H168N_TTNET版本红郡拷机内存耗尽问题

【作者】　　邵存金　邱晓筱

【时间】　　2012-11-29

【概述】　关于H168N_TTNET内存耗尽问题，本文从协议栈和无线驱动方面分析了产生内存耗尽的可能原因，无线驱动着重分析了发包队列长度以及skb_copy代码。

1　故障描述

【环境】

红郡拷机测试

H168N_TTNET

【故障描述】

单板恢复出厂默认值后进行配置

1.三条连接，外网PPPOE拨号连接、dhcp IPTV连接、dhcp TR069连接

2.播放三路IPTV，一路32M，两路2M。

3.笔记本和台式进行无线连接BT下载。总下载速率在2.5MB左右

TT-11.28.zip

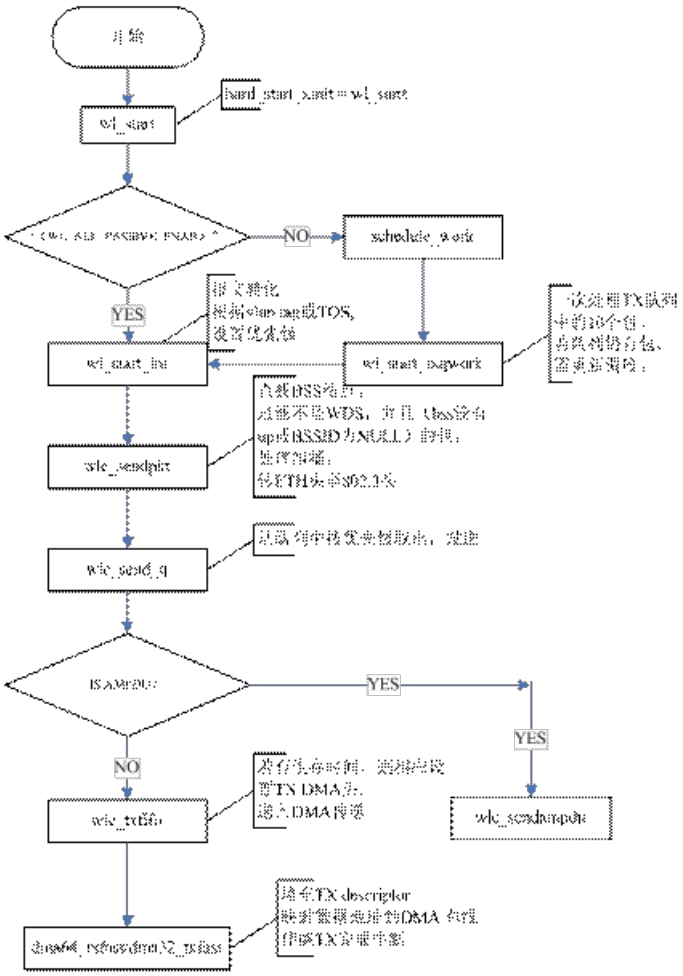
2　查证过程

日期【2012-11-29】

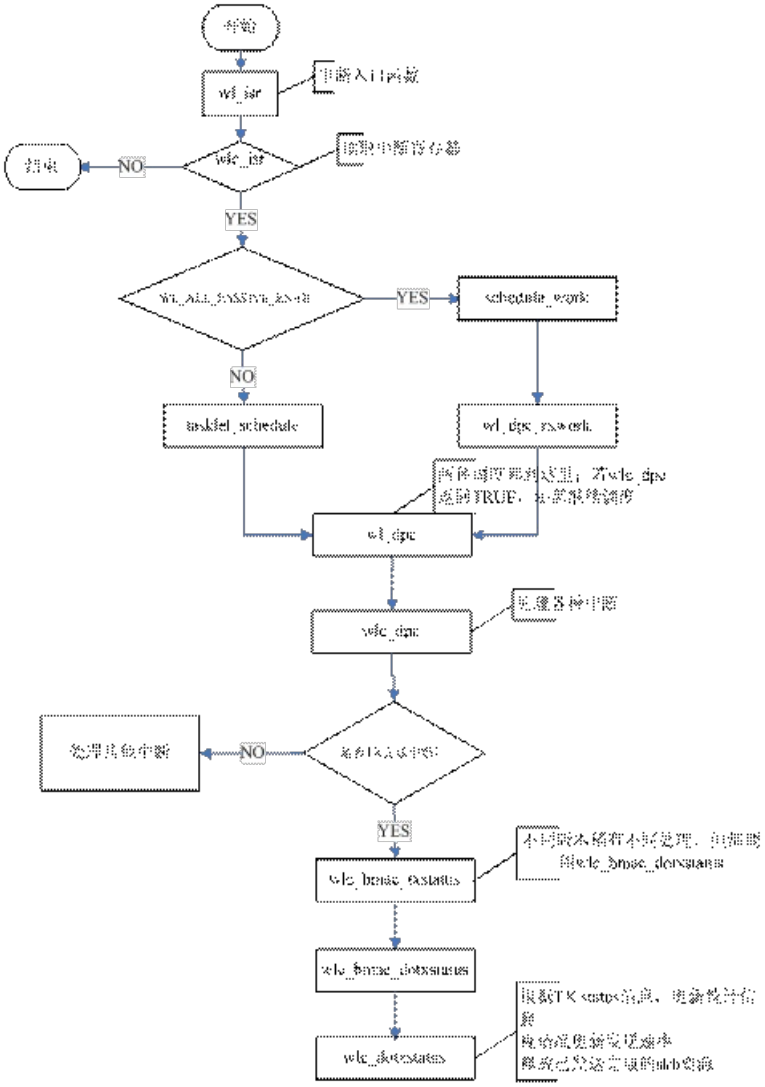
由于该故障暂时没有定位到是属于哪个模块出现问题，今天根据之前提出的两点疑问，着重走查了无线驱动发包函数代码：

疑问1：协议栈的skb到达无线驱动发包，发送完成后，这个skb是否有释放，在哪里释放的？

答案：发送完成后的skb在无线驱动里是有释放的，在TX进入到dma64_txfast()函数中，填充TX descriptor，映射数据地址到DMA 总线，使能TX完成中断。TX完成中断使能后，根据TX status信息，更新统计信息，视情况更新发送速率，释放已发送完成的skb资源。具体代码流程如下：



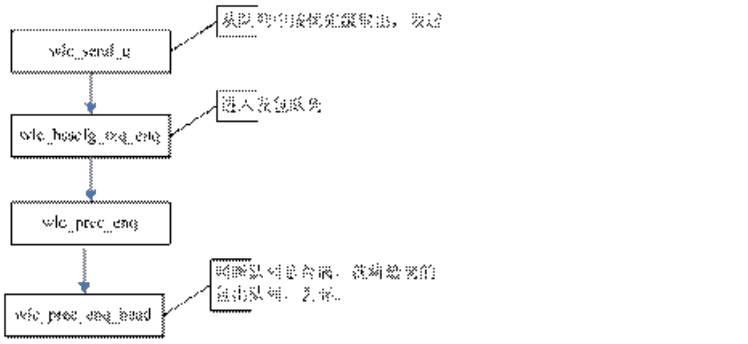
TX完成中断使能后：



疑问2：skb进入无线驱动后，发包队列有长度限制没有？如果队列满了，怎么处理？

