

OsmoMSC - Feature #1778

avoid mismatching TCH/F vs TCH/H pchan types

07/21/2016 04:50 PM - neels

Status: Resolved	Start date: 07/21/2016
Priority: Low	Due date:
Assignee: neels	% Done: 100%
Category: A interface (general)	
Target version:	
Resolution:	Spec Reference:
Description	
<p>In an osmo-nitb configuration with both TCH/F and TCH/H available, a voice call may end up with TCH/H to one phone and TCH/F to the other. The result is:</p>	
<pre>DCC <0001> gsm_04_08.c:1649 Cannot patch through call with different channel types: local = TCH_F, remote = TCH_H</pre>	
<p>If I configure the BSC to have both TCH_H and TCH_F slots, my two test phones try to call each other with mismatching TCH types. The CHANNEL REQUIRED messages send TCH/H and TCH/F, respectively.</p>	
<p>phones: Samsung S4 vs. Sony Ericsson SE K800i.</p>	
<p>If the CN is configured with only TCH/H or only TCH/F, the two phones talk to each other fine.</p>	
<p>If one side started off in TCH/H, the other side should be urged towards TCH/H as well. So we should "clamp" the callee to the caller's pchan type.</p>	
Related issues:	
Related to OsmoBTS - Feature #1776: Implement fully dynamic TCH/F + TCH/H + P...	Closed 07/12/2016
Related to OsmoNITB - Bug #1663: don't connect channels of incompatible voice...	Closed 03/17/2016
Related to OsmoBTS - Bug #1865: TCH/F_PDCH is not working as TCH/F channel fo...	Rejected 12/02/2016
Related to OpenBSC - Bug #1971: TCH/F_TCH/H_PDCH: BSC doesn't allocate TCH/F	Rejected 03/08/2017
Related to OsmoBSC - Feature #1606: hand-over for load distribution among BTSs	Resolved 02/23/2016
Related to OsmoBSC - Feature #1608: various handover improvements, meta-issue	Rejected 02/23/2016
Blocks OsmoBTS - Bug #1853: validate dynamic TCH/PDCH support in osmo-bts-trx	Resolved 11/18/2016

History

#1 - 07/21/2016 04:56 PM - neels

- Related to Feature #1776: Implement fully dynamic TCH/F + TCH/H + PDCH switching added

#2 - 07/21/2016 04:58 PM - neels

<http://lists.osmocom.org/pipermail/openbsc/2016-July/009538.html>

#3 - 07/25/2016 08:23 AM - msuraev

- Related to Bug #1663: don't connect channels of incompatible voice codec added

#4 - 07/26/2016 01:00 PM - neels

- Status changed from New to Feedback

- Assignee set to laforge

- Priority changed from Normal to Urgent

In the light of one sponsor's desire to have full TCH/F and TCH/H dynamic

capabilities and <https://gerrit.osmocom.org/603> , how urgent would you mark this issue?

(Setting to "Urgent" like [#1776](#), pending feedback.)

#5 - 07/28/2016 02:28 PM - neels

I've taken a look at the chain of actions.
Here are some details (with some log messages added that are not in master).

