

研发产品测试 - 错误 #2657

【16P_pre1T1】E500模拟128用户运行60km/h移动速度并灌包，DU出现内存问题挂死

2025-01-06 14:29 - 郝雷

状态:	已解决	开始日期:	2025-01-06
优先级:	普通	计划完成日期:	
指派给:	郝雷	% 完成:	0%
类别:		预期时间:	0.00 小时
目标版本:		耗时:	0.00 小时
问题归属:	DU	FPGA板卡类型:	
发现问题版本:	Rel_2.1.16P	CPU类型:	
目标解决问题版本:	Rel_2.1.16P		

描述

E500模拟128个用户，接入后所有用户模拟60km/h的fast fading模型，上下行灌包2M每用户。

由于移动时SIR降低，多个UE掉线重接，PHY上出现APIErr，

1月 06 10:49:22 yzrt run.sh³²⁵⁵: DU2PHY_API_USE_NULL_DATA: phyInstance⁰ ApiErr_NUM²⁰

1月 06 10:49:22 yzrt run.sh³²⁵⁵: [FATAL][STOP_PHY] DU2PHY_API_USE_NULL_DATA: phyInstance⁰ ApiErr_NUM²¹

1月 06 10:49:22 yzrt run.sh³²⁵⁵: [FATAL][STOP_PHY] DU2PHY_API_USE_NULL_DATA: phyInstance⁰ ApiErr_NUM²¹

DU上答应内存错误，

1月 06 11:07:31 yzrt run.sh⁹⁷⁸⁵: cmGetMemBlkSetForAlloc failed 2

几分钟后DU挂死。

历史记录

#1 - 2025-01-07 15:03 - 匿名用户

场景：高速移动，pdch聚合等级自适应开关打开，大上行子帧配比

UE的cqi都很差，选取的pdch聚合等级都是4，根据大上行的子帧配比，每tti调度4用户计算需要的cce总数为 $4*4*10 =$

160CCE，而大上行子帧配比单符号的CCE总数为 $3*32 =$

96CCE，CCE总数远远不够，日志中有很多的pdch资源分配失败，资源分配失败后可能存在未及时释放的问题，导致后续的内存耗尽，DU挂死。

修改coreset1占用符号数为2符号后，CCE总数为 $3*32*2 = 192CCE >$

160CCE，测试30分钟，未出现内存耗尽和挂死现象，但是出现高误码和低速率的问题，

测试单UECORESET1配置2symbol，发现同样存在高误码和低速率的问题

关联问题单#2660

#2 - 2025-01-07 15:04 - 匿名用户

该问题暂等解决#2660后进一步观察

#3 - 2025-01-15 10:04 - 匿名用户

- 文件6xaipEKWar.jpg 已添加

- 文件8K8DX4Vsk3.jpg 已添加

- 文件7lmmq0aRE4.jpg 已添加

```

=====du_mem_size:[2025-01-03 18:45:03:679]=====
size : 5      6      13     16     24     25     26     27     28     29     30     31     32     33     34     35     36     37     38     39     40     41     42     43
count : 2      401    11     2686   10092  13     45     37     45     31     36     21     1055  28     31     27     1803  32     31     33     39731  24     32     36
=====
=====show_globRegCb_info :=====
region:poolId  bktIdx  size    validBkts  freeBkts  bktAllocErr1:2:3  bktFreeErr1:2:3  blkAllocErr1:2  blkFreeErr1:2  blkBound-A:F  blkInvalid-A:F  blkDup-A:F
0:0            0        256     2076       4268     0:0:0           0:0:0           0:0           0:0           0:0           0:0           0:0
0:1            1        2048     3205       858      0:0:0           0:0:0           0:0           0:0           0:0           0:0           0:0
0:2            2        8192     0          74       0:944:21        0:0:0           944:0         0:0           0:0           0:0           0:0
0:3            3        16384    64         14       0:0:0           0:0:0           0:0           0:0           0:0           0:0           0:0
0:4            4        76800    34         30       0:0:0           0:0:0           0:0           0:0           0:0           0:0           0:0

