

信道化 Iub 口数据采集卡的驱动设计

梁燕, 黄标

(重庆邮电大学 通信与信息工程学院, 重庆 400065)

摘要:在高速数据采集中存在几个关键问题,如中断服务例程直接完成数据包处理导致 CPU 占用过高,频繁系统调用完成内核态到用户态数据搬移导致内存溢出采集丢包,系统时间回流导致多卡采集不同步,呼叫详细记录(CDR)合成失败等。文章在信道化 Iub 口数据采集卡的驱动设计中,提出了“利用延迟调度(DPC)、零拷贝缓冲环管理机制,采用动态的时间修正算法保证多卡的时间同步”的方法,有效降低了 CPU 开销,实现高速数据采集不丢包,解决了系统时间回流而导致的 CDR 合成问题。

关键词:Iub 接口;Windows 驱动模型;延迟调度;零拷贝

中图分类号:TH16;TP24 **文献标识码:**A

Designing of Driver System for Channelized Iub Data Acquisition's Card

LIANG Yan, HUANG Biao

(School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: In the scenario of high-speeding data collection, there exist some crucial problems such as high CPU usage due to direct completing the data packet processing by Interrupt service routine, memory overflow loss due to moving the data from kernel mode to user mode by frequently system call, and the un-synchronization among the multiple collection cards due to the system time back-off as well as the Call Detail Record (CDR) synthesis failure, etc. In this paper, a method of “using combination of deferred procedure call (DPC), zero-copy buffer ring management and using the dynamic time adjusting algorithm to ensure the multiple-card time synchronization” has been proposed to design the driver system for channelized Iub data acquisition's card. The method can release the CPU overhead effectively, achieve high-efficient data collection and solve the problem of CDR synthesis arising from the system time back-off.

Key words: Iub interface; Windows driver model(WDM); deferred procedure call(DPC); zero-copy

0 引言

Iub 接口是 3G 无线接入网中无线网络控制器(RNC)与 3G 基站(简称 Node B)间的逻辑接口,是信令监测的重点。实现 Iub 接口的信令采集需在 RNC 侧采用高性能的数据采集卡,实现对整个 RNC 的包围监测,从而达到全面衡量网络质量的需求。在 Iub 口数据采集卡的设计中,通常是通过直接存储访问(DMA)方式将收到的数据包传递给内核层的驱动程序,驱动程序负责维护和管理内核态的双倍速率同步动态随机存储器(DDR SDRAM)缓存环,并利用延迟调度(DPC)机制把 DMA 传递的数据重新进行组装,及时为数据包添加正确的时间标签后映射到用户态空间。驱动程序的设计需解

决两个关键问题:

1) 在高速的数据采集中减少内存拷贝次数,降低系统调用导致的 CPU 开销,让 CPU 把更多的资源分配给上层协议分析。

2) 如何保证单台设备上多张板卡的时间标签同步,解决多台设备系统时间漂移而导致的时间“回流”问题。

本文采用 Windows 驱动模型(WDM)的 PCI 驱动架构,利用 DPC 和零拷贝缓冲环管理机制实现高效的数据采集,减少内存拷贝,降低系统开销;采用动态的修正算法保证单台设备多张板卡的时间标签同步,解决多台设备系统时间漂移而导致的时间“回流”,以及上层应用的 CDR 合成准确性问题^[1]。

收稿日期:2012-03-08

作者简介:梁燕(1977-),女,重庆涪陵人,讲师,硕士生,主要从事嵌入式系统、通信网测试技术的研究。该文为重庆市科技进步奖一等奖《移动通信网信令监测与优化关键技术研究及其应用》(2011-J-1-11-D01)子项目。E-mail:liangyan@cqupt.edu.cn

1 整体架构设计

图 1 为信道化 Iub 口采集设备的驱动与应用编程接口(API)、固件间的关系。驱动负责响应上层 API 的调用,把相应的参数和命令写入 DDR 内存,控制板卡固件执行相应操作,完成采集硬件的初始化配置。驱动程序采用 DPC 机制处理 DDR 的共享控制状态队列,负责把数据按照自定义的格式重新进行组装,及时为数据包添加时间标签,并写入扩展内存。

