001> gsm_04_08.c:3779 (bts 0 trx 0 ts 2 ti 08 sub 20134) Received 'MNCC_LCHAN_MODIFY' from MNCC
in state 1 (INITIATED)
<0003> gsm_04_08_utils.c:500 -> CHANNEL MODE MODIFY mode=0x01
<0001> gsm_04_08.c:3779 (bts 0 trx 0 ts 2 ti 08 sub 20134) Received 'MNCC_CALL_PROC_REQ' from MNCC
in state 1 (INITIATED)
<0001> gsm_04_08.c:1605 new state INITIATED -> MO_CALL_PROC
<0001> gsm_04_08.c:143 (bts 0 trx 0 ts 2 ti 80) Sending 'CALL_PROC' to MS.
<0001> gsm_04_08.c:2049 queue tch_recv_mncc request (1)
<0001> gsm_04_08.c:3779 (bts 0 trx 0 ts 2 ti 08 sub 20134) Received 'MNCC_SETUP_RSP' from MNCC in
state 3 (MO_CALL_PROC)
```

```

<0001> gsm_04_08.c:2208 starting timer T313 with 30 seconds
<0001> gsm_04_08.c:1605 new state MO_CALL_PROC -> CONNECT_IND
<0001> gsm_04_08.c:143 (bts 0 trx 0 ts 2 ti 80) Sending 'CONNECT' to MS.
<0000> abis_rsl.c:2012 (bts=0,trx=0,ts=2,ss=0) SAPI=0 DATA INDICATION
<0003> osmo_msc.c:107 CIPHERING MODE COMPLETE
<0003> osmo_msc.c:111 No authentication/cipher operation in progress !!!
<0000> abis_rsl.c:2012 (bts=0,trx=0,ts=2,ss=0) SAPI=0 DATA INDICATION
<0000> gsm_04_08.c:3983 Dispatching 04.08 message, pdisc=3
<0001> gsm_04_08.c:3877 (bts 0 trx 0 ts 2 ti 8 sub 20134) Received 'CONNECT_ACK' from MS in state
28 (CONNECT_IND)
<0001> gsm_04_08.c:1647 stopping pending timer T313
<0001> gsm_04_08.c:1605 new state CONNECT_IND -> ACTIVE
<0001> gsm_04_08.c:1667 (bts 0 trx 0 ts 2 ti 8 sub 20134) Sending 'MNCC_SETUP_COMPL_IND' to MNCC.
<0003> osmo_msc.c:82 MSC assign failure (do nothing).
...

```

## LCR logs

```

...
Codec negotiation LCR<->BSC bearer capa='given by MS'
  speech version='Full Rate given' ignored='Not TCH/F'
  error 'All given payload types unsupported'
MNCC_LCHAN_MODIFY LCR<->BSC speech version='Full/Half Rate given' mode 0x01
...

```

## Whats? O\_o

As I can see, OpenBSC ignores classmask of the calling phone, and allocates TCH/H instead of TCH/F. Despite the unsupported channel type, both sides getting connected...

My knowledge about a normal behavior in such situation is limited, so I have the following questions:

- 1) Should a phone reject a channel with unsupported type?
- 2) Why BSC allocates exactly TCH/H instead of TCH/F?

### Related issues:

Related to OsmoMSC - Feature #1778: avoid mismatching TCH/F vs TCH/H pchan types	<b>Resolved</b> <b>07/21/2016</b>
--	-----------------------------------

## History

### #1 - 03/28/2017 08:08 AM - fixeria

- Status changed from New to In Progress
- % Done changed from 0 to 10
- Parent task set to #1778

### #2 - 03/28/2017 08:20 AM - fixeria

Looking at the source code, I just figured out, that TCH/F on TCH/F\_TCH/H\_PDCH was reasonably disabled due to <http://osmocom.org/issues/1778>

### chan\_alloc.c / lchan\_alloc(bts, type, allow\_bigger):

```

...
case GSM_LCHAN_TCH_F:
...
  /* Try fully dynamic TCH/F_TCH/H_PDCH as TCH/F... */
  if (!lchan && bts->network->dyn_ts_allow_tch_f) {
    lchan = _lc_dyn_find_bts(bts,
                          GSM_PCHAN_TCH_F_TCH_H_PDCH,
                          GSM_PCHAN_TCH_F);
    if (lchan)
      type = GSM_LCHAN_TCH_F;
  }
  /* ...and as TCH/H. */
  if (!lchan) {
    lchan = _lc_dyn_find_bts(bts,

```

```
        GSM_PCHAN_TCH_F_TCH_H_PDCH,  
        GSM_PCHAN_TCH_H);  
    if (lchan)  
        type = GSM_LCHAN_TCH_H;  
    }  
    break;
```

Take a look at `bts->network->dyn_ts_allow_tch_f`.

```
/*  
 * For osmo-nitb, skip TCH/F for now, because otherwise dyn TS  
 * always imply the possibility to have a mix of TCH/F and  
 * TCH/H channels; if two phones request a TCH/F and a TCH/H,  
 * respectively, they cannot call each other. If we deny TCH/F,  
 * they will both fall back to TCH/H, and dynamic channels are  
 * usable. See http://osmocom.org/issues/1778.  
 *  
 * A third-party MSC may well be able to handle a TCH/H TCH/F  
 * mismatch.  
 */  
bsc_gsmnet->dyn_ts_allow_tch_f = false;
```

### **#3 - 05/29/2017 06:41 PM - fixeria**

- Status changed from *In Progress* to *Closed*
- % Done changed from 10 to 100

### **#4 - 10/09/2017 02:45 PM - neels**

- Status changed from *Closed* to *Rejected*
- Parent task deleted (#1778)

### **#5 - 10/09/2017 02:45 PM - neels**

- Related to Feature #1778: avoid mismatching TCH/F vs TCH/H pchan types added