

## 故障现象：

1. 在931DII中，发送100Mbps的下行流量，插拔DSL线，重新建链后，Modem下行业务不通。
2. 在H108N3.0项目中，播放IPTV节目一个晚上之后，重启Modem，IPTV无法继续播放，Modem也无法拨号。偶然出现。
3. 在H108N3.0项目中，下行发送30Mbps流量，重启Modem，下行业务不通。偶然出现。

## 关联版本：

CSP平台的各个版本都可能出现该问题。

931DII - BCM4.02

H108N3.0 - BCM4.06

## 故障原因：

当DSL 重新插上后，DMA打开，由于有大流量，所以收包BD 环很快被灌满，并且有中断，但是ifconfig ptm0 up 滞后，这个时候虽然进入了中断，但是dev 的一些标志位没设置上，不符合收包条件，中断退出，然后ifconfig ptm0 up ，这个时候中断可以被正常处理了，但是BD环已经满了，DMA往里放不进去包了，再也没中断产生了。形成了一个局面：DMA 放不进去包，没有中断，而收包任务因为没有中断也没法去收包清空BD环，陷入了死锁。

```
bcmxtmrt_rxISR()
{
    for( i = 0; i < MAX_DEV_CTXS; i++ )
    {
        if( (pDevCtx = pGi->pDevCtxs[i]) != NULL &&
            pDevCtx->ulOpenState == XTMRRT_DEV_OPENED ) //ifconfig nas1up延迟，导致在进入中断时，这里的Dev状态还是CLOSED，无法对包进行处理，直接退出。
        {
            /* Device is open. Schedule the poll function. */
#if LINUX_VERSION_CODE >= KERNEL_VERSION(2,6,30)
            napi_schedule(&pDevCtx->napi);
#else
            netif_rx_schedule(pDevCtx->pDev);
#endif

            ((volatile DmaChannelCfg *) pRxDma)->intStat = DMA_BUFF_DONE;
            pGi->ulIntEnableMask |=
                1 << (((UINT32) pRxDma - (UINT32)pGi->pRxDmaBase) /
                    sizeof(DmaChannelCfg));
            ulScheduled = 1;
        }
    }
}
```

broadcom没有这个问题，原因是应用程序和我们不一样，他的DMA打开和ptm0 up 能收包之间的时间间隔可能很短，所以没这个问题。

## 解决方法：

将ptm0 up 的位置提到DMA打开之前，现在没有问题了。

刚才和黄忠华讨论了一下，觉得加一个定时器也是可行的，在Modem里面定时地去处理BD环。类似与XTM的发包的定时器。只是有些浪费，因为定时器肯定只会在建链的时候用到一次。

## bcmxtmrt.c::doLinkUp()

```
if( pGi->ulDrvState == XTMRRT_INITIALIZED )
{
    if( pDevCtx->ulAdminStatus == ADMSTS_UP )
    {
        /*modified by liujc and lijinh, open the ptm0 dev early*/
        rtnl_lock();

        dev_open(pDevCtx->pDev); //在打开中断之前，提前打开接口做好收包的准备工作

        rtnl_unlock();
    }

    /* Enable receive interrupts and start a timer. */
}
```

```

for( i = 0, pRxDma = pGi->pRxDmaBase; i < MAX_RECEIVE_QUEUES;
    i++, pRxDma++ )
{
    if( pGi->RxBdInfos[i].pBdBase )
    {
        BcmHalInterruptEnable(SAR_RX_INT_ID_BASE + i);
        pRxDma->cfg = DMA_ENABLE;
    }
}

```

## 定位手段：

### 1. 查看DSL的收发信元计数

```
xdslctl info -show
```

	Down	Up
Total Cells:	22478800	961310
Data Cells:	266	872 //我们需要关注Data Cells
Drop Cells:	0	
Bit Errors:	0	0

### 2. 查看XTM层的计数

```
xmctl intf --stats
```

atm/ptm interface statistics for port 0

in octets	17784
out octets	63282
in packets	150
out packets	234 //需要关注收发包的个数和长度。
in OAM cells	0

这里需要注意，我们的Modem中，一个信元的长度是52字节，其中48字节为有效载荷，4字节为头部。按照标准，应该是5字节的头部。

### 3. 查看中断计数

```
cat /proc/interrupts
```

```

CPU0
7: 240079 BCM63xx timer
51: 9 BCM63xx brcm_51
52: 0 BCM63xx brcm_52

```

在bcmxtmrt.c:: bcmxtmrt\_rx\_isr()函数中，我们看到收包中断为SAR\_RX\_INT\_ID\_BASE+port。通过分析，SAR\_RX\_INT\_ID\_BASE为51，port为0，所以对应的中断号为51。

在出现故障时，可以看到这个中断号的计数一直没有增长过。

### 4. 查看接口计数

```

nas1 Link encap:Ethernet HWaddr 02:10:18:01:00:03
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:50 errors:0 dropped:0 overruns:0 frame:0
TX packets:54 errors:0 dropped:0 overruns:0 carrier:54
collisions:0 txqueuelen:1000
RX bytes:3532 (3.4 KiB) TX bytes:4384 (4.2 KiB)

```

Tx计数表示包将被发到硬件，carrier计数表示包被硬件发出去的个数。

### 5. 查看包打印

```
echo 1 > /proc/log4k/packet_nas
```

```
echo 0 > /proc/log4k/packet_nas
```

可以看到WAN口实际的收发包情况，并将每个包的前60个字节