

STM32F2x7 以太网 FreeRTOS 下驱动问题的说明

STM32F2x7 FreeRTOS 下网络压力测试错误无法恢复的问题

一. 前言

很多朋友反映 ST 官网的例程 `stsw-stm32060`（也就是 STM32F2x7 的以太网例程）中的 FreeRTOS 驱动有问题，表现为在网络压力测试下向 STM32 进行 ping 测试，过段时间后会出现无法恢复的网络延时，甚至出现 ping 不通（网络不通的情况，注意不是超时）。

事实上，ST 的工程师已经发现了这个 bug，他们在 `stsw-stm32070`（STM32F4x7 的以太网例程，不过 STM32F2x7 还没有更新这个驱动）中修复了这个 bug。但是许多朋友并不知道这件事，这篇文档就是对这个问题进行一些说明，另外将新的驱动移植到了 STM32F2x7 中（暂时就叫做 `STM32F2x7_ETH_LwIP_V1.2.0(RC)`）:-D。

二. Stsw-stm32060 中的问题产生原因

解释这个问题前，首先要对 ST 的驱动程序进行一些说明，以太网的数据是通过 FIFO 送入 DMA 缓冲区的，lwip 的底层输入函数（`ethernetif_input` 进程）将数据从 DMA 缓冲区拷贝到协议栈缓冲区，数据由协议栈自行处理（`tcp_ip` 进程）。

1. 以太网的 DMA 有两个拥有者

主机：数据可以被处理；

DMA 自身：只有在这种情况下 DMA 可以接收数据并产生中断。

特别需要注意的当 DMA 描述符所对应的缓冲区满的时候，该描述符会为主机所拥有；驱动程序是将数据拷贝到 lwip 的数据缓冲区后将该 DMA 拥有者置为 DMA 自身。

2. ST 的例程是采用中断机制来处理以太网的数据接收，FreeRTOS 下使用 Counter Semaphore 来对中断进行描述（Counter Semaphore 的增长是在中断处理函数中的），中断服务程序（`ethernetif_input`，后文称这个函数为 ISR，但这个任务并不在中断处理函数中，这个任务平常是阻塞的，只有得到信号量后才会运行）仅处理一个以太网 packet。

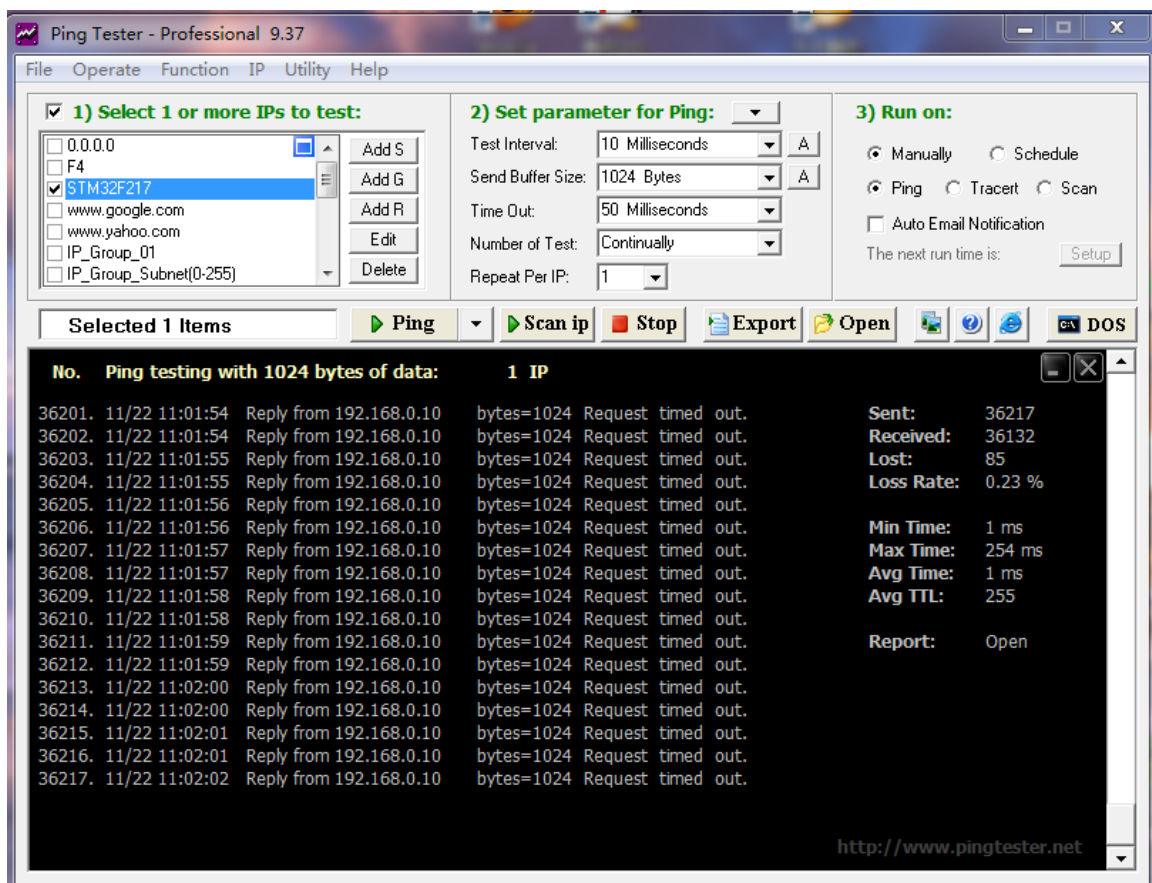
正常情况下，在数据包到达 STM32 的以太网接收缓冲区时，会进入中断，中断程序会将 Counter Semaphore+1，从而触发中断服务程序 ISR（`ethernetif_input`）对这个包进行处理。

Counter Semaphore	1	2	3	4	5	6	...
DMA Buffer	Data1	Data2	Data3	Data4	Data5	Data6	...