MS 1 exten 46180, IMSI: ..0038
MS 2 exten 47828, IMSI: ..0012

- MS 1: Channel Requested, type = TCH/H

```
20160728154822906 DRR <0003> gsm_04_08_utils.c:190 chreq_type_necil[5] = TCH_H
20160728154822906 DRLD <0000> chan_alloc.c:379 (bts=0,trx=0,ts=2,pchan=TCH/F_TCH/H_PDCH as PDCH) Allocating lchan=0 as TCH_H
20160728154823380 DRR <0003> bsc_api.c:502 CLASSMARK CHANGE CM2(len=3) CM3(len=6)
20160728154823777 DRR <0003> bsc_api.c:647 BSC: Passing unknown 04.08 RR message type 0x60 to MSC
20160728154823777 DRR <0003> gsm_04_08.c:1281 MSC: Unimplemented GSM 04.08 RR msg type 0x60
20160728154823979 DRR <0003> bsc_api.c:590 GRPS SUSPEND REQUEST
20160728154824178 DCC <0001> gsm_04_08.c:3600 (bts 0 trx 0 ts 2 ti 8 sub 46180) Received 'SETUP' from MS in state 0 (NULL)
20160728154824178 DCC <0001> gsm_04_08.c:3605 Unknown transaction ID 8, creating new trans.
20160728154824178 DCC <0001> transaction.c:71 subscr=0x80dea0, net=0x7dd210
20160728154824178 DCC <0001> gsm_04_08.c:1323 new state NULL -> INITIATED
20160728154824178 DCC <0001> gsm_04_08.c:2003 Subscriber 901990000000038 (46180) sends SETUP to 47828
20160728154824178 DCC <0001> gsm_04_08.c:1384 (bts 0 trx 0 ts 2 ti 8 sub 46180) Sending 'MNCC_SETUP_IND' to MNCC.
20160728154824178 DMNCC <0006> mncc_builtin.c:346 (call 80000001) Call created.
20160728154824178 DMNCC <0006> mncc_builtin.c:355 (call 80000001) Received message MNCC_SETUP_IND
20160728154824178 DMNCC <0006> mncc_builtin.c:125 (call 80000001) Creating new remote instance 1.
20160728154824178 DMNCC <0006> mncc_builtin.c:134 (call 80000001) Accepting call.
20160728154824178 DMNCC <0006> gsm_04_08.c:3290 receive message MNCC_CALL_PROC_REQ (lchan_type 0 NONE)
20160728154824178 DCC <0001> gsm_04_08.c:3507 (bts 0 trx 0 ts 2 ti 08 sub 46180) Received 'MNCC_CALL_PROC_REQ' from MNCC in state 1 (INITIATED)
20160728154824178 DCC <0001> gsm_04_08.c:1323 new state INITIATED -> MO_CALL_PROC
20160728154824178 DCC <0001> gsm_04_08.c:142 (bts 0 trx 0 ts 2 ti 80) Sending 'CALL_PROC' to MS.
20160728154824178 DMNCC <0006> mncc_builtin.c:142 (call 80000001) Modify channel mode: SPEECH_V1
20160728154824178 DMNCC <0006> gsm_04_08.c:3290 receive message MNCC_LCHAN_MODIFY (lchan_type 0 NONE)
20160728154824178 DCC <0001> gsm_04_08.c:3507 (bts 0 trx 0 ts 2 ti 08 sub 46180) Received 'MNCC_LCHAN_MODIFY' from MNCC in state 3 (MO_CALL_PROC)
20160728154824178 DMSC <000a> bsc_api.c:402 Sending ChanModify for speech: SPEECH_V1 on channel TCH_H
20160728154824178 DRR <0003> gsm_04_08_utils.c:502 -> CHANNEL MODE MODIFY mode=0x01
20160728154824178 DMNCC <0006> mncc_builtin.c:150 (call 80000001,lchan 3 TCH_H) Forwarding SETUP to remote
.
```

- mncc_tx_to_cc receives SETUP REQ with lchan_type still explicitly known.

```
20160728154824178 DMNCC <0006> gsm_04_08.c:3290 receive message MNCC_SETUP_REQ (lchan_type 3 TCH_H)
20160728154824179 DCC <0001> transaction.c:71 subscr=0x81d3d0, net=0x7dd210
20160728154824179 DCC <0001> gsm_04_08.c:3431 274018000000012 created transaction
20160728154824179 DCC <0001> gsm_04_08.c:3459 274018000000012 paging with rsl_channeed=3
20160728154824179 DPAG <0007> paging.c:291 Start paging of subscriber 19 on bts 0.
20160728154824617 DRR <0003> gsm_04_08_utils.c:527 CHANNEL MODE MODIFY ACK
20160728154824617 DRR <0003> osmo_msc.c:76 MSC assign complete (do nothing).
```

- In the paging command going to MS 2, we would always send RSL_CHANNEED_TCH_F, I changed to RSL_CHANNEED_TCH_ForH, but no effect. There is no RSL_CHANNEED_TCH_H (TS 08.05 9.3.40)

```
20160728154824679 DPAG <0007> paging.c:83 Going to send paging commands: imsi: '274018000000012' tmsi: '0xbb01b012'
20160728154825179 DPAG <0007> paging.c:83 Going to send paging commands: imsi: '274018000000012' tmsi: '0xbb01b012'
```

- MS 2 answers with Channel Requested, as TCH/F type [1]