答案：当PKT送入发包队列中是有判断队列满的操作的，如果队列满了，则将队首的PKT出队列，并释放。另外，对于入队列失败的包，会立即释放。

详细代码流程如下：




```
/* Free dequeued packet */
if (dqp != NULL) {
    PKTFREE(wlc->osh, dqp, TRUE);
}

return TRUE;
```

入队列失败的包立即释放：

```
if (BSS_TX_SUPR(cfg)) {
    ASSERT(!(wlc->block_datafifo & DATA_BLOCK_TX_SUPR));
    if (wlc_bsscfg_txq_enq(wlc, cfg, pkt[0], prec))
        PKTFREE(wlc->osh, pkt[0], TRUE);
    continue;
}
```

```
bool
wlc_bsscfg_txq_enq(wlc_info_t *wlc, wlc_bsscfg_t *bsscfg, void *sdu, uin
{
    /* Caller should free the packet if it cannot be accomodated */
    if (!wlc_prec_enq(wlc, bsscfg->psq, sdu, prec)) {
        WL_P2P(("wl%d: %s: txq full, frame discarded\n",
                wlc->pub->unit, __FUNCTION__));
        WLCNTINCR(wlc->pub->_cnt->txnobuf);
        return TRUE;
    }

    return FALSE;
}
```