```

查看DU memory_log之后发现大小为8192的内存块耗尽且出现分配失败的统计

```

56 1160 1192 1208 1268 1299 1461 1493 1524 1784 1816 2000 2056 2632 2704 3200 3556 3560 4672 6144 7040 8200 9232 11696 11704 12664 14620 15252 16384 16408 20480 22916 25192 2
1 16 2794 1 1 3 1 33 1 219 72209 222 8235 125 64 1 1 130 219 1 659 500 1 1 125 1 125 378 2 1 1 20 1

```

进一步定位发现在使用8192的内存中size为2632的count过多，查看前后的统计日志，发现该size大小的内存只申请，未释放，所以怀疑该内存块大小的内存存在泄漏

```

[01-03 18:45:04.982][DBG_26]Missing byte segment's SN[1573] soStart[0] soEnd[485] UEID[17120] CELLID[1]
[01-03 18:45:04.982][DBG_07]rgSCHLvlTomCrcInd: UL CRC, cellId[1] ueId[17052] crcFail[0] procId[6] pdccchTime[783 0] puschTime[783 0] ulNackCounter[222],ulAckCounter[2333],ulCurrentBler[8]
[01-03 18:45:04.982][DBG_07]rgSCHLvlTomCrcInd: nrti[17052] harqId[6] numLyrs[1] iMcs[9] rbStart[4] rbNum[66] tbSize[1441] crcFail[0] ucAckNack[1] txCnt[2] rv[2] pusch[783 3] pdccch[783 0]
[01-03 18:45:04.982][DBG_07]rgSCHLvlTomCrcInd: UL CRC, cellId[1] ueId[17041] crcFail[0] procId[12] pdccchTime[783 0] puschTime[783 3] ulNackCounter[228],ulAckCounter[2336],ulCurrentBler[8]
[01-03 18:45:04.982][DBG_07]rgSCHLvlTomCrcInd: nrti[17041] harqId[12] numLyrs[1] iMcs[10] rbStart[70] rbNum[66] tbSize[1441] crcFail[0] ucAckNack[1] txCnt[2] rv[2] pusch[783 3] pdccch[783 0]
[01-03 18:45:04.982][DBG_07]rgSCHLvlTomCrcInd: UL CRC, cellId[1] ueId[17120] crcFail[0] procId[10] pdccchTime[783 0] puschTime[783 3] ulNackCounter[227],ulAckCounter[2349],ulCurrentBler[8]
[01-03 18:45:04.982][DBG_07]rgSCHLvlTomCrcInd: nrti[17120] harqId[10] numLyrs[1] iMcs[10] rbStart[136] rbNum[66] tbSize[1441] crcFail[0] ucAckNack[1] txCnt[1] rv[0] pusch[783 3] pdccch[783 0]
[01-03 18:45:04.982][DBG_07]rgSCHLvlTomCrcInd: UL CRC, cellId[1] ueId[17064] crcFail[0] procId[2] pdccchTime[783 0] puschTime[783 0] ulNackCounter[223],ulAckCounter[2351],ulCurrentBler[8]
[01-03 18:45:04.982][DBG_07]rgSCHLvlTomCrcInd: nrti[17064] harqId[2] numLyrs[1] iMcs[9] rbStart[202] rbNum[67] tbSize[1473] crcFail[0] ucAckNack[1] txCnt[1] rv[0] pusch[783 3] pdccch[783 0]
[01-03 18:45:04.982][DBG_07]DEBU: UE[17052] Decoded Short BSR, Bsr[31]
[01-03 18:45:04.982][ERR_07]cmGetMemBlkSetForAlloc: failed 2, region[0] listValidBktSetCnt[65] listFreeBktSetCnt[9] crntMemBlkSetElem[0x4f15d70:0] first[0x4f15d30:0] last[0x4f15d70:0]
[01-03 18:45:04.982][ERR_07]rg_ucl.c:811 cmDynAlloc failed, region[0] size[2632],bktSize[8160],bktIdx[2]
[01-03 18:45:04.982][ERR_07]rgTOMUclProcMsg:rgAllocShrablSBuf FAIL
[01-03 18:45:04.982][ERR_07]RgLiTfuDatInd:DBG_CELLID[1] Unable to handle Data Indication CRNTI[17052]
[01-03 18:45:04.982][DBG_07]DEBU: UE[17041] Decoded Short BSR, Bsr[31]
[01-03 18:45:04.982][ERR_07]cmGetMemBlkSetForAlloc: failed 2, region[0] listValidBktSetCnt[65] listFreeBktSetCnt[9] crntMemBlkSetElem[0x4f15d70:0] first[0x4f15d30:0] last[0x4f15d70:0]
[01-03 18:45:04.982][ERR_07]rg_ucl.c:811 cmDynAlloc failed, region[0] size[2632],bktSize[8160],bktIdx[2]
[01-03 18:45:04.982][ERR_07]rgTOMUclProcMsg:rgAllocShrablSBuf FAIL
[01-03 18:45:04.982][ERR_07]RgLiTfuDatInd:DBG_CELLID[1] Unable to handle Data Indication CRNTI[17041]

