第 34 卷 第 24 期 计 算 机 工 程 2008 年 12 月
Vol.34 No.24 Computer Engineering December 2008
·网络与通信· 文章编号：1000—3428(2008)24—0106—03 文献标识码：A 中图分类号：TP393.07

基于 IXA 的网络测试系统设计与研究

韩英强，胡越明
(上海交通大学计算机科学与工程系，上海 200240)

摘 要：将网络处理器应用于网络测试领域，设计并实现一个网络测试系统。讨论网络测试领域的研发现状，从软硬件角度叙述该系统的架构及主要功能，分析其处理流程和设计过程中的关键问题。对主要模块的测试结果验证了系统性能，表明网络处理器为开发高性能网络测试系统提供了一个较好的解决方案。

关键词：网络处理器；互联网交换架构；网络测试

Design and Research of Network Test System Based on IXA

HAN Ying-qiang, HU Yue-ming
(Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200240)

【Abstract】 This paper applies the network processor into the network test field, designs and implements a network test system. It discusses the research and development status about the network test field, depicts the main functions about the the software and hardwar of the system, and analyses the disposal flow and the key problem in the system designing. It validates the capability of the system by testing its main modules, and testing results show that the network processor provides a good solution for the exploitation of a network test system with high capability.

【Key words】 network processor; Internet eXchange Architecture(IXA); network test

1 概述

网络性能评估涉及对网络设备性能的测试以及对网络拓扑结构合理性的验证，随着互联网的发展和普及、网络产品数量与种类的增多，其重要性越来越明显。网络测试系统可以为各种测试产生大量数据，用于评估被测试实体的性能。

多数网络仿真测试领域的产品是电信级的，如 Spirent 公司的 SMARTBits、Navtel 公司的 IP InterEmulator。上述产品性能稳定、用户界面友善，但存在一些固有缺陷，如系统设置复杂、功能无法真正定制。

网络处理器是近年来兴起的一种适应高速网络的专用处理器，它通过专门优化包处理，将数据包以线性速度送到下一个节点。基于 Intel 互联网交换架构(Internet eXchange Architecture, IXA)网络处理器开发的网络测试系统具有以下优势：(1)多端口同时工作。网络处理器固有的数据处理性能可以实现对多个端口数据进行同时线速仿真。(2)实现了多功能组合。利用网络处理器微引擎代码的动态下载特性可以根据需要不断添加新的功能以满足不同网络测试的需要。(3)数据处理速率极高。基于IXA的网络测试系统对数据量的限制不在于处理器的计算能力，而是带宽问题。例如，IXP2800和IXP2400网络处理器分别以 10 Gb/s和 2.5 Gb/s的速度工作，并采用支持处理器同时执行大量操作的“超任务链接”[1]技术，从而确保网络测试系统以最高性能运行。(4)多种网络测试。网络处理器提供与MAC, PHY设备的标准接口，使不同网络(如Ethernet, ATM, Token2Ring等)的MAC, PHY设备可以与网络处理器实现连接。

2 网络测试系统的主要功能模块

网络测试包括网络设备的测试和对网络系统本身的测试。网络测试系统包括以下功能模块：(1)流量模块。主要功能是完成数据包发送，但它可以发送的数据不止数据包一种结构，而是包括多种不同的、带有层次结构的数据。(2)数据捕捉模块。作用是记录当前端口接收到的所有数据包，可以通过多种图形化视图界面对这些数据包的内容进行观察和分析。(3)协议模拟模块。主动产生各种真实的网络流量，能模拟在不同网络环境下产生不同网络流量。(4)网络攻击模块。用户可以选择不同攻击方式，对相应网络产品进行测试。

3 基于 IXA 的测试系统架构

一个完整的网络测试系统设计包括系统硬件架构设计和系统软件架构设计。硬件架构是一个系统中硬件接口之间的链接关系，软件架构是软件模块之间的调用关系。

3.1 硬件架构

为了便于使用，本文以一台采用通用 CPU 的 PC 机系统来运行系统配置管理软件作为人机界面，实现对网络测试系统的配置和管理，可以方便地通过 PCI/Ethernet 来实现两者的通信，如图 1 所示。处理器单元对网络处理器微引擎单元和其他硬件单元进行控制，并执行上层协议和各种应用程序，网络处理器微引擎单元主要对数据包进行处理和收发，网络处理器和外界进行数据交换需要通过网络接口单元。