日期【2012-11-30】

邱晓筱：

所里停电，将版本拿到红郡拷机，3路IPTV（30M）+4个STA连无线下载，未限制任务数。（邱晓筱）

日期（2012-12-01）

邱晓筱：

拷机24小时单板正常，未复现。

日期（2012-12-03）

邵存金：

今天，无线驱动主要想到两点：

1. 根据开发经理跟南京沟通的结果，刘昕颖科长建议我们加一些打印，看无线驱动skb_copy的数量和最终释放的数量是否一致。这个主要通过一个全局变量作为skb copy计数器，在无线发包函数中skb copy成功后，将该计数加1，在skb free的地方将该计数器减1，确认最终该值是否会为0，即表示skb copy的报文，最终都会被释放。在实验室环境验证，通过打印确认，无线发包开始时copy 的skb，在经过发包流程后，最终会被释放。计数器值会为0。
2. 合入121116_613001707540_王磊_H118N 联通 内存不足时对于数据包缓存数目的打印入库_无线模块.7z，在无线驱动中增加wl ampdu_txq命令查看当前发送队列中pkt数目以及最大长度。11n模式下，由于支持ampdu功能，每个STA有单独的发包缓存队列。非11n模式下，所有的STA公用一个发包缓存队列。针对这两种情况，在代码中做了相应出现。最终命令实现打印如下:

```
# wl ampdu_txq
--->>current wl_txq_len=0,wl_txq_max=993, skb copy count =51
---->MAC=F0:7D:68:F8:9C:43,scb_ampdu->txq_length=44,scb_ampdu->max_txq_length=4096, skb copy count =59
---->MAC=C8:BE:19:F3:B1:C2,scb_ampdu->txq_length=0,scb_ampdu->max_txq_length=4096, skb copy count =77
---->MAC=18:3D:A2:05:1C:3C,scb_ampdu->txq_length=1,scb_ampdu->max_txq_length=4096, skb copy count =84
---->MAC=A0:88:B4:8B:47:E8,scb_ampdu->txq_length=3,scb_ampdu->max_txq_length=4096, skb copy count =95
0
```

日期（2012-12-06）

邵存金：

由于H168N V1.1通用版（trunk）没有出现该问题，将上述两点分析，着重对比无线驱动代码：

1. Skb_copy()代码，通用版与TTNET版本有差异，有可能引起内存泄露。

```
{
    struct sk_buff *orig_skb = skb;
    skb = nbuff_xlate((pNBuff_t )skb);
    if (skb == NULL)
    {
        nbuff_free((pNBuff_t) orig_skb);
        WLCNTINCR(wl->pub->_cnt->txnobuf);
        /*120523_613001384430_周娟娟_H368N TR069无线参数增补
        WLCNTINCR(wl->pub->_bss_cnt[bsscfg->_idx].txnobuf);
        return 0;
    }
```

2. 通过分析TTNET版本AMPDU队列长度为4096，对比无线驱动发现，H168N V1.1通用版的AMPDU队列长度是1024，如果CPU很忙，无线驱动没有及时将AMPDU队列缓存的报文发送出去，无线驱动中会缓存较多的AMPDU缓存报文，且11n模式下，每个STA单独对应一个AMPDU队列，这样导致无线驱动中缓存的报文太多，占用大量的内存，有可能导致内存耗尽。
3. 通过上面两点分析进行代码同步，准备版本拷机。

日期（2012-12-06）

邵存金：

通过周末两天的拷机验证，内存耗尽的故障没有出现。

3 总结

通过该故障的分析，总结如下：

1. 对于多个不同版本，有的版本出现问题，有的版本没有出现，最直观的方法就是对比代码。由于H168N涉及的项目较多，无线驱动应该及时同步，保持一致，这样既有利于故障的同步，也有利于代码的维护。至于驱动升级，我之前较为被动，以为这是开发经理的计划，如果没有出现问题，开发经理一般也不会想到升级驱动。所以，后续版本驱动升级，应该更加主动的向开发经理提出，保持版本一致。
2. 对于碰到问题，要发散，可以头脑风暴多想想相关点，有时候这也是解决疑难问题的一种捷径。内存耗尽不一定就是因为内存泄露引起，也有可能是系统内某个队列值过大，在极限情况下占有内存过多所导致。
3. 这个问题不排除其他代码有内存泄漏或占有内存过大，因为无线驱动代码没有申请4KB大小的内存，从slabinfo信息来看，申请4KB内存的数目很大。目前只是分析了无线驱动。后续继续观察。