

OsmoPCU - Bug #1940

implement T3141 - Timeout after Send Immediate assign for TBF

02/04/2017 07:23 AM - lynxis

Status: Stalled	Start date: 02/04/2017
Priority: High	Due date:
Assignee: pespin	% Done: 0%
Category:	
Target version:	
Spec Reference: 44.018 44.060	

Description

T3141 is started after sending a immediate assign and stopped when the MS established the TBF.
From 44.018 version 11.7.0 Release 11:
3.5.2.1.5:
"On the network side, if timer T3141 elapses before a successful contention resolution procedure is completed, the newly allocated temporary block flow is released as specified in 3GPP TS 44.060 and the packet access is forgotten."

For DL TBF:
- The pcu uses **Tassign_agch** similar to T3141.
For UL TBF:
- It's missing. But at least T3169 is started (TFI/USF/TBF re-use timer).

3gpp 44.018 describes T3141.
3gpp 44.060 describes T3169.

History

#1 - 02/04/2017 07:24 AM - lynxis

- Spec Reference set to 44.018 44.060

#2 - 02/28/2017 03:14 PM - laforge

- Assignee set to lynxis

#3 - 08/17/2017 06:55 AM - laforge

- Assignee changed from lynxis to sysmocom

#4 - 10/29/2017 06:46 PM - laforge

- Priority changed from Normal to High

#5 - 10/02/2018 03:45 PM - laforge

#6 - 04/15/2019 07:45 AM - laforge

- Assignee changed from sysmocom to lynxis

#7 - 09/25/2019 01:56 PM - pespin

Picking up this one. Any idea on a good default value for timer T3141? I cannot find any default value in specs, it simply says "Its value is network dependent."

#8 - 09/25/2019 01:57 PM - pespin

- Assignee changed from lynxis to pespin

#9 - 09/25/2019 02:01 PM - pespin

I see now the Tassign_agch mentioned in the ticket is X2002 (T=-2002) in current pcu code base. It used to be called T_ASS_AGCH_USEC.

I have been looking at the code and starting some changes but I'm currently stuck not understanding behavior of some parts of the osmo-pcu code.

TS 44.018 states the following for T3141:

When EC operation is not enabled the packet uplink resource is assigned to the mobile station in an IMMEDIATE ASSIGNMENT or an IMMEDIATE PACKET ASSIGNMENT message (if supported) sent in unacknowledged mode on the same CCCH timeslot on which the network has received the CHANNEL REQUEST or the EGPRS PACKET CHANNEL REQUEST message. The network may assign packet resource to the mobile station in an IMMEDIATE PACKET ASSIGNMENT message if the received EGPRS PACKET CHANNEL REQUEST message indicates the support of this assignment message. There is no further restriction on what part of the downlink CCCH timeslot the IMMEDIATE ASSIGNMENT and IMMEDIATE PACKET ASSIGNMENT message can be sent. Timer T3141 is started on the network side.

So iirc, this timer only applies to assignments done through CCCH and not the assignments done through PACCH.

Currently in osmo-pcu, we have timers -2000, -2001 and -2002 using same timer object T0. -2000 and -2001 correspond to PACCH assignment related procedures, while -2002 seems to be more or less what's specified in the spec as T3141.

So my idea is to break current T0 timer object into T3141 and keep PACCH related stuff in T0 (which may actually become T3113 from TS 44.060, I still need to check that).