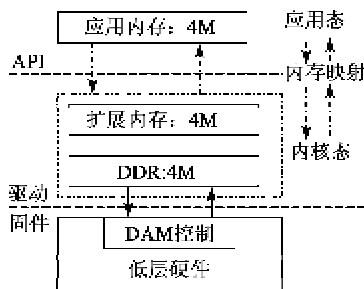


图 1 整体架构

2 DPC 延迟调度设计

DPC 为推迟了的过程(函数)调用,在 DISPATCH_LEVEL 级别上执行。DPC 主要是为设备驱动程序提供的,最常用就是当处理时间片(Repeat Time)到来时,内核自动启动执行其 DPC 函数。在 WDM 的功能驱动设计中,由于数据到达 PC 机的时间不定,如果直接采用硬件中断来实现数据处理,在流量大时,硬件中断频繁,若同时把复杂的数据处理放到中断服务例程(ISR)中执行,故会导致丢包、严重时导致 CPU 占用率太高,操作系统出现“假死机”,上层应用很难获得处理时间片。为了解决这个问题,通常使用 DPC 机制来完成复杂的数据处理,ISR 只完成一些必要的硬件恢复操作就尽快退出,把需花大量时间的工作交给 DPC 延迟队列处理。只要找到“线速的最大流量、动态缓存大小、DPC 延迟函数执行效率”三者之间的折中点,就可设置一个合适的 DPC 延迟队列产生时间,保证驱动在不丢包的情况下完成数据的采集,但又不过多的占用系统资源。图 2 为在驱动设计中 DPC 对象与其回调函数 DpcForPack 和定时器 KTIMER 间的关系和配置流程。一旦启动 DPC 后,只要 DPC 的处理时间片到达,回调函数 DpcForPack()就完成数据的处理,负责给采集到的每一个数据包打上精准的时间标签^[2]。

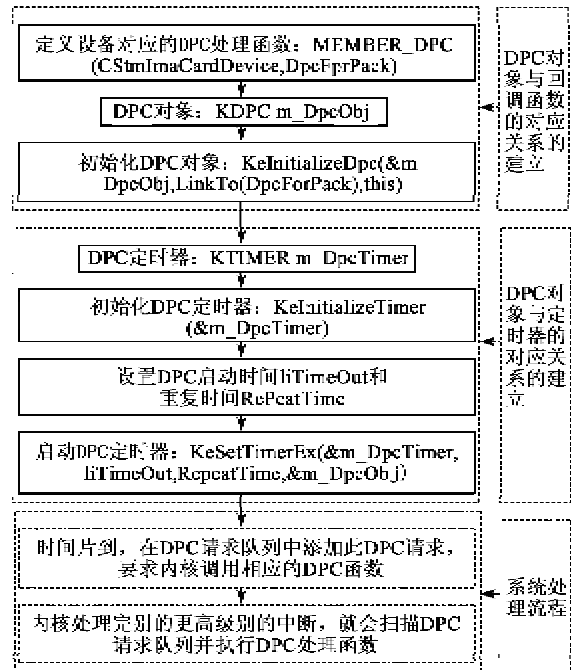


图 2 DPC 实际使用方法

图 2 中涉及到一个系统函数“KeSetTimerEx()”,其原型为“BOOLEAN KeSetTimerEx (IN PKTIMER Timer, IN LARGE_INTEGER DueTime, IN LONG Period, IN PKDPC Dpc)”,其中, Timer 为指向一个 KTIMER 数据结构; DueTime 为所定的时间长度,指此 DPC 定时以当前时间为参考点(负数),经过 DueTime 后第一次启动 DPC,单位为毫秒,注意与参数 Period 区别; Period 为 DPC 定时器的重复周期,单位为毫秒; Dpc 为 KDPC 结构指针,指向前面设置好的 KDPC 结构。

3 零拷贝缓冲环管理设计

高性能数据包捕获平台要求大量的内存用来缓存网络数据包,分配静态缓冲区可消除每次数据包传输时都要进行的缓冲区分配和释过程所付出的系统开销代价;零拷贝技术首先利用 DMA 技术将网络数据包直接传递到系统内核预先分配的地址空间中,避免 CPU 的参与;同时,将系统内核中存储数据包的内存区域映射到用户空间,应用程序直接对这块内存进行访问,减少了系统内核向用户空间的内存拷贝和常规的系统调用开销。在本设计中预留一段固定大小的物理内存存放数据包,并将此缓冲区分成大小为“可配置长度”的固定缓冲块用来存放数据流,将此缓冲区构造成一个环形的缓冲队列。图 3 为环形缓冲队列。由图可知,环形缓冲队列为

驱动分配的静态缓冲区域,该区域被驱动直接映射到用户空间,驱动和上层应用通过 Win32 事件通知机制来同步管理缓冲环,减少内存的拷贝操作,从而减少系统 CPU 开销^[3]。