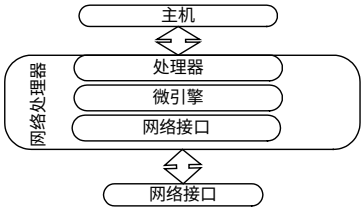

图 1 网络测试系统硬件架构

作者简介：韩英强(1981－)，男，硕士研究生，主研方向：计算机网络，网络安全；胡越明，副教授

收稿日期：2008-05-05 **E-mail：**yingqiang.h@163.com

3.2 软件架构

如图 2 所示，系统软件架构由用户配置管理模块、网络数据调度管理模块和网络数据处理模块 3 个主要模块构成，具体如下：

(1)用户配置管理模块，运行于系统配置管理平台上，具体形式为 Windows 或 Linux 下的应用程序。它提供了一个良好的人机交互界面，方便了系统的管理与控制。

(2)网络数据调度管理模块，运行于 IXP 的 XScale 微处理器上。它先根据用户配置管理模块提供的信息决定在 IXP 的微引擎上运行哪个指令模块，然后将用户配置管理模块提供的功能运行参数转换成数据处理模块可用的参数。

(3)网络数据处理模块，运行于网络处理器 IXP 的微引擎上。该模块可以根据需要动态更改并下载对应的微码指令集以实现相应功能。网络数据处理调度管理模块运行一个服务端程序并等待用户配置管理模块与其建立通信。正确建立两者之间的通信后，客户端实现用户操作控制的图形化，并将用户操作配置转化成数据，将其发送给主机。主机针对客户端的请求做出响应，并将结果发送给客户端显示。

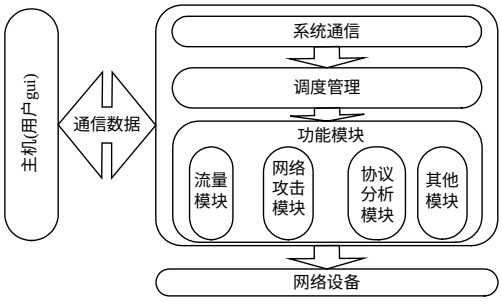

图 2　网络测试系统软件架构

4　网络测试系统的实现

网络测试系统是一个复杂系统，涉及很多数据结构、策略和流程。基于 IXA 的网络测试系统可以在 IXP2400, IXP2800, IXP425 等板块上实现。在不同板块上系统的实现略有不同，但基本思想和基本策略大致相同，本文对该系统实现方法的介绍以 IXP425 为主。

4.1 主要数据结构

在本文系统实现中，需要设计很多复杂的数据结构，这些数据结构主要用于数据的传递和信息的交换，其中最主要的数据结构包括：

(1)通信信息数据结构，用于将用户的配置信息传递给 IXP 的主要数据结构。其中包含如下信息：功能模块 ID，有效数据的长度，有效数据部分。其中，有效数据部分是对具体功能模块的配置信息；功能模块 id 分别代表流量模块、协议分析模块、网络攻击模块等。

(2)网络缓存结构。缓存管理贯穿整个系统，网络缓存是整个系统操作的主要数据结构。发送数据时，在IXP的软件包中，该数据结构是IX_OSAL_MBUF[2]，主要分为 3 层，如图 3 所示。

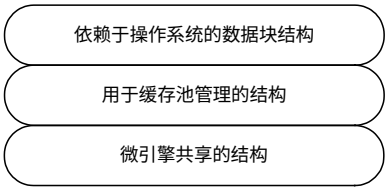

图 3　IX_OSAL_MBUF 的 3 层结构

依赖操作系统的部分是本文关注的重点，该部分与操作系统的网络缓存相对应，Linux系统中对应的是skbuffs[3]。IX_OSAL_MBUF 中的 ix_osbuf_ptr 保存 Linux 系统的中的 skbuffs指针，该部分数据格式如图 4 所示。

	0	1	2	3
0	下一个缓冲地址(ix_next)			
4	下一个数据包的地址(ix_nextPacket)			
8	数据部分地址(ix_data)			
12	有效数据的长度(ix_len)			
16	缓存类型 (ix_type)	缓存标识 (ix_flags)	保留字段 (ix_reserved)	
20	保留字段			
24	缓存的长度			
28	前一个缓冲地址(ix_priv)			