20160728154825280 DRR <0003> gsm_04_08_utils.c:190 chreq_type_necil[11] = TCH_F

- lchan_alloc() allocates a TCH/F [2]

```
20160728154825280 DRLL <0000> chan_alloc.c:379 (bts=0,trx=0,ts=3,pchan=TCH/F_TCH/H_PDCH as PDCH) Allocating lchan=0 as TCH_F
20160728154825484 DRR <0003> gsm_04_08.c:1216 PAGING RESPONSE: MI(TMSI)=3137450002
20160728154825484 DRR <0003> gsm_04_08.c:1236 <- Channel was requested by 274018000000012
20160728154825489 DPAG <0007> paging.c:379 Stop paging on bts 0, calling cbfn.
20160728154825489 DCC <0001> gsm_04_08.c:1454 Paging subscr 47828 succeeded!
20160728154825489 DCC <0001> gsm_04_08.c:1925 starting timer T303 with 30 seconds
20160728154825489 DCC <0001> gsm_04_08.c:1323 new state NULL -> CALL_PRESENT
20160728154825489 DCC <0001> gsm_04_08.c:142 (bts 0 trx 0 ts 3 ti 00) Sending 'SETUP' to MS.
20160728154825623 DRR <0003> bsc_api.c:502 CLASSMARK CHANGE CM2(len=3) CM3(len=11)
20160728154826182 DRR <0003> bsc_api.c:647 BSC: Passing unknown 04.08 RR message type 0x60 to MSC
20160728154826182 DRR <0003> gsm_04_08.c:1281 MSC: Unimplemented GSM 04.08 RR msg type 0x60
20160728154826325 DRR <0003> bsc_api.c:590 GRPS SUSPEND REQUEST
20160728154826463 DCC <0001> gsm_04_08.c:3600 (bts 0 trx 0 ts 3 ti 0 sub 47828) Received 'CALL_CONF' from MS in state 6 (CALL_PRESENT)
20160728154826463 DCC <0001> gsm_04_08.c:1364 stopping pending timer T303
20160728154826463 DCC <0001> gsm_04_08.c:1925 starting timer T310 with 180 seconds
20160728154826463 DCC <0001> gsm_04_08.c:1323 new state CALL_PRESENT -> MO_TERM_CALL_CONF
20160728154826463 DCC <0001> gsm_04_08.c:1384 (bts 0 trx 0 ts 3 ti 0 sub 47828) Sending 'MNCC_CALL_CONF_IN D' to MNCC.
20160728154826463 DMNCC <0006> mncc_builtin.c:355 (call 1) Received message MNCC_CALL_CONF_IND
20160728154826463 DMNCC <0006> gsm_04_08.c:3290 receive message MNCC_LCHAN_MODIFY (lchan_type 2 TCH_F)
20160728154826463 DCC <0001> gsm_04_08.c:3507 (bts 0 trx 0 ts 3 ti 00 sub 47828) Received 'MNCC_LCHAN_MODIFY' from MNCC in state 9 (MO_TERM_CALL_CONF)
20160728154826463 DMSC <000a> bsc_api.c:402 Sending ChanModify for speech: SPEECH_V1 on channel TCH_F
20160728154826463 DRR <0003> gsm_04_08_utils.c:502 -> CHANNEL MODE MODIFY mode=0x01
20160728154826644 DRR <0003> gsm_04_08_utils.c:527 CHANNEL MODE MODIFY ACK
20160728154826644 DRR <0003> osmo_msc.c:76 MSC assign complete (do nothing).
20160728154826781 DCC <0001> gsm_04_08.c:3600 (bts 0 trx 0 ts 3 ti 0 sub 47828) Received 'ALERTING' from MS in state 9 (MO_TERM_CALL_CONF)
20160728154826781 DCC <0001> gsm_04_08.c:1364 stopping pending timer T310
20160728154826781 DCC <0001> gsm_04_08.c:1925 starting timer T301 with 180 seconds