```

查看du_log后找出代码中具体的内存泄漏的结构体为RgRguDedDatInd；

#4 - 2025-01-21 14:01 - 匿名用户

- 文件 WO5fdFw2bN.jpg 已添加
- 文件 z2Ali0FYYO.jpg 已添加

走读代码发现RgRguDedDatInd确实存在未释放的场景，但是没有异常日志打印(未释放的地方都有异常日志)，发现日志中在pdccch分配失败的时候有时序错乱的打印

```

Search "SCH slotDiff over" (6 hits in 1 file of 1 searched)
D:\LOG\1月\128UE移动60km\128UE移动60km\yzrt_du_20250103184452.log (6 hits)
Line 86151: [01-03 18:44:52.726][ERR_07]rgSCHLvlTomTtiInd: SCH slotDiff over 2, g_slotDiffCount[900] lvl1[579 4] lvl2[579 6] slotDiff[2]
Line 499732: [01-03 18:44:54.523][ERR_07]rgSCHLvlTomTtiInd: SCH slotDiff over 2, g_slotDiffCount[910] lvl1[759 7] lvl2[759 9] slotDiff[2]
Line 947805: [01-03 18:44:56.469][ERR_07]rgSCHLvlTomTtiInd: SCH slotDiff over 2, g_slotDiffCount[920] lvl1[954 6] lvl2[954 8] slotDiff[2]
Line 1307238: [01-03 18:44:58.407][ERR_07]rgSCHLvlTomTtiInd: SCH slotDiff over 2, g_slotDiffCount[930] lvl1[87 14] lvl2[87 16] slotDiff[2]
Line 1993756: [01-03 18:45:01.163][ERR_07]rgSCHLvlTomTtiInd: SCH slotDiff over 2, g_slotDiffCount[940] lvl1[385 17] lvl2[385 19] slotDiff[2]
Line 2873161: [01-03 18:45:04.823][ERR_07]rgSCHLvlTomTtiInd: SCH slotDiff over 2, g_slotDiffCount[950] lvl1[767 7] lvl2[767 9] slotDiff[2]

```

内存耗尽时因为在聚合等级都选择4的场景下，pdccch很多分配失败，因为当前tti的某个UE的pdccch分配失败的话会继续从重传/新传队列中取下一个UE分配pdccch，因为资源有限的原因，pdccch在当前tti一直分配失败，耗时过多，导致的时序错乱，从而导致用户面数据未及时释放或出现坏内存，引起内存耗尽

参照下行的处理方式，上行和下行的pdccch连续分配失败达到一定次数之后不再继续分配，避免pdccch连续分配失败，反复分配造成lvl1和lvl2的时序错乱。

该修改属于针对pdccch资源不足，分配失败的鲁棒性保证。

pdccch的分配方式修改跟踪在问题单#2727和#2660

#5 - 2025-02-12 09:45 - 匿名用户

- 状态从 新建 变更为 转测试
- 指派给从 匿名用户 变更为 郝雷

和入#2660 #2727后该问题不在出现；

代码落入17P

#6 - 2025-02-13 18:29 - 郝雷

- 状态从 转测试 变更为 已解决

文件

6xaipEKWar.jpg	101 KB	2025-01-15	匿名用户
8K8DX4Vsk3.jpg	59.3 KB	2025-01-15	匿名用户

7lmmq0aRE4.jpg	391 KB	2025-01-15	匿名用户
WO5fdFw2bN.jpg	891 KB	2025-01-21	匿名用户
z2Ali0FYyO.jpg	177 KB	2025-01-21	匿名用户