Some of those changes include:

```
diff --git a/src/bts.cpp b/src/bts.cpp
index df58494..beb6a6e 100644
--- a/src/bts.cpp
+++ b/src/bts.cpp
@@ -81,9 +81,9 @@ static struct osmo_tdef T_defs_pcu[] = {
     { .T=1, .default_val=30, .unit=OSMO_TDEF_S, .desc="BSSGP (un)blocking procedures timer (s)", .val=0 },
     { .T=2, .default_val=30, .unit=OSMO_TDEF_S, .desc="BSSGP reset procedure timer (s)", .val=0 },
     { .T=3190, .default_val=5, .unit=OSMO_TDEF_S, .desc="Return to packet idle mode after Packet DL Assignment on CCCH (s)", .val=0 },
+    { .T=3141, .default_val=200, .unit=OSMO_TDEF_MS, .desc="Time to wait after IMM.ASS confirm until contention resolution procedure times out (ms)", .val=0 },
     { .T=-2000, .default_val=2, .unit=OSMO_TDEF_MS, .desc="Tbf reject for PRR timer (ms)", .val=0 },
     { .T=-2001, .default_val=2, .unit=OSMO_TDEF_S, .desc="PACCH assignment timer (s)", .val=0 },
-    { .T=-2002, .default_val=200, .unit=OSMO_TDEF_MS, .desc="Waiting after IMM.ASS confirm timer (ms)", .val=0 },
     { .T=-2030, .default_val=60, .unit=OSMO_TDEF_S, .desc="Time to keep an idle MS object alive (s)", .val=0 }, /* slightly above T3314 (default 44s, 24.008, 11.2.2) */
     { .T=-2031, .default_val=2000, .unit=OSMO_TDEF_MS, .desc="Time to keep an idle DL TBF alive (ms)", .val=0 },
     { .T=0, .default_val=0, .unit=OSMO_TDEF_S, .desc=NULL, .val=0 } /* empty item at the end */
@@ -601,7 +601,7 @@ int BTS::rcv_imm_ass_cnf(const uint8_t *data, uint32_t fn)
    LOGP(DRLCMAC, LOGL_DEBUG, "Got IMM.ASS confirm for TLLI=%08x\n", tlli);

    if (dl_tbf->m_wait_confirm)
-        T_START(dl_tbf, T0, -2002, "assignment (AGCH)", true);
+        T_START(dl_tbf, T3141, 3141, "assignment (AGCH)", true);

    return 0;
}

diff --git a/src/tbf.h b/src/tbf.h
index fd65b20..b1d1a6c 100644
--- a/src/tbf.h
+++ b/src/tbf.h
@@ -121,9 +121,12 @@ enum tbf_egprs_counters {
    #define LOGPTBF(tbf, level, fmt, args...) LOGP(DTBF, level, "%s " fmt, tbf_name(tbf), ## args)

    enum tbf_timers {
-        /* internal assign/reject timer */
+        /* internal PACCH assign/reject timer */
        T0,

+        /* Time to wait after IMM.ASS confirm until contention resolution procedure times out. */
+        T3141,
+
        /* Wait for reuse of USF and TFI(s) after the MS uplink assignment for this TBF is invalid. */
    };
};
```

```

T3169,

diff --git a/src/tbf.cpp b/src/tbf.cpp
index 3ca39bf..2223caa 100644
--- a/src/tbf.cpp
+++ b/src/tbf.cpp
@@ -630,7 +630,7 @@ bool gprs_rlcmac_tbf::timers_pending(enum tbf_timers t)
     return osmo_timer_pending(&Tarr[t]);

     /* we don't start with T0 because it's internal timer which requires special handling */
-   for (i = T3169; i < T_MAX; i++)
+   for (i = T3141; i < T_MAX; i++)
         if (osmo_timer_pending(&Tarr[i]))
             return true;

@@ -658,6 +658,7 @@ static inline void tbf_timeout_free(struct gprs_rlcmac_tbf *tbf, enum tbf_timers

#define T_CBACK(t, diag) static void cb_##t(void *_tbf) { tbf_timeout_free((struct gprs_rlcmac_tbf *)_tbf, t,
diag); }

+T_CBACK(T3141, true)
T_CBACK(T3169, true)
T_CBACK(T3191, true)
T_CBACK(T3193, false)
@@ -707,6 +708,9 @@ void gprs_rlcmac_tbf::t_start(enum tbf_timers t, int T, const char *reason, bool
    case T0:
        Tarr[t].cb = tbf_timer_cb;
        break;
+   case T3141:
+       Tarr[t].cb = cb_T3141;
+       break;
    case T3169:
        Tarr[t].cb = cb_T3169;
        break;

```

However, in callback of T0 (tbf_timer_cb(): tbf->handle_timeout()), I still see some references to GPRS_RLCMAC_FLAG_CCCH and I'm unsure what to do with those (what is that used for).

Also, I still need to find out where the new T3141 needs to be disarmed. According to TS 44.018:

3.5.2.1.4 Packet access completion

The one phase packet access procedure is completed at a successful contention resolution. The mobile station has entered the packet transfer mode. Timer T3141 is stopped on the network side.

So I'd expect that some of the t_stop(T0,...) probably needs to become t_stop(T3141,...), but I'm really unsure at this point. There's currently 2 places where T0 is stopped:

```

src/pdch.cpp:337:         new_tbf->t_stop(T0, "control acked (DL-TBF)"); <--- rcv_control_ack().
src/tbf.cpp:1321:        new_dl_tbf->t_stop(T0, "assignment (DL-TBF)") <--- create_dl_ass(), this looks like simply dropping all previous assignments?

```

So I have the feeling I'd need to check in pdch.cpp:337 where was the assignment done (CCCH or PACCH) and depending on that stop T0 or T3141, but I'm not really sure right now.

Mentioned changes can be found in this branch: <https://git.osmocom.org/osmo-pcu/log/?h=pespin/t3141>

#11 - 09/26/2019 10:55 AM - pespin

- Status changed from New to In Progress

#12 - 10/08/2019 11:44 AM - pespin

- Status changed from In Progress to Stalled

#13 - 12/05/2019 04:09 PM - pespin

After a bit more of investigation and implementing some more test cases in osmo-pcu, it seems our T0 (handling different timers, such as T-2002 or X2002 in this case) actually states the time osmo-pcu should wait between first sending the IMM Assign to the BTS over PCUIF and the time it can start sending the DATA for which the IMM Ass was requested.

So basically:

- IMM Ass is sent, T0 (value X2002) is armed

- X2002 triggers, Downlink data is sent on RTS req from BTS).

Before X2002 triggers, the TBF is not in flow state and upon RTS req from BTS it will send dummy block.