

B5G_UE - 错误 #5583

版本R1.1.0_T01，DD口，ue1和ue2多次脱网及再在网后，ue1不能发出hello广播

2026-07-24 16:44 - 李 常

状态:	反馈	开始日期:	2026-07-24
优先级:	高	计划完成日期:	
指派给:	李 常	% 完成:	0%
类别:		预期时间:	0.00 小时
目标版本:		耗时:	0.00 小时
描述			
高概率发生。			
相关的问题:			
关联到 B5G_UE - 错误 #5589: 【B5g_ue】B5Gue_R1.1.0_T1,DD移动测试，没有恢...		反馈	2026-07-27

历史记录

#1 - 2026-07-24 16:45 - 李 常

- 状态 从 新建 变更为 进行中
- 指派给 被设置为 李 玮璇

#2 - 2026-07-24 16:49 - 李 玮璇

从日志看，调整天线多次释放建立流程后，出问题时nc没有日志打印，threadL3无法正常进入业务，像是while已经无法触动下一轮，但是threadL3的核却一直在小号cpu，dpdk的timer管理陷入某种异常无法循环业务导致nc无法发送hello
此情况之前有过偶现
[Switching to thread 11 (Thread 0x7f73ffd900 (LWP 980))]
#0 0x0000000000dae7c0 in __rte_timer_reset ()
(gdb) bt
#0 0x0000000000dae7c0 in __rte_timer_reset ()
#1 0x0000000000daf464 in rte_timer_manage ()
#2 0x00000000008707e4 in wnTimerManage () at fwk/ngTimer/csrc/wn5gNrPsTimerApi.c:82
#3 0x0000000000898ba0 in threadL3 (ueCb=0x384a5000) at src/l3/rrc/csrc/yzL3Mani.c:597
#4 0x0000000000dbd0d8 in eal_thread_loop ()
#5 0x0000007f959da7e4 in start_thread (arg=0x7fd6c2ce4f) at pthread_create.c:486
#6 0x0000007f9586070c in thread_start () at ../sysdeps/unix/sysv/linux/aarch64/clone.S:78
(gdb) quit

#3 - 2026-07-24 16:50 - 李 玮璇

- 文件 perf.data 已添加

#4 - 2026-07-25 12:21 - 李 常

- 优先级 从 一般 变更为 高

#5 - 2026-07-27 08:52 - 李 玮璇

差堆栈__rte_timer_reset内部卡住，查资料有可能是跨核操作定时器导致核对应的定时器链表损坏，需要先修改rrc和pdcp、sdap定时器机制后再做测试和排查

#6 - 2026-07-31 11:37 - 李 玮璇

经过这几天增加日志测试，发现触发场景在快速多次反复重建链路时，sac单次定时器快到期时，触发鉴权开始对该定时器进行操作场景下，导致以下两种问题现象：
【问题现象一】threadL3卡在__rte_timer_reset
(gdb) p *((struct rte_timer *)0x1006d788)
\$1 = {expire = 14744347765, sl_next = {0x1006d788, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0}, status = {{state = 3, owner = 2}, u32 = 131075}, period = 0, f = 0x969c80 <wnSacAuthTimerMsgSend>, arg = 0x187c69480}

```
1.Children  Self Command Shared Object Symbol
2..... #
   60.27%   60.27% threadL3 psarmapp      [.] 0x00000000009af880
   39.73%   39.73% threadL3 psarmapp      [.] 0x00000000009af88c
Line 74 of "src/l2/mac/csrc/cm_mblk.c" starts at address 0x9af884 <cmAllocEvtnt+84>
and ends at 0x9af89c <cmAllocEvtnt+108>.
(gdb) info line *0x00000000009af880
```

Line 72 of "src/l2/mac/csrf/cm_mblk.c" starts at address 0x9af880 <cmAllocEvtnt+80> and ends at 0x9af884 <cmAllocEvtnt+84>.

【问题现象二】hello定时器丢失，但是线程不卡死，sac定时器还能处理

基本确认在反复重建下，有些场景会导致dpdk定时器异常。目前确认触发业务相似操作中。。。只发现sac定时器快到期时触发鉴权开始重置定时器后容易出问题，但是因不涉及跨核同步操作所以无法确认根因；
0731在西安分享后，决定从以下方向定位：
1、在线程循环打印中把hello、sac的回调函数地址打印出来
2、问题发生时业务相同性
3、操作隔离，定时器隔离
4、rte_timer_dump_stats(stderr);挪在外部周期打印，关注定时器数量

#7 - 2026-08-04 13:58 - 李玮璇

- 状态从进行中变更为反馈
- 指派给从李玮璇变更为李常

【触发场景】鉴权的主动方（devId小的）多次反复重发快速删除和重建rb，2s内发起第二次鉴权开始；在该场景有错误重置定时器的操作，代码在25年开始就有，因最近开始频繁且快速2s内删除建立rb的场景测试暴露出来
【问题根因】dpdk的定时器机制下，sac重置一个PENDING状态的定时器使用了rte_timer_init；该操作只修改了定时器的“身份证”（status），将其强制改为STOPPED，但完全没有通知处理跳表，导致“状态-链表”不一致内部逻辑混乱
【解决方案】先stop再reset（目前使用），或者直接使用reset，这两个函数有操作处理状态和跳表
解决版本：已合入trunk和自同步分支
SHA-1: f1d8e1be31abbc392a199fc8537e454914f5e8fa

- Merge branch '376-test-analyze-the-dpdk-timer-problem' into 'dev_d2d_trunk'

#8 - 2026-08-05 09:17 - 李玮璇

- 关联到 错误#5589: 【B5g_ue】B5Gue_R1.1.0_T1,DD移动测试，没有恢复dd连接 已添加

文件

perf.data	3.53 MB	2026-07-24	李玮璇
-----------	---------	------------	-----