图 4　IX_OSAL_MBUF 中依赖操作系统的数据块字段

4.2 高速缓冲策略

高速缓冲可以有效提高系统性能，增强系统处理数据的能力。由于 IX_OSAL_MBUF 缓存可能被操作系统的网络协议栈进行多次访问，因此对 IX_OSAL_MBUF 进行高速缓冲处理，可以有效提高系统的操作效率。在本文系统中，在发送数据过程和接收数据过程中，高速缓冲策略有所不同，具体如下：

(1)发送数据。IX_OSAL_MBUF 在高速缓冲器中的备份数据被修改后，这种修改可能没有被写入内存中的 IX_OSAL_MBUF。为了确保内存中的数据的是最新的，必须将高速缓冲器中数据被修改的部分清空。高速缓冲器的清空策略必须遵循以下原则：

1)系统程序负责将IX_OSAL_MBUF中的数据部分清空，但不是清空整个数据部分，而只是清空 XScale 处理器修改的部分。

2)IXP 软件框架负责清空 IX_OSAL_MBUF 数据头部分的 ix_ne 共享部分。此清空操作必须在将 IX_OSAL_MBUF 提交给网络处理器微引擎(NPE)之前完成。

(2)接收数据。NPE 修改 IX_OSAL_MBUF 后，这种修改将反映在内存单元中，而 XScale 处理器 Cache 中的数据备份并没有被更新。为了确保 Cache 的数据备份是最新的，必须使 Cache 中相应部分的数据无效。接收数据的处理策略应该遵循以下原则：

1)系统程序负责使 IX_OSAL_MBUF 的数据部分无效。因为只有系统程序知道需要访问数据部分的那些字段，所以在将 IX_OSAL_MBUF 提交给 IXP 软件框架之前，最好使 IX_OSAL_MBUF 对应的 Cache 部分无效。

2)IXP 软件框架负责修改 IX_OSAL_MBUF 数据头部分的 ix_ne 共享部分。由于 IXP 软件框架在将 IX_OSAL_MBUF 传递给 NPE 之前，会修改 IX_OSAL_MBUF 中的数据头部分，因此需要使数据头部分无效。这个操作应该在将 IX_OSAL_MBUF 提交给 NPE 之前完成。

4.3 接收和发送流程

在系统设计中，无论流量产生模块还是协议分析模块，数据包的接收和发送是整个系统的关键。

4.3.1 发送流程

发送流程是一个复杂的过程，涉及基本发送流程、相应的缓存管理和优先级管理，具体如下：

(1)基本发送流程。以太网访问组件提供了一种将具有相应优先级的帧发送到指定以太网 MAC 上的机制。一个发送流程可以概括为如下 7 个步骤：

1)将合适的 NPE 镜像(NPE image)下载到 NPE 中，并对其初始化。

2)发送的端口号必须初始化。

3)为该端口注册一个回调函数，当发送缓存被放到 TxDone 队列中时，该回调函数会被调用。

4)配置好该端口后，它必须被激活才可以用来发送数据。

5)提交一个数据帧，设置合适的优先级，该操作会把一个 IX_OSAL_MBUF 放到该端口的发送队列中。

6)NPE 用以太网协处理器和以太网 MAC 来发送“发送队列”中的一个数据帧，该数据帧被发送出去后，相应的 IX_OSAL_MBUF 会被放到 TxDone 队列中。

7)完成一个数据帧的发送后，为该端口注册的回调函数被调用，该回调函数接收一个指向 IX_OSAL_MBUF 的指针，并进行一些清理操作。

(2)发送缓存管理和优先级。因为发送过程中需要对数据进行缓存处理，所以发送系统在不同模块中要用不同队列对相应的数据进行缓存处理，主要有以下 3 个队列：

1)以太网发送组件(IxEthAcc)中的软件队列，当下游队列满时，该软件队列用来对数据进行缓存。

2)IxQMgr(IXP 软件框架中的一个队列管理组件)队列，该队列是 NPE 上游的一个队列，用来向 NPE 传递数据或从 NPE 接收数据。IxQMgr 支持 2 种队列，即发送队列(TxEnet)和发送完成队列(TxEnetDone)。对于发送队列，每个端口(port)支持 128 个入口。对于发送完成队列，组件单独支持 128 个入口。

3)NPE 微码(microcode，是 NP 系列支持的一种语言)队列，该队列用来保存 IX_OSAL_MBUF 中的数据头部分，支持 64 个入口。

数据经过 3 个队列的顺序如图 5 所示，数据在没有被放入硬件队列之前，先缓存到软件队列中。当 IxQMgr 队列为空或队列中的数据达到最低门槛时，会从软件队列中取出数据放入 IxQMgr 队列中。此时，数据头会被传输到 NPE 队列中，如果设定了优先级，NPE 还会对数据帧进行重排序，以确保高优先级先被发送出去。完成一个数据帧发送后，相应的缓存会被放到发送完成队列(TxEnetDone)中，并交由系统的回调函数来处理。