20160728154826781 DCC <0001> gsm_04_08.c:1323 new state MO_TERM_CALL_CONF -> CALL_RECEIVED
20160728154826781 DCC <0001> gsm_04_08.c:1384 (bts 0 trx 0 ts 3 ti 0 sub 47828) Sending 'MNCC_ALERT_IND' to MNCC.
20160728154826781 DMNCC <0006> mncc_builtin.c:355 (call 1) Received message MNCC_ALERT_IND
20160728154826781 DMNCC <0006> mncc_builtin.c:168 (call 1) Forwarding ALERT to remote.
20160728154826781 DMNCC <0006> gsm_04_08.c:3290 receive message MNCC_ALERT_REQ (lchan_type 0 NONE)
20160728154826781 DCC <0001> gsm_04_08.c:3507 (bts 0 trx 0 ts 2 ti 08 sub 46180) Received 'MNCC_ALERT_REQ' from MNCC in state 3 (MO_CALL_PROC)
20160728154826781 DCC <0001> gsm_04_08.c:1323 new state MO_CALL_PROC -> CALL_DELIVERED
20160728154826781 DCC <0001> gsm_04_08.c:142 (bts 0 trx 0 ts 2 ti 80) Sending 'ALERTING' to MS.
20160728154829403 DCC <0001> gsm_04_08.c:3600 (bts 0 trx 0 ts 3 ti 0 sub 47828) Received 'CONNECT' from MS in state 7 (CALL_RECEIVED)
20160728154829403 DCC <0001> gsm_04_08.c:1364 stopping pending timer T301
20160728154829403 DCC <0001> gsm_04_08.c:1323 new state CALL_RECEIVED -> CONNECT_REQUEST
20160728154829403 DCC <0001> gsm_04_08.c:1384 (bts 0 trx 0 ts 3 ti 0 sub 47828) Sending 'MNCC_SETUP_CNF' to MNCC.
20160728154829403 DMNCC <0006> mncc_builtin.c:355 (call 1) Received message MNCC_SETUP_CNF
20160728154829403 DMNCC <0006> mncc_builtin.c:196 (call 1) Acknowledge SETUP.
20160728154829403 DMNCC <0006> gsm_04_08.c:3290 receive message MNCC_SETUP_COMPL_REQ (lchan_type 0 NONE)
20160728154829403 DCC <0001> gsm_04_08.c:3507 (bts 0 trx 0 ts 3 ti 00 sub 47828) Received 'MNCC_SETUP_COMPL_REQ' from MNCC in state 8 (CONNECT_REQUEST)
20160728154829403 DCC <0001> gsm_04_08.c:1323 new state CONNECT_REQUEST -> ACTIVE
20160728154829403 DCC <0001> gsm_04_08.c:142 (bts 0 trx 0 ts 3 ti 00) Sending 'CONNECT_ACK' to MS.
20160728154829403 DMNCC <0006> mncc_builtin.c:203 (call 1) Sending CONNECT to remote.
20160728154829403 DMNCC <0006> gsm_04_08.c:3290 receive message MNCC_SETUP_RSP (lchan_type 0 NONE)
20160728154829403 DCC <0001> gsm_04_08.c:3507 (bts 0 trx 0 ts 2 ti 08 sub 46180) Received 'MNCC_SETUP_RSP' from MNCC in state 4 (CALL_DELIVERED)
20160728154829403 DCC <0001> gsm_04_08.c:1925 starting timer T313 with 30 seconds
20160728154829403 DCC <0001> gsm_04_08.c:1323 new state CALL_DELIVERED -> CONNECT_IND
20160728154829403 DCC <0001> gsm_04_08.c:142 (bts 0 trx 0 ts 2 ti 80) Sending 'CONNECT' to MS.
20160728154829403 DMNCC <0006> mncc_builtin.c:209 (call 1) Bridging with remote.
20160728154829403 DMNCC <0006> gsm_04_08.c:3290 receive message MNCC_BRIDGE (lchan_type 0 NONE)
```