但是，如果有数据包送到接收缓冲区，但是没有及时产生中断的话（在操作系统中这是非常容易引起的，比如更高级的中断或者临界代码，这些操作都会屏蔽掉其它中断）。此时就会发生一种奇怪的现象，即 Counter Semaphore 的值与数据缓冲区的 packet 映射不对应。如下图所示

Counter Semaphore	3	5	6	7	8	15	...
DMA Buffer	Data1	Data2	Data3	Data4	Data5	Data6	...

这个时候网络状况就非常糟糕了，实际的 ISR 程序执行的非常晚，表现的方式就是大量的延时甚至是 Timeout。



但这还不是最糟糕的，一旦 DMA Buffer 满了，同时 Counter Semaphore 中没有任何可用信号量。按照刚才提到的，DMA 是无法再接收数据并产生中断了（RBUS = 1），即无法

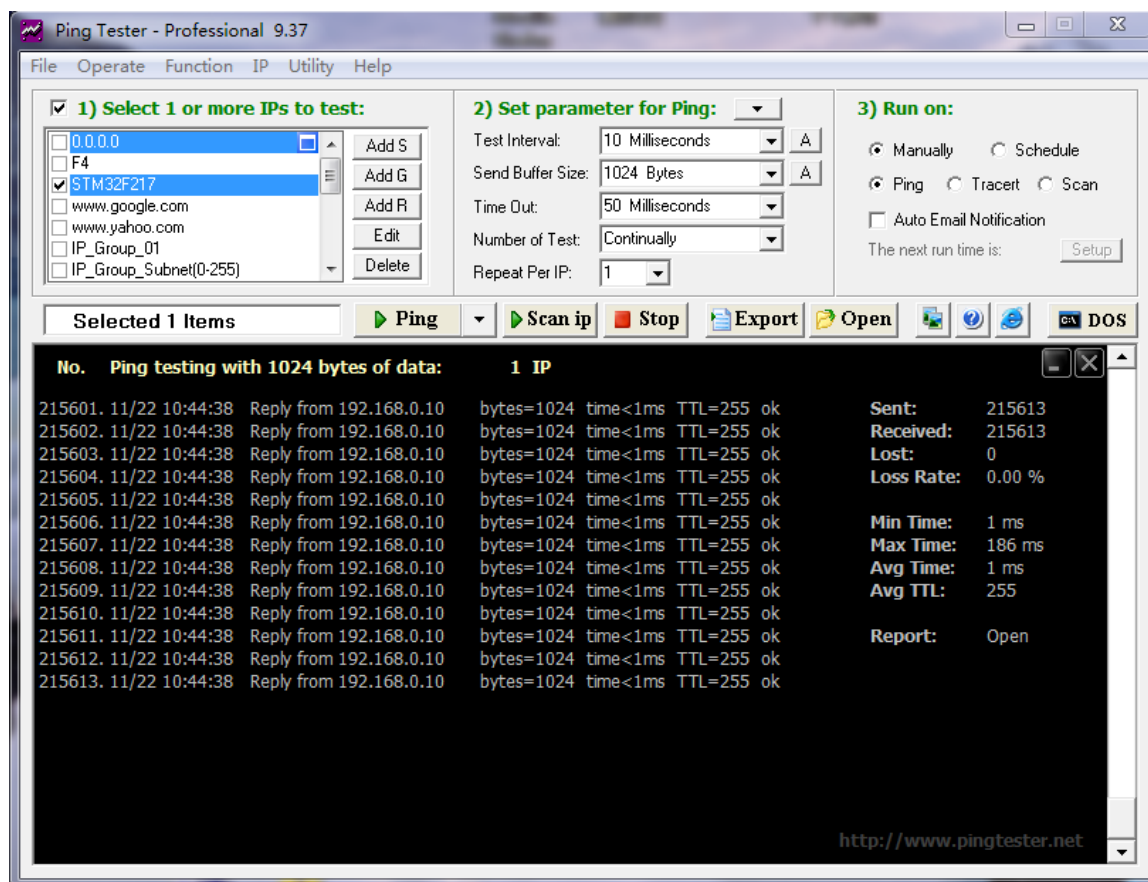
进入中断处理函数，从而也无法触发触发中断服务程序。网络就表现为断线。此时 ping 的响应就不是 timeout，而是 IP 无法到达（我在写这篇文档时候没出现这个情况，所以没有截图，但是以前是遇到过的）。

三. Stsw-stm32070 的改进

1. **ethernetif_input** 中处理多个以太网 packet，即只要进入这个程序，就会将以太网的缓冲处理完。

2. FreeRTOS 下中断采用二值信号量来触发阻塞任务。

大家可以简单分析一下，在这种情况下，是不会出现 stm32060 中的问题的。丢包和延时仍然是可能出现的。这是我自己利用新的驱动做的测试，2 个 ping tester(10ms, 1kbytes)，同时还打开 http 服务（事实上已经 http 已经完全没有响应了）和 cmd 的 ping(1s, 1000bytes)，这个测试压力比许多应用，stm32F4x7(168MHz)的仍然能保持网络的畅通性。



四. 结语

可以看出新的驱动确实较之过去有了明显的提升。由于 STM32F4 和 F2 的以太网外设是一样的，所以驱动也是相互兼容的。

最后需要说明的是，驱动也是工程师写的，在资源有限的情况下必然存在一些问题（总是有需要完善的地方）。对于应用工程师而言，压力测试的环境应该参考自己的实际应用场景来设计，远超出实际应用的压力测试其实是没有太大的意义。

欢迎朋友们继续交流。

ST 上海
2013.11.22