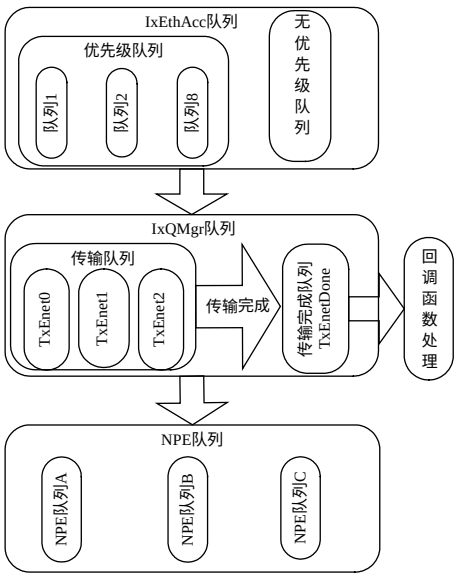

图 5　数据发送队列

4.3.2　接收流程

接收数据是数据捕获模块的关键部分，无论数据分析还是流量统计，接收流程都是其中必不可少的一部分，涉及接收基本流程和接收缓存策略，具体如下：

(1)接收基本流程。在接收数据前，必须给以太网访问组件提供一个接收缓存，该缓存用来存储接收到的数据。数据接收流程主要包括如下 9 个步骤：

1)将合适的 NPE 镜像下载到 NPE 中，并对其初始化。

2)接收端口号必须初始化。

3)为该接收端口注册一个单缓存或多缓存的回调函数。

4)在接收数据前预先载入接收缓存。

5)在配置好接收端口且载入了接收缓存后，必须激活接收端口，来接收数据。

6)数据帧被接收，并被放入 IxQMgr 的接收队列中。

7)IxQMgr 激活以太网访问组件(IxEthAcc)来处理整个接收队列，或采用轮询方式来处理一定数量的缓存。

8)以太网访问组件(IxEthAcc)从接收队列中取出数据帧。

9)以太网访问组件(IxEthAcc)激活回调函数来进行其他处理。

(2)接收缓存管理。接收子系统必须给 NPE 提供必须的接收缓存。为了保证接收子系统具有较高效率，IX_OSAL_MBUF 必须大小合适。由于 NPE 以 64 Byte 的大小写入数据，因此 IX_OSAL_MBUF 需要以 64 Byte 对齐。

IxQMgr 组件给 NPE 提供了 3 个缓存队列，每个端口对应一个队列。如果一个端口对应的 IxQMgr 缓存队列已满，则需要把相应缓存暂时放到软件队列中，当该 IxQMgr 缓存队列中的缓存达到最低值时，将软件队列中的缓存(如果软件队列有缓存的话)转移到该 IxQMgr 缓存队列中。

在系统设计中，要确保有充足的缓存来接收数据。否则，当一个以太网帧到达时，如果没有缓存来存放该帧，NPE 将丢弃该以太网帧，导致丢包现象。

5　系统性能

本文系统主要涉及 2 种基本性能测试，即发送数据性能测试和接收数据性能测试。通过这些基本测试可知，此系统具有较高性能。

本文测试环境为 100 Mb/s 以太网。接收数据和发送数据的性能测试结果分别如图 6、表 1 所示。

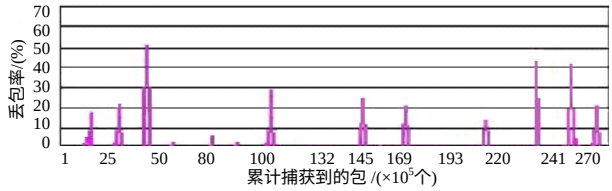

图 6　接收数据的性能测试结果

表 1　发送数据的性能测试结果

测试用例	速率
TCP 链接建立速度	60 000 次/s~70 000 次/s
TCP 链接维护个数	1 900 000 个~2 000 000 个
HTTP GET 发送速率	34 000 次/s~40 000 次/s

6　结束语

网络测试系统需要测试多方面内容，具有复杂性。采用 IXP 设计网络测试系统有如下优点：(1)系统具有较高性能。IXP 是专门针对网络应用的高性能处理器，很适合数据的产

(下转第 111 页)