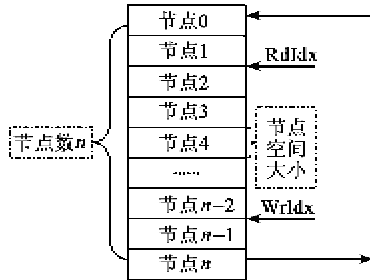


图3 环形缓冲队列

4 时间修正算法设计

卡采用 DPC 延迟处理调度机制,来实现周期性的数据处理。每次进入 DpcForPack(),获取到的系统时钟 dst 总是要与上一次进入 DPC 的系统时钟相减,求取差值 DiffTime,按照此次要处理的数据包个数,把差值均匀分配给每个包,然后再与 dst 相加来获得每个数据包的时间标签,图4为时间标签的分布示意图。如先假定抓包个数为 n_{total} ,数据包 i (初始值为 1,最大值为 $n_{total} - 1$)的时间标签为 $dst + (DiffTime \div n_{total}) \times i$ 。

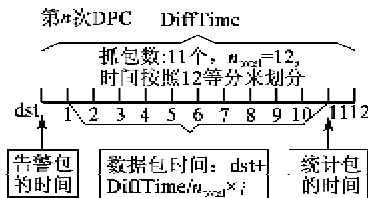


图4 时间标签分布示意图

图5为 DpcForPack()中时间标签的处理流程。为了防止多台采集机系统时间的漂移,实现多卡采集时间标签的同步,代码还需要周期性的修正系统启动时间,防止因为时间“回流”而导致的通信协议

流程合成出错问题。

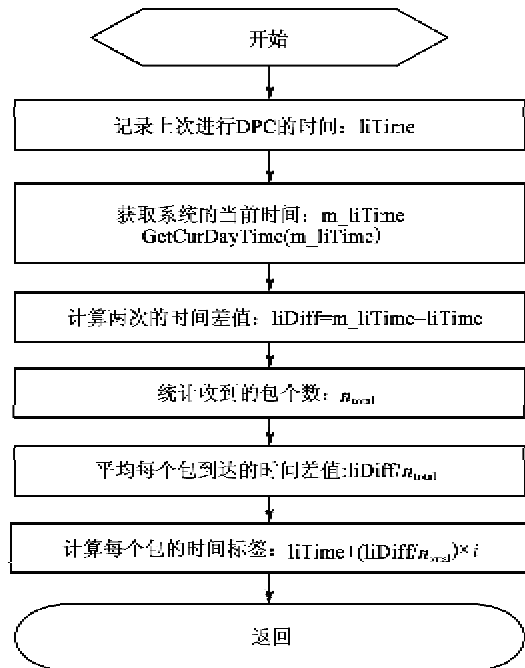


图5 DPC处理函数的时间标签处理流程

5 结束语

本文分析了信道化 Iub 口数据采集的驱动整体架构设计,详细介绍了驱动设计涉及到 DPC 延迟调度、精确时间标签、缓冲环管理几个关键技术;并给出了项目实际操作中的实现方法,目前基于本文技术设计的数据采集卡在 3G 无线监测仪表和系统上得到了广泛的应用,工作稳定可靠,能很好的完成信道化 Iub 口数据采集任务。

参考文献:

- [1] 武安河. Windows 2000/XP WDM 设备驱动程序开发[M]. 2版. 北京:电子工业出版社,2005.
- [2] 居锦武,王兰英. Windows 2000 异步过程调用的分析与应用[J]. 计算机技术与发展,2008,18(5):191-193.
- [3] 孙友. PCI 设备的 WDM 驱动程序设计[J]. 电子测量与仪器学报,2004,18(21):231-237.

(上接第 455 页)

- [7] HUANG C F, TSENG Y C. A survey of solutions to the coverage problem in wireless sensor networks[J]. Journal of Internet Technology, 2005, 15(9):121-125.
- [8] 朱有产, 窦炳林. 基于依赖图的网络故障定位算法[J]. 大连理工大学学报, 2005, 45: 73-76.

- ZU Youcan, DOU Binlin. Network fault location algorithm based on dependency graphs[J]. Journal of Dalian University of Technology, 2005, 45: 73-76.
- [9] Crossbow. MTS/MDA sensor and data acquisition boards user's manual[M]. S. l.;Crossbow Technology Inc,2003(5):85-88.