- bridge is completed, but pchans mismatch

```
20160728154829403 DCC <0001> gsm_04_08.c:1638 Setting up TCH map between (bts=0,trx=0,ts=3,TCH_F) and (bts=0,trx=0,ts=2,TCH_H)
20160728154829403 DCC <0001> gsm_04_08.c:1649 Cannot patch through call with different channel types: local = TCH_F, remote = TCH_H
```

how to solve?

It would be nice to see a Channel Requested from MS 2 coming in as TCH/H already, at [1].

Q: Who decides for TCH/F there? Is osmo-bts somehow telling the MS to do TCH/F?

Even though a TCH/F request has come in, lchan_alloc() could still decide to allocate TCH/H anyway at [2].

Q: can rsl_rx_chan_rq() in abis_rsl.c feasibly find out the transaction the channel is requested for?

Then we could look up the first channel's lchan type and ask lchan_alloc() for the same.

```
static int rsl_rx_chan_rq(struct msgb *msg)
{
    [...]
    trans = trans_find_by_callref(bts->network, callref);
    if (!trans)
        return -EIO;
    if (!trans->conn)
        return 0;
```

So, can we find out the callref in rsl_rx_chan_rq()?

#6 - 07/28/2016 02:32 PM - neels

neels wrote:

It would be nice to see a Channel Requested from MS 2 coming in as TCH/H already, at [1].

Q: Who decides for TCH/F there? Is osmo-bts somehow telling the MS to do TCH/F?

It does not seem to be the MS choosing TCH/F, because it's the same if I swap the phones from caller to callee: The caller always picks TCH/H, and the callee then responds to the paging with TCH/F.

So it's not like the Sony phone simply prefers TCH/F. The Samsung replies to the paging with TCH/F just the same.

#7 - 07/28/2016 03:54 PM - neels

workaround that disables TCH/F for dyn TS in <https://gerrit.osmocom.org/603>

#8 - 07/29/2016 05:27 PM - laforge

- Status changed from Feedback to New

- Assignee changed from laforge to sysmocom

- Priority changed from Urgent to Normal

The work-around is ok for now, but let's leave the ticket open until we have a proper fix. Probably after the BSC/MSC split.

#9 - 10/14/2016 11:36 AM - dexter

- Status changed from New to Stalled

Status update:

- GSM 04.08 Version 5.3.0, 10.5.2.8 "Channel Needed" (page 318)
It is not possible to ask the mobile specifically for a TCH_H or a TCH_F, We can ask for TCH_F explicitly, but not for TCH_H. Here we can only ask for "TCH_F or TCH_H". Apart from that the channel needed flag is only a "recommendation" to the mobile. The mobile may ask for a different channel than it was asked for
- Experiment:
It is possible to allocate an arbitrary channel type to the mobile, this means we can freely choose between TCH_H and TCH_F during the channel assignment. I checked this with an experiment and it works. However, it is not possible to recognize the mobile when the rach request

occurs. So we do not know early enough which mobile is requesting the channel, so we can not assign the right channel type.

- Possible solution:

It may be possible to solve the problem by assigning a new traffic channel when we detect that the channel types do not match up. This means the problem is more difficult than I thought. It is probably not that easy to assign a new channel and close the old one. This probably requires the implementation of a state machine.

#10 - 11/03/2016 01:32 PM - neels

dexter wrote:

- Possible solution:

It may be possible to solve the problem by assigning a new traffic channel when we detect that the channel types do not match up. This means the problem is more difficult than I thought. It is probably not that easy to assign a new channel and close the old one. This probably requires the implementation of a state machine.

'handover' could be the right keyword here.

In openbsc branch neels/dyn_pdch_defrag there are some patches from jolly that use handover to defragment lchan usage within the same BTS. Possibly a similar mechanism could be used to change TCH/F to TCH/H?

https://git.osmocom.org/openbsc/log/?h=neels/dyn_pdch_defrag
originally: https://git.osmocom.org/openbsc/log/?h=jolly/dyn_pdch

#11 - 11/05/2016 06:29 PM - laforge

On Thu, Nov 03, 2016 at 01:32:25PM +0000, neels [REDMINE] wrote:

'handover' could be the right keyword here.

It is simply a 'late assignment', one of the most normal procedures in every GSM network: Phone sends RACH, network assigns SDCCH (to save bandwidth/resources), phone sends SETUP, network then check if the callee is available. If it is, network ASSIGNS (not IMM. ASSIGN!) the TCH/F or TCH/H channel. We already do late assignment in OsmoBSC, AFAIK. It is only OsmoNITB which for historical reasons is still doing very early assignment (immediate TCH/F or TCH/H assignment in response to RACH).

--

- Harald Welte <laforge@gnumonks.org> <http://laforge.gnumonks.org/>

=====
"Privacy in residential applications is a desirable marketing option."
(ETSI EN 300 175-7 Ch. A6)

#12 - 11/15/2016 01:41 AM - neels

For the record: We have this workaround in place to disable use of TCH/F in osmo-nitb only: <https://gerrit.osmocom.org/603>. It is planned to add a config item to leave the choice up to the user. Due to implementation details, I'm waiting for <https://gerrit.osmocom.org/1140> before adding the config item -- until then one could patch the nitb code and set 'bsc_gsmnet->dyn_ts_allow_tch_f = true;' near line 313 of openbsc/src/osmo-nitb/bsc_hack.c. This would of course almost certainly cause this issue to re-appear.

#13 - 12/07/2016 01:27 PM - neels

gerrit patch 1140 is now merged to master, so we could now add a VTY config for osmo-nitb to allow/disallow TCH/F on dynamic TCH/F_TCH/H_PDCH timeslots. (My attention is with the VLR though.)

#14 - 02/28/2017 05:59 PM - laforge

- Assignee changed from sysmocom to neels

#15 - 05/31/2017 08:53 AM - msuraev

- Related to Bug #1865: TCH/F_PDCH is not working as TCH/F channel for CS voice call in USRP B210 board (osmo-trx) added

#16 - 05/31/2017 08:57 AM - msuraev

- Related to Bug #1853: validate dynamic TCH/PDCH support in osmo-bts-trx added

#17 - 05/31/2017 10:09 AM - msuraev

- Related to deleted (Bug #1853: validate dynamic TCH/PDCH support in osmo-bts-trx)

#18 - 05/31/2017 10:09 AM - msuraev

- Blocks Bug #1853: validate dynamic TCH/PDCH support in osmo-bts-trx added

#19 - 10/09/2017 02:45 PM - neels

- Related to Bug #1971: TCH/F_TCH/H_PDCH: BSC doesn't allocate TCH/F added

#20 - 10/09/2017 02:49 PM - neels

- Priority changed from Normal to Low

- % Done changed from 100 to 0

having [#1971](#) as subtask and resolving that despite nothing being fixed made this %-done jump to 100 in error. Go back to 0.

Note that in the new split way, separate MSC and BSC talking via SCCP/M3UA, RTP will be routed via the new osmo-mgw, and the plan is to resolve any codec mismatches via transcoding there. Hence, we will most probably not spend any effort on fixing such situations for OsmoNITB.

#21 - 11/27/2017 02:59 PM - neels

- Related to Feature #1606: hand-over for load distribution among BTSs added

#22 - 11/27/2017 02:59 PM - neels

- Related to Feature #1608: various handover improvements, meta-issue added

#23 - 11/27/2017 03:02 PM - neels

the handover code I'm looking at now could provide a solution to this too, it has a Late Assignment mechanism. Remains to be seen whether we have a sane way to tell the BSC that the other call leg is of a different codec ... so actually the transcoding in MGW would be the most general solution.

#24 - 11/27/2017 07:42 PM - laforge

On Mon, Nov 27, 2017 at 03:02:35PM +0000, neels [REDMINE] wrote:

the handover code I'm looking at now could provide a solution to this too, it has a Late Assignment mechanism. Remains to be seen whether we have a sane way to tell the BSC that the other call leg is of a different codec ...

I think the MSC can send at any time during a call a new "ASSIGNMENT COMMAND" over the A interface, and that can contain different channel type / codec information than the previous channel.

#25 - 12/10/2017 08:01 PM - laforge

- Project changed from OpenBSC to OsmoMSC

#26 - 12/23/2017 05:49 PM - laforge

- Category set to A interface (general)

#27 - 09/03/2018 05:21 PM - laforge

ping? Is this now solved after the external handover code was merged?

#28 - 09/03/2018 05:21 PM - laforge

#29 - 09/12/2018 02:12 PM - neels

- Status changed from Stalled to Resolved

- % Done changed from 0 to 100

Hey, what? I already answered to this one a week ago. Somehow must have forgotten to click "Submit"...

The root reason why we ran into mismatching channel types was that we did not do late assignment:

- phone sends a Channel Request for a TCH/F channel
- we used to accept that and assign a TCH/F
- call gets established, second call leg answers to paging
- second call leg gets assigned a voice channel according to codec choices, likely ending up as TCH/H
- mismatch.

Since we do late assignment, we always assign an SDCCH first.

We also more accurately select a voice type by the MSC, BSC and MS supported codec types.

So we will pick our own most desired channel type instead of sticking with the initial Immediate Assignment one.

Also, if we did end up assigning a TCH/* for initial Channel Request and its chan mode mismatches the one we picked, we will assign a different channel now.

It is still possible to end up with mismatching channel types,

thinking: one MS has no TCH/H capability thus gets TCH/F, and the core net config prefers TCH/H so the second phone gets TCH/H, but this is far less likely than it used to be.

Bottom line is, we need transcoding to deal with all of the above issues, and for now we're doing the best we can to avoid mismatches in osmo-bsc.git.

Note: old osmo-nitb still has this problem, since it doesn't do